Université de Paris

École doctorale de Sciences Mathématiques de Paris Centre (ED 386)

Laboratoire : Mathématiques Appliquées à Paris 5, MAP5-UMR 8145 CNRS

THÈSE DE DOCTORAT

Spécialité : Mathématiques appliquées

par

Arthur Leroy

Apprentissage de données fonctionnelles par modèles multi-tâches : application à la prédiction de performances sportives

dirigée par
Servane Gey

Présentée et soutenue publiquement le 9 décembre 2020 devant le jury composé de :

Christophe BIERNACKI	PR, Université Lille 1	Rapporteur
Etienne BIRMELE	PR, Université de Strasbourg	Examinateur
Servane GEY	MCF-HDR, Université de Paris	Directrice
Benjamin GUEDJ	CR, Inria et UCL	Co-encadrant
Julien JACQUES	PR, Université Lumière Lyon 2	Examinateur
Pierre LATOUCHE	PR, Université de Paris	Co-encadrant
Emilie LEBARBIER	PR, Université Paris Nanterre	Examinatrice
Adeline LECLERCQ-SAMSON	PR, Université Grenoble Alpes	Rapporteure

Après avis des rapporteurs : Adeline LECLERCQ-SAMSON
Christophe BIERNACKI

MAP5 - Université de Paris
45 rue des Saints Pères
75006 Paris

Abstract

The present document is dedicated to the analysis of functional data and the definition of multi-task models for regression and clustering. The purpose of this work is twofold and finds its origins in the problem of talent identification in elite sports. This context provides a leading thread illustrative example for the methods and algorithms introduced subsequently while also raising the problem of studying multiple time series, assumed to share information and generally observed on irregular grids. The central method and the associated algorithm developed in this thesis focus on the aspects of functional regression by using multi-task Gaussian processes (GPs) models. This non-parametric probabilistic framework proposes to define a prior distribution on functions, generating data associated with several individuals. Sharing information across those different individuals, through a mean process, offers enhanced modelling compared to a single-task GP, along with a thorough quantification of uncertainty. An extension of this model is then proposed from the definition of a multi-task GPs mixture. Such an approach allows us to extend the assumption of a unique underlying mean process to multiple ones, each being associated with a cluster of individuals. These two methods, respectively called Magma and MagmaClust, provide new insights on GP modelling as well as state-of-the-art performances both on prediction and clustering aspects. From the applicative point of view, the analyses focus on the study of performance curves of young swimmers, and preliminary exploration of the real datasets highlights the existence of different progression patterns during the career. Besides, the algorithm Magma provides, after training on a dataset, a probabilistic prediction of the future performances for each young swimmer, thus offering a valuable forecasting tool for talent identification. Finally, the extension proposed by MagmaClust allows the automatic construction of clusters of swimmers, according to their similarities in terms of progression patterns, leading once more to enhanced predictions. The methods proposed in this thesis have been entirely implemented and are freely available.

Keywords: Gaussian Processes, multi-task learning, functional data, curve clustering, EM algorithms, variational inference

Résumé

Ce manuscrit de thèse est consacré à l'analyse de données fonctionnelles et la définition de modèles multi-tâches pour la régression et la classification non supervisée. L'objectif de ce travail est double et trouve sa motivation dans la problématique d'identification de jeunes sportifs prometteurs pour le sport de haut niveau. Ce contexte, qui offre un fil rouge illustratif des méthodes et algorithmes développés par la suite, soulève la question de l'étude de multiples séries temporelles supposées partager de l'information commune, et généralement observées à pas de temps irréguliers. La méthode centrale développée durant cette thèse, ainsi que l'algorithme d'apprentissage qui lui est associé, se concentrent sur les aspects de régression fonctionnelle à l'aide d'un modèle de processus Gaussiens (GPs) multi-tâche. Ce cadre probabiliste non-paramétrique permet de définir une loi a priori sur des fonctions, supposées avoir généré les données de plusieurs individus. Le partage d'informations communes entre les différents individus, au travers d'un processus moyen, offre une modélisation plus complète que celle d'un simple GP, ainsi qu'une pleine prise en compte de l'incertitude. Un prolongement de ce modèle est par la suite proposé via la définition d'un mélange de GPs multi-tâche. Cette approche permet d'étendre l'hypothèse d'un unique processus moyen sous-jacent à plusieurs, chacun associé à un groupe d'individus. Ces deux méthodes, nommées respectivement Magma et MagmaClust, offrent de nouvelles perspectives de modélisation ainsi que des performances remarquables vis-à-vis de l'état de l'art, tant sur les aspects de prédiction que de clustering. D'un point de vue applicatif, l'analyse se concentre sur l'étude des courbes de performances de jeunes nageurs, et une première exploration des données réelles met en évidence l'existence de différents patterns de progression au cours de la carrière. Par la suite, l'utilisation de l'algorithme Magma, entrainé sur la base de données, attribue à chaque sportif une prédiction probabiliste de ses performances futures, offrant ainsi un précieux outil d'aide à la détection. Enfin, l'extension via l'algorithme MagmaClust permet de constituer automatiquement des groupes de nageurs de part les ressemblances de leurs patterns de progression, affinant de ce fait encore les prédictions. Les méthodes détaillées dans ce manuscrit ont également été entièrement implémentées et sont partagées librement.

Mots-Clefs : Processus Gaussiens, apprentissage multi-tâche, données fonctionnelles, clustering de courbes, algorithmes EM, méthodes variationnelles

Remerciements

A l'heure de dresser le bilan de l'aventure que constitue la thèse, parfois solitaire en cette si particulière année 2020, il est bon de se rappeler à quel point celle-ci résulte d'un effort collectif. Que ce soit consciemment ou par un geste anodin, toutes les personnes cités ci-dessous ont contribué à leur manière à l'aboutissement de ce projet, et pourraient légitimement revendiquer un mot, une phrase, ou tout un chapitre pour certains.

Il était difficile d'évoquer en introduction autre chose que l'esprit d'équipe pour exprimer mon sentiment à l'égard de mes trois encadrants de thèse. Chacun à votre façon et avec une complémentarité remarquable, vous m'avez donné le goût, les outils et l'exigence nécessaire pour tenter de participer au bel effort collectif qu'est la recherche scientifique. Pas une fois en 3 ans vous ne m'avez fait sentir comme un petit doctorant ignorant (ce n'est pas faute d'avoir dit des bêtises en pagaille pourtant). La patience et la confiance que vous m'avez témoigné ont contribué à m'offrir un cadre de travail serein et idéal.

Merci tout d'abord à toi Servane, qui m'a fait confiance il y a maintenant plus de trois ans, alors même que je sortais de nul part, pour mener un projet encore bien vague à l'époque. Ton soutien permanent, qu'il soit scientifique ou amical (dont les multiples pauses café, sans café pour moi), aura été essentiel pour m'aider à garder le cap durant les nombreuses tempêtes qui ont émaillé ce voyage, et pour finalement nous amener à bon port.

Merci à toi aussi Benjamin, embarqué de la première heure également ! J'espère que tu me pardonneras de t'avoir honteusement pillé des scripts sur GitHub, ton process de rédaction, et quelques unes de tes expressions venues du 'bon côté de la Manche'. Je finis d'essayer de percer le mystère de ton énergie même en l'absence de sommeil, et ta capacité à comprendre les enjeux d'un problème en 7 secondes chrono, et après j'arrête promis.

Et pour finir, j'aimerais te remercier, Pierre, très sincèrement. Car j'ai encore du mal à comprendre aujourd'hui où tu as trouvé la bienveillance pour me tendre la main durant une période très compliquée, où tu n'avais probablement pas grand chose à gagner à m'aider, à part des ennuis. Je te dois la découverte de sujets qui m'ont passionné pendant la 2ème partie de cette thèse, une plus grande rigueur, et une bien meilleure connaissance de la forêt de Fontainebleau (et de l'inspiration qu'elle offre pour faire des maths) ! Rarement des séances de travail m'auront paru aussi stimulantes et seront passées aussi vite, au point de trouver les années de thèse presque trop courtes.

A l'heure du coup de sifflet final, il m'apparait évident que sans cette équipe, de deux titulaires indiscutables, et d'une recrue de choc au mercato, le match de mon doctorat aurait été perdu par forfait avant même la mi-temps.

Par ailleurs, j'aimerais également remercier chaleureusement Christophe Biernacki et Adeline Leclercq-Samson pour avoir accepté de rapporter ma thèse. Il me tarde de pouvoir échanger avec vous autour des remarques et questions que vous avez très aimablement pris

le temps de soulever après lecture de mes travaux. Tu avais également participé, Adeline, à ma soutenance de mi-thèse aux côtés d'Étienne, et tiens de nouveau à vous remercier pour votre écoute à cette occasion. Vous avoir tous les deux dans mon jury est un réel plaisir, doublé d'une certaine fierté. Merci enfin à Julien Jacques et Émilie Lebarbier de me faire l'honneur de participer à ce jury. J'ai parcouru pendant ma thèse certains de vos travaux, qui m'ont chaque fois offert de précieuses clefs de compréhension, et j'espère en découvrir de nouvelles à l'occasion des échanges à venir.

Tous ceux qui me connaissent (et ceux qui ont déjà mal au crâne après la lecture de ce début de remerciements) savent que la concision n'est pas exactement mon point fort. Je vais pourtant essayer en un minimum de pages, de saluer un maximum des personnes (en espérant en oublier le moins possible, pardonnez moi d'avance) qui auront eu un rôle important dans des aspects plus personnels de ma vie durant la thèse.

Puisqu'il faut définir un ordre, choisissons le chronologique. Merci donc à Geoffroy et Adrien pour votre confiance initiale, et pour la part non-négligeable des premières réflexions autour de ce projet qui vous reviennent. Une mention spéciale évidemment à mon irremplaçable Moussa-illon, 'à jamais le premier' parmi mes frères et sœurs de thèse ! Pour basculer vers un autre co-auteur de talent, un grand merci Robin et à bientôt sur nos futurs projets. Au rang des irremplaçables, un certain Andy, a.k.a. le Zidane pépère du moulin neuf, atteint également une place de choix, et mérite amplement toute ma gratitude. J'aimerais pouvoir en dire plus à chacun d'entre vous, Guillaume, Juliana (ailière droite épidémiologiste au pied cassé, encore désolé !), Thibault, Adrien, Nicolas, Julien, Quentin, Joana, l'irremplaçable Hélène, my favorite not-so-American-and-almost-French mate Stacey, Jérémy, mes deux ex-stagiaires préférés, Romain et Maxime, et bien d'autres parmi les stagiaires, et les co-équipiers du presque-officiel et quasi-invincible Insep Football Club !

J'aimerais également adresser un mot aux compagnons de la désormais lointaine période de stage, des premiers pas dans le monde de la recherche avec l'équipe Tao, mais aussi à Thomas, Helena, et le reste de la team EDF. Salutations évidemment à tout le groupe jeune de la SFdS, et bien sûr au groupe *Stat et Sport*. Merci à Christian, Brigitte, Jérémy, Geoffrey, Marta et tous les autres d'avoir lancé cette belle aventure.

Un très grand merci à Anne et Fabienne pour leur soutien indéfectible, à Marie-Hélène, Julien, et l'ensemble des équipes administratives et techniques pour leur aide si précieuse durant ces années passées au MAP5. Une salutation générale pour tous ceux, permanents ou passagers, que j'ai pu croiser et côtoyer, et qui contribuent à cette ambiance si particulière et chaleureuse du MAP5 que j'ai envie de nommer *l'esprit terrasse du 7ème*. Une pensée amicale également pour l'équipe de l'IUT STID, Jérôme, Élisabeth, Florence, FX, Marie, Magali et bien d'autres, qui m'ont si gentiment accueilli et aidé sur le versant pédagogique de ce doctorat.

Cher(e) lecteur(trice), reprends ton souffle, car nous nous attaquons maintenant à un très gros morceau : les fameux éphémères du MAP5, c'est-à-dire des frères et sœurs de thèse en pagaille, une grande et belle famille, changeante, diverse, mais à laquelle il est impossible de ne pas s'attacher sincèrement. Un peu de travail avant beaucoup de bières, un programme de qualité, partagé avec les habitués du Friday Beer que sont Pilou, mon co-équipier touch-rugbyman préféré, Anton et son plus beau turban, Alasdair la meilleure des recrues britanniques pour notre top 14 académique, Pierre et Anaïs (mais pas trop New-

ton, Carabistouille vaincra !), Alexandre premier *manifesteur* de France, suivi de près par Cambyse. Comment faire à présent pour mentionner Claire, l'âme du MAP5, et Vincent, LaTeX-maniac devant l'éternel, à la hauteur de leur génialité ? Je peux le dire maintenant, parfois j'aurais aimé être digne du bureau des imagistes. Mais c'était assez rare, parce que j'avais déjà autour de moi une équipe de choc. Des *oh Dio !*, *Madonna !*, *ca**o !*, des cris et des siestes, con i miei cari italiani preferiti : Andrea, Alessandro e Marta. Avec Sinda, Léo, Antoine, avec Ousmane Sackoooo, mon poto des rythmes de sommeil incompréhensibles. Et que dire de ma colocataire de bureau préférée, Juliana, désolé pour le Brésil mais maintenant qu'on l'a récupérée, on la garde ! Je ne pouvais pas non plus oublier Vivien, le meilleur des acolytes de conf', ainsi que Yen, Allan, Safa, Fabien, Florian, Alkeos, Nourra, et Warith, pour ta sympathie et ton sourire permanent (et non je ne mentionnerai même pas tes légendaires irruptions inopinées, parce qu'à vrai dire, je les aime bien). Je finirai par évidemment mentionner l'inimitable Rémi L., qui n'a même pas besoin d'un mot pour m'arracher un fou rire, et d'ailleurs tous les Rémi, et tous les Antoine, et aussi tous les bébés doctorants qui prennent la relève et que j'espère apprendre à mieux connaitre pendant cette demi-année qu'il me reste à passer dans ce temple de la bonne humeur.

Mes derniers mots iront à ceux qui, même s'ils ont peut-être vu la partie académique de cette aventure de plus loin, auront été indispensables à mon bien-être quotidien (et probablement à ma santé mentale). Je pense à ceux avec qui j'ai partagé des cours de maths mémorables en terminale, Jordan, Antoine, Joséphine, parce qu'on n'y aurait pas forcément cru à l'époque, mais que ça commence à faire un joli CV cumulé ! A Clément et Anna, mes premiers compagnons du monde des 'vraies' maths, et au CRI qui a pris la suite et bien plus depuis : Julie, Florian, Adime, Coralie, Anne et Noémie, toujours présents et toujours au top maintenant qu'on est devenu grands. A Marie, pour sa présence et son soutien dans les toutes dernières mais difficiles étapes du chemin. J'adresse également une pensée chaleureuse à ma famille *côté Pinçon* dans son ensemble. A ma sœur chérie, à mon beauf préféré, mes neveux et nièces d'amour, à mes grands-parents adorés, mes cousins et cousines débilement géniaux, et mes tontons et tatas que je kiffe tout autant, merci du fond du cœur. Si j'ai pu mettre autant d'investissement dans ce travail, c'est en grande partie grâce à vous, à votre soutien, à la décompression immédiate que je ressens en vous voyant, et à l'exemple que vous êtes pour moi, chacun à votre façon. J'avais promis de la concision (oui, je sais, c'est raté dans les grandes largeurs) et cela devient vraiment très difficile au moment d'évoquer la place occupée par mon père dans cette aventure. Si j'ai bien compris une chose avec le temps, c'est que dans la vie, il n'y a pas de problèmes. Enfin, du moment qu'il y a suffisamment de réseau pour pouvoir appeler papa. Par bien des aspects, je remarque son influence au travers des nombreuses expériences qui ont ponctué cette thèse, et j'espère qu'il ressentira également la fierté que j'ai souvent éprouvé à être son fiston. Au rang des personnes sans qui cette thèse n'aurait pas existé, tu mérites amplement, Lucie, une place sur le podium. Il y aurait encore une fois beaucoup à dire, mais avant tout merci, sincèrement, pour ton soutien sans faille, dans les bons comme dans les moments difficiles, qui n'ont pas manqué ces dernières années. J'espère que tu as bien profité de ces quelques mois pendant lesquels tu auras été docteure et pas moi, car la fête touche à sa fin ! Je tiens également à adresser une pensée chaleureuse à l'ensemble de la famille Bernard.

Et à la fin... je ne sais pas si c'est l'Allemagne qui gagne, mais je sais sans la moindre hésitation qui sont ceux qui restent. Mes acolytes de toujours, mes confidents quotidiens, Matthieu et Valentin, les constantes fondamentales de mon (petit) univers. (Qui ne manqueront pas, j'espère, de me chambrer au plus vite sur mes envolées lyriques). Et évidemment à

tous ceux qui les entourent et sont devenus, au fil du temps, des familles d'adoption : à Lola, Julie, Christophe, Madou, Eric, Armelle, Didier, Mathilde, Anne, Victoire, et bien d'autres, un grand merci à tous !

Exercice rare et précieux que celui de remercier toutes celles et ceux qui nous entourent et le méritent, mais à qui on ne prend pas toujours le temps de le dire. Pour celles et ceux dont le temps est malheureusement écoulé, je réserve mes mots en pensées.

A la mémoire de ma mère, ce manuscrit lui est dédié.

Contributions

Contexte

L'étude des données fonctionnelles constitue un thème important des recherches en statistique des dernières décennies. De nombreuses observations venant du monde réel peuvent être vues comme intrinsèquement fonctionnelles dès lors qu'elles dépendent du temps, ou de tout autre continuum. Parmi les domaines d'application, celui encore largement inexploité des sciences du sport fournit de nombreuses données et problématiques qui s'inscrivent dans ce cadre. La collaboration avec des fédérations sportives, notamment la Fédération Française de Natation (FFN) que l'on remercie, autour de la problématique de la détection des jeunes sportifs prometteurs est à l'origine de cette thèse, et les données associées constituent le fil rouge illustratif des méthodologies présentées dans ce manuscrit. Des études récentes (Boccia et al., 2017 ; Kearney and Hayes, 2018) sur les carrières de nombreux sportifs suggèrent une faible relation entre le niveau de performance dans les jeunes années et celui à l'âge adulte. En vue d'éclairer la problématique de l'identification des jeunes talents, notre travail s'intéresse principalement à deux questions : Existe-t-il des profils typiques de progression pour les sportifs ? Est-il possible d'utiliser ces éventuelles similarités entre individus pour améliorer des prédictions de la performance future ?

Les données présentées sont issues d'un recueil rétrospectif des performances en compétition des licenciés de la FFN. Pour une épreuve donnée, la fonction représentative du niveau d'un sportif au cours du temps est appelée *courbe de performance* et constitue notre principal objet d'étude. Les observations ponctuelles de cette courbe étant fournies par les résultats en compétition, notre jeu de données regroupe un ensemble de séries temporelles observées irrégulièrement d'un individu à l'autre. La relative parcimonie et l'irrégularité des données observées, illustrées sur la Figure 1, ont constitué les principaux moteurs des développements méthodologiques proposés dans cette thèse. En effet, les problématiques classiques d'apprentissage supervisé et non-supervisé dans un tel contexte souffrent d'un manque de modélisations pertinentes disponibles, notamment dans un cadre probabiliste. Après un état de l'art des méthodes existantes et utiles à nos développements (chapitre 1), les trois principaux chapitres de la thèse aspirent à proposer des réponses d'efficacité croissante à nos enjeux. Le chapitre 2 offre une première exploration du jeu de données et met en évidence la présence de profils de performance à travers une application de méthodes de clustering de courbes. Ensuite, nous développons dans le chapitre 3 un nouveau modèle de processus Gaussiens multi-tâches pour la régression, ainsi que l'algorithme d'apprentissage et les formules de prédictions associées, fournissant une modélisation probabiliste adaptée et des performances supérieures à l'état de l'art. Enfin, reprenant l'idée des structures de groupe introduite au chapitre 1, le chapitre 4 propose une généralisation du modèle précédent à l'aide d'un mélange de processus Gaussiens multi-tâches, permettant d'effectuer des

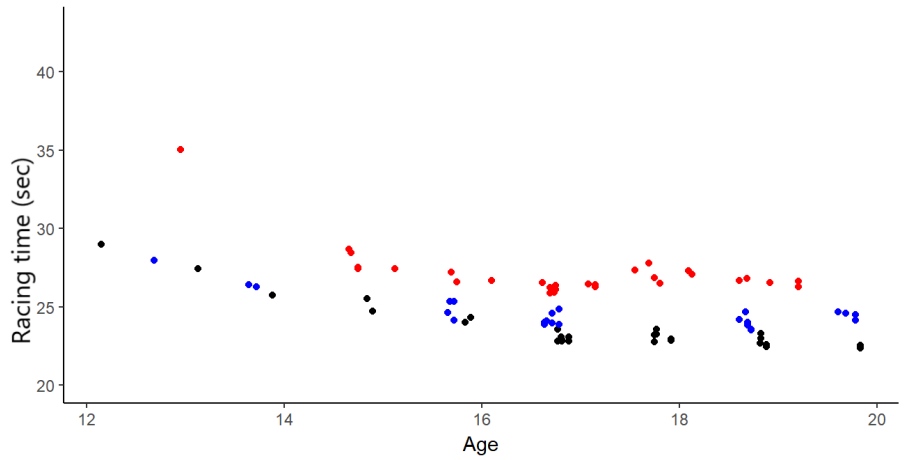

Figure 1 – Exemple de données observées pour 3 nageurs différents (respectivement en bleu, rouge et noir). L'age des abscisses indique l'âge du nageur. L'axe des ordonnées indique la performance en course en secondes, ici pour des compétitions de 50m nage libre homme.

prédictions cluster-spécifiques pondérées.

Clustering de courbes de progression de nageurs

Dans la lignée de la revue des différentes méthodes de clustering de courbes existantes, une étude comparative sur données synthétiques de plusieurs algorithmes, regroupés dans le package R *funcy*, est proposée en introduction. La première modélisation de notre exemple fil rouge se fait en deux étapes. Pour parer à la problématique des séries temporelles observées irrégulièrement, nous décomposons premièrement nos données dans une même base de B-splines, définissant ainsi des données fonctionnelles comparables d'un individu à l'autre. Ainsi, les méthodes classiques d'analyse de données fonctionnelles (FDA, pour Functional Data Analysis en anglais) peuvent être appliquées et nous avons choisi l'algorithme *funHDDC* (Bouveyron and Jacques, 2011 ; Schmutz et al., 2018) pour ses performances et la possibilité d'étudier des fonctions multidimensionnelles. En effet, un clustering utilisant simplement les coefficients des B-splines comme variable d'entrée fournit des groupes peu informatifs, surtout représentatifs de la position des courbes les unes par rapport aux autres sur l'axe des ordonnées. Une idée intéressante, comme souvent en FDA, consiste à utiliser les dérivées des courbes de progression comme variable supplémentaire pour apporter de l'information sur les dynamiques d'évolution. Nous montrons que cette approche apporte une plus-value et un regroupement fidèle à ce que les experts des fédérations observent en pratique. Nous identifions en particulier des patterns de progressions plus ou moins tardifs, permettant par exemple de rattraper un retard initial dans les performances des plus jeunes années (voir Figure 2). Cette approche, bien qu'ayant mis en évidence la présence de structures de groupes dans les données, souffre de plusieurs faiblesses de modélisation. D'une part, le peu de données disponibles pour certains individus complique le cadre paramétrique global de la décomposition B-splines, menant à des modélisations individuelles parfois insatisfaisantes. D'autre part, cette approche fréquentiste n'offre pas de quantification de l'incertitude, pour la modélisation et/ou la prédiction, qui serait pourtant précieuse dans ce type de problème d'aide à la décision. Autant d'obstacles menant aux développements méthodologiques au

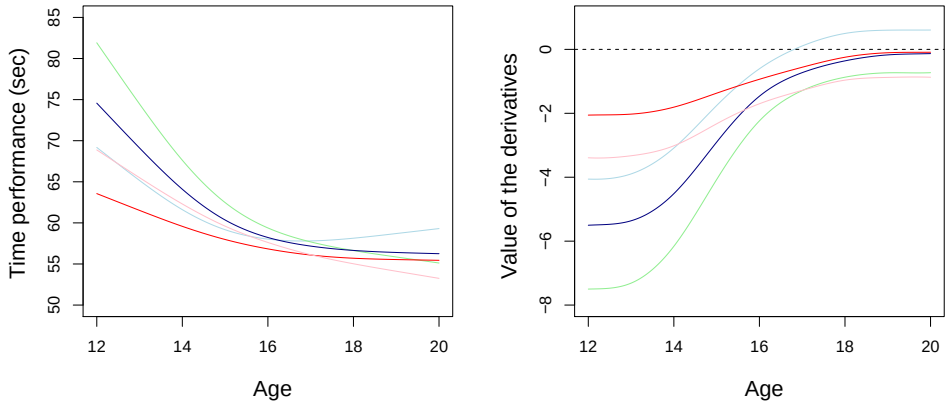

Figure 2 – Courbes moyennes (gauche) et dérivées moyennes (droite) issues du clustering en 5 groupes des courbes de performance de nageurs français entre 12 et 20 ans pour le 100m nage libre masculin.

coeur de cette thèse, qui prennent place dans le cadre probabiliste non-paramétrique des processus Gaussiens (GPs, pour Gaussian processes en anglais).

Processus Gaussiens multi-tâches avec processus moyen partagé

Le cadre des GPs offre une modélisation élégante pour les données fonctionelles, mais souffre toutefois de limitations lorsque les points d'observations sont peu nombreux et/ou mal répartis sur le domaine d'étude. Notre jeu de données étant composé de nombreux individus ($\simeq 10^4$) ayant chacun peu d'observations ($\simeq 10^1$), la définition d'un modèle multi-tâche autorisant le partage d'informations entre individus permet de tirer le meilleur parti de cette situation. L'originalité de l'approche repose sur l'introduction d'un processus moyen, commun à tous les individus, qui fournit une valeur a priori pour la prédiction, embarquant de l'information sur tout le domaine d'étude. Pour un individu i, la donnée fonctionnelle $y_i(t)$ est supposée générée par le modèle suivant :

$$y_i(t) = \mu_0(t) + f_i(t) + \epsilon_i(t), \ \forall i, \forall t,$$

avec μ_0 le GP moyen commun à tous, f_i un GP centré spécifique à l'individu i, et ϵ_i un terme de bruit. A l'aide de données, l'inférence de ce modèle consiste alors à estimer les hyper-paramètres des différents noyaux de covariance associés à ces GPs, et à calculer la loi hyper-posterior du processus μ_0. Ces quantités étant interdépendantes, nous dérivons un algorithme Espérance-Maximisation (EM)(voir Algorithm 2) qui est utilisé pour les estimer alternativement. Ensuite, nous établissons les Proposition 3.4 et Proposition 3.5 permettant de déduire des formules de prédiction GP exactes, qui intègrent à la fois l'information du processus moyen et son incertitude dans la loi a posteriori finale. Cette loi prédictive multi-tâche étend la pertinence de la modélisation GP sur un large domaine d'étude, même en l'absence d'observations individuelles, comme le montre la Figure 3. Le partage d'information entre individus, à travers le processus μ_0, s'avère également efficace dans le cadre des courbes de nageurs, sur lesquelles les performances prédictives sont très satisfaisantes. L'algorithme

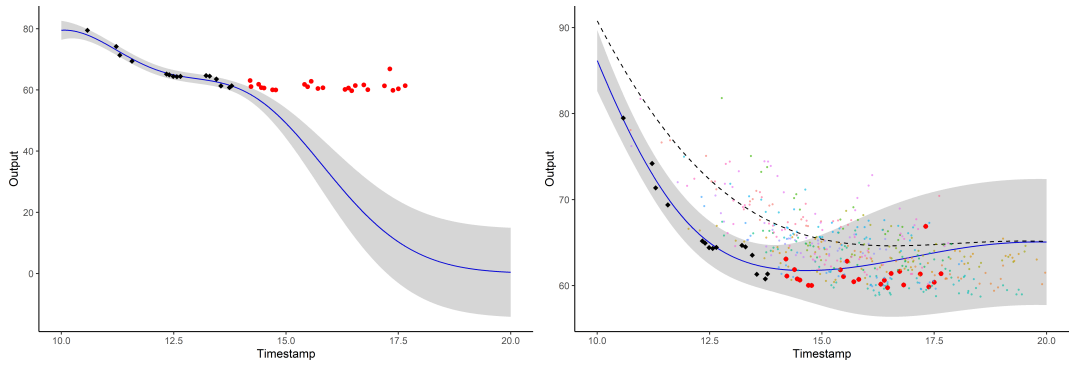

Figure 3 – Courbe de prédiction (bleu) et intervalle de crédibilité à 95% (gris) d'un nouvel individu pour la régression GP classique (gauche) et pour MAGMA (droite). Le processus moyen μ_0 est représenté en ligne brisée, les points d'observation en noir, et les points de test en rouge. Les points colorés en arrière plan illustrent les d'observations des individus ayant servi pour l'entrainement du modèle.

implémentant cette méthode, appelé MAGMA (pour Multi-tAsk Gaussian processes with common MeAn), présente de meilleurs résultats et fournit un nouveau cadre d'application plus général que les alternatives.

Clustering de courbes et prédiction groupe-spécifique de GPs multi-tâche

Reprenant l'idée d'une possible structure de groupe dans les données, une généralisation du modèle précédent à l'aide d'un mélange de GPs est ensuite proposée. En effet, pour certains jeux de données, l'hypothèse d'un unique processus central sous-jacent peut être trop restrictive. Ainsi, le modèle génératif se définit à présent comme suit :

$$y_i(t) = \mu_k(t) + f_i(t) + \epsilon_i(t), \ \forall i, \forall t, \tag{1}$$

avec $\mu_k(t)$ le GP moyen spécifique au k-ème groupe, alors que $f_i(t)$ et $\epsilon_i(t)$ restent, respectivement, le GP et le bruit spécifique à l'individu i. Ce nouveau modèle dépend également d'une variable multinomiale latente Z_i, contrôlant l'appartenance des individus à chaque cluster. Dans cette approche, il est à présent nécessaire d'estimer les hyper-paramètres des noyaux de covariance, conjointement des lois hyper-posterior des processus μ_k et des variables Z_i. Les dépendances a posteriori entre ces dernières quantités nous forcent maintenant à introduire un algorithme Variationnel EM (VEM) (voir Algorithm 3) pour l'inférence. Nous dérivons les lois variationnelles approximées dans les propositions Proposition 4.1 et Proposition 4.2, permettant leur utilisation ultérieure dans de nouvelles formules de prédiction GP. Un algorithme EM est également établi pour estimer les hyper-paramètres associés à un nouvel individu, partiellement observé, ainsi que ses probabilités d'appartenance aux différents clusters. Par intégrations successives sur les processus moyens μ_k (Proposition 4.4), puis sur les Z_i (Proposition 4.5), une loi de mélange Gaussien multi-tâche peut à nouveau être déduite, définie comme une somme pondérée de prédictions GP cluster-spécifiques. L'algorithme associé est appelé MAGMACLUST et fournit une implémentation complète de cette méthode. Nous illustrons au travers de simulations l'intérêt d'une telle approche et sa supériorité lorsque les données présentent des structures de groupes. Finalement, nous apportons une conclusion aboutie à la problématique des courbes de progressions des nageurs (Figure 4), leur clustering, et la prédiction probabiliste des performances futures. Cette approche re-

groupe ainsi les différents aspects balayés durant cette thèse, fournissant à la fois une réponse satisfaisante aux attentes applicatives initiales, ainsi qu'un apport méthodologique notable pour étudier des problèmes connexes.

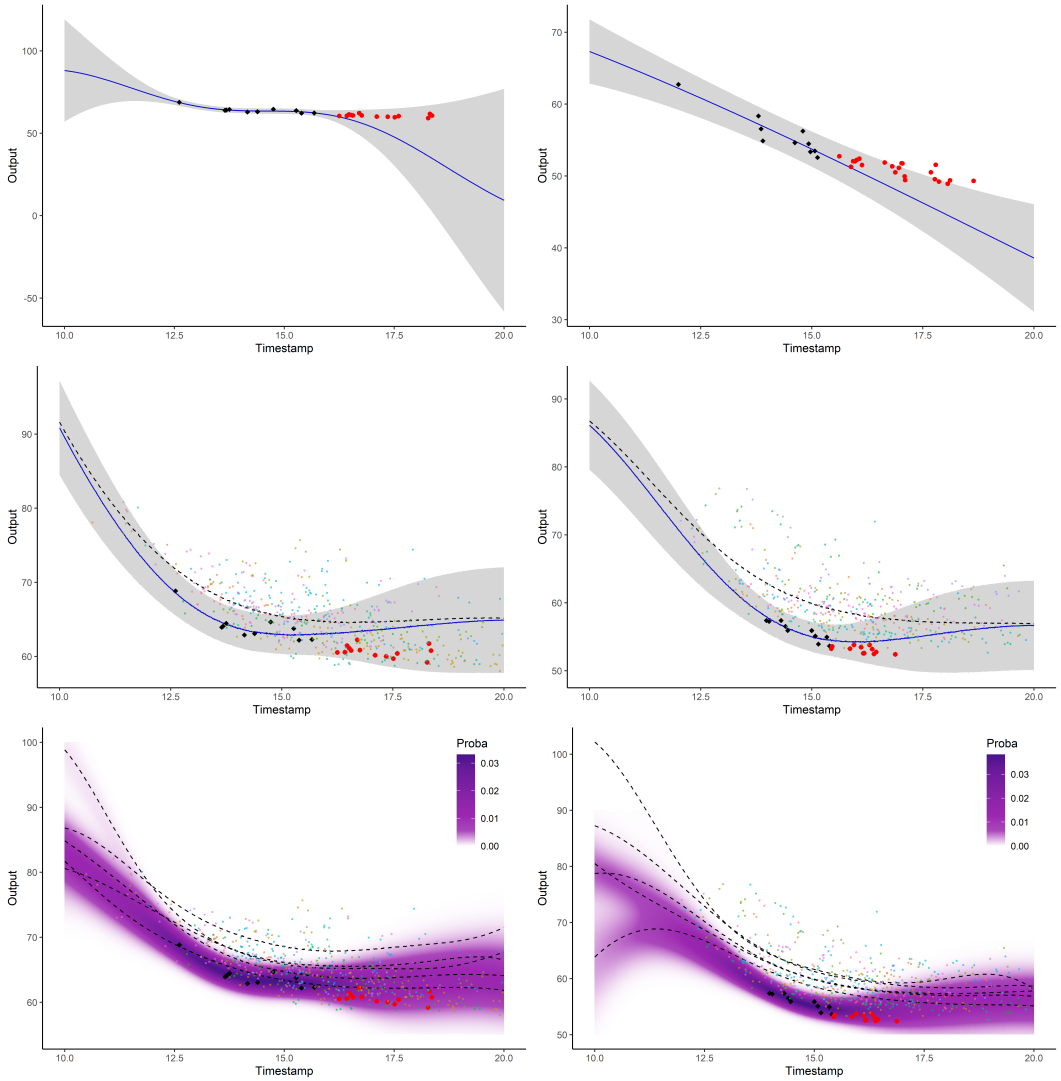

Figure 4 – Gauche : données femmes. **Droite :** données hommes. Distribution prédictive de la courbe de performance d'un nageur pris au hasard, obtenue par régression GP (haut), MAGMA (milieu), et MAGMACLUST (bas). Les processus moyens sont représentés en lignes brisées, les points d'observation en noir, et les points de test en rouge. Les points colorés en arrière plan illustrent les d'observations des individus ayant servi pour l'entrainement du modèle.

Articles publiés et prépublications

Le travail présenté dans ce manuscrit a donné lieu à une publication (Leroy et al., 2018) (chapitre 2), ainsi qu'à deux articles en cours de révision (Leroy et al., 2020b,a) (chapitre 3

et 4). En parallèle, durant cette thèse, deux articles ont été co-écrits (Moussa et al., 2019 ; Pla et al., 2019) sur des thématiques de sciences du sport, disjointes du présent document. La liste de ces publications est détaillée ci-dessous :

- A. Leroy, A. Marc, O. Dupas, J. L. Rey, and S. Gey. Functional Data Analysis in Sport Science : Example of Swimmers' Progression Curves Clustering. *Applied Sciences*, 8 (10) :1766, Oct. 2018. doi : 10.3390/app8101766

- A. Leroy, P. Latouche, B. Guedj, and S. Gey. MAGMA : Inference and Prediction with Multi-Task Gaussian Processes. *PREPRINT arXiv:2007.10731 [cs, stat]*, July 2020b

- A. Leroy, P. Latouche, B. Guedj, and S. Gey. Cluster-Specific Predictions with Multi-Task Gaussian Processes. *PREPRINT arXiv:2011.07866 [cs, LG]*, Nov. 2020a

- I. Moussa, A. Leroy, G. Sauliere, J. Schipman, J.-F. Toussaint, and A. Sedeaud. Robust Exponential Decreasing Index (REDI) : Adaptive and robust method for computing cumulated workload. *BMJ Open Sport & Exercise Medicine*, 5(1) :e000573, Oct. 2019. ISSN 2055-7647. doi : 10.1136/bmjsem-2019-000573

- R. Pla, A. Leroy, R. Massal, M. Bellami, F. Kaillani, P. Hellard, J.-F. Toussaint, and A. Sedeaud. Bayesian approach to quantify morphological impact on performance in international elite freestyle swimming. *BMJ Open Sport & Exercise Medicine*, 5(1) : e000543, Oct. 2019. ISSN 2055-7647. doi : 10.1136/bmjsem-2019-000543

Implementations

Les algorithmes décris dans les chapitres 3 et 4 ont été implémentés en packages R, qui constituent la contribution pratique de cette thèse. Une version courante de ces codes peut être librement trouvée aux adresses suivantes :

- MAGMA : `https://github.com/ArthurLeroy/MAGMA`,
- MAGMACLUST : `https://github.com/ArthurLeroy/MAGMAclust`.

Contents

1

Introduction

1.1 State of the art

1.1.1 Functional data analysis and sports science applications

The purpose of this section is twofold: at first, the sports science literature provides references to introduce the talent detection problem in sports as well as an overview of the longitudinal data studies in this field. Secondly, we present different aspects of the functional data analysis (FDA) and some of the associated classical methods, along with a brief review of the curve clustering state-of-the-art.

1.1.1.a Talent detection in sports

In the elite sport context, a classical problem lies in the detection of promising young athletes (Johnston et al., 2018). With professionalisation and evolution of training methods, differences in competitions became tighter and tighter in recent years (Berthelot et al., 2015). Besides, it has been shown (Moesch et al., 2011) that the development of some specific abilities during adolescence is a key component of improvement. Hence, many sports federations or structures have paid interest in the subject and tried to understand the mechanisms behind what could be called *talent* (Vaeyens et al., 2008), and its evolution during the young years of a career. A key feature to take into account is morphology since it obviously influences performance in many sports (Mohamed et al., 2009; Pla et al., 2019). Morphology is also known as a major noise factor of the talent detection paradigm since the differences in physical maturity lead to promote some young athletes over others (Goto et al., 2018) just because of their temporary advantages in height and/or weight. Well-known problems occur when these maturity rhythms are ignored, such as in training centres, with an over-representation of athletes born during the first months of the year (Wattie et al., 2015). Moreover, it appeared in several studies (Boccia et al., 2017) that performance at young ages provides in itself a poor predictor of the future competition results. Only a small portion of elite athletes before 16 years old remains at a top level of performance later (Kearney and Hayes, 2018). Thereby, it seems clear that the classical strategy, which consists in training intensively in specific structures only the best performers of a young age range, reaches its limits. Although there are numerous elements that influence performance (Vaeyens et al., 2009), several works (Ericsson et al., 2018) seem to indicate that the evolution of an athlete over time is more suited to predict future abilities than raw values at given ages. Different patterns of progression may exist, and it might be important to take them into account if one wants to improve the quality of talent detection strategies. Our work in this context aims at providing a more global vision of the progression phenomenon by saving its genuine continuous nature. Therefore, modelling performance data as functions over time and study them as such might offer new perspectives and provide insights to sport structures for their future decisions.

1.1.1.b Longitudinal data in sports

For a long time, sports science has been interested in time-dependent phenomenons. If at first, people only kept track of performance records, there is currently a massive amount of various available data. Among them, one specific type is generally called *time series* or *longitudinal data*. Many of the data recorded and studied in sports science nowadays can be considered as time series depending on the context. From the heart rate during a sprint race (Lima-Borges et al., 2018), the number of injuries in a team over a season (Carey

et al., 2018), to the evolution of performances during a whole career (Boccia et al., 2017), the common ground remains the evolution of a characteristic regarding a time period. An interesting property of such data lies in the dependency between two observations at two different instants, leading to the fact that the independent and identically distributed (iid) hypotheses are generally not verified. However, most of the usual statistical tools classically used in sports science, such as the law of large number or central limit theorem, need these properties*. Thus, all the statistical methods based on these results (hypothesis testing, method of moments, ...) collapse, and specific tools are required to study time series. There is a whole literature related to this subject, and we defer to the monograph Brockwell and Davis (2013) for details. These methods focus on the study of time-dependent processes that generate discrete observations. For instance, since a recurrent topic in this manuscript concerns clustering, a really comprehensive review about the clustering of time series can be found in Warren Liao (2005).

Despite the usefulness of the time series approach, new modellings have been proposed (de Boor, 1972) for longitudinal data. In many cases, the studied phenomenon is actually changing continuously over time. Thus, the object we want to know about is generally more of a function than a series of points. Moreover, the authors in Carey et al. (2018) highlight that it may be damageable to discretise phenomenons that are intrinsically functional. They claim that continuous methods perform better than discrete ones on the specific case of the relationship between training load and injury in sports.

In some particular cases, it thus seems natural to model a continuous phenomenon as a random function of time, formally a stochastic process, and consider our observations as just a few records of an infinite-dimensional object. The field of functional data analysis (FDA) then gives a new range of well-suited methods to work on longitudinal data. There has been substantial theoretical improvements in this area for the past two decades, and some of the tools have been successfully applied to sports science problems. We can cite Forrester and Townend (2015) in which curve clustering is used to analyse the foot-strike of runners or Mallor et al. (2010) for a thorough FDA on the muscle fatigue. Another example is given by Helwig et al. (2016) that proposes a functional version of ANOVA using splines to overcome common issues that occur in sports medicine. Finally, the work presented in Liebl et al. (2014) uses curve clustering methods to study different types of footfall in running. The methodology used in this paper is closely related to the one we present in Chapter 2, and the authors claim that this approach improved analysis of footfall compared to former empirical and observational ways to classify runners. We exhibit in the following sections that FDA can be used to handle some of the questions we deal with in this thesis such as: How to study the evolution of swimmers' performances from their racing times in competition? Competitors probably participate to different numbers of races during their careers, and their performances are recorded at different ages, leading to difficulties when it comes to comparing them, keeping in mind that discretisation has been shown problematic in Wattie et al. (2015).

If FDA remains marginally applied in the sports field, many examples can be found in a wide range of other domains. We can cite for instance meteorology, with the article of Ramsay and Dalzell (1991) that describes the study of temperatures among Canadian weather stations, which has become a classic dataset over the years. Another famous dataset

*Note that there exist several versions of these theorems with more or less flexible hypotheses, depending on the context. We talk here about the most common versions, classically used in applied science.

is presented in Gasser et al. (1984) as an application to biology by studying the growth of children as a time-continuous phenomenon. Those works and datasets are today considered as benchmarks to test new methods, but many fields such as economy (Liebl, 2013), energy (Bouveyron et al., 2018), medicine (Shen et al., 2017) or astronomy (Velasco Herrera et al., 2017) have used FDA and contribute to this active research topic.

1.1.1.c Functional data analysis

Although we previously mentioned several real-life phenomenons behaving over a continuum, one may fairly argue that there are no such things as infinite-dimensional data in practice. In reality, even in the case of functional data, the actual observations come as a set of finite-dimensional objects. Whereas models and methods developed for functional data sometimes resemble those of the conventional multivariate case, the underlying functions that generated the observations are assumed to be smooth at a certain degree. Without this smoothness property no real gain could be expected from this approach, and speaking rather loosely, a functional data can be thought of as a multivariate data with order on its dimensions.

We have seen that FDA allows us to take into account the intrinsic nature of functional data, but apart from this philosophical advantage in terms of modelling, it also provides answers to recurrent problems in some datasets. For example, how to compare time series observed on irregular grids of measurement? This question occurs in the following chapters, where we develop several strategies to tackle this issue. Another fundamental advantage of FDA lies in the ability to work on the derivatives of the observed functions since it is often interesting to look at the dynamic of time-dependent processes. Even the second derivative, often referred to as *acceleration*, or superior order derivatives might provide valuable information in practice. As time is often the continuum over which functional data are observed, it becomes usual in the following that we refer to the input variables as *timestamps*. However, different continua might be involved, such as spatial, frequency, position, and so forth.

The first and fundamental step of a functional data analysis generally lies in the reconstruction of the functional signal from a discrete set of observations. Let us assume to collect a set of data points $\mathbf{y} = \{y^1, \dots, y^N\}$ observed at timestamps $\mathbf{t} = \{t_1, \dots, t_N\}$ coming from a model of the type:

$$y^j = x(t_j) + \epsilon^j, \ \forall j = 1, \dots, N,$$

where $x(\cdot)$ represents some functional relationship between input and output variables, and ϵ^j is a noise term (due to genuine uncertainty, measurement error, ...). In this case, we can proceed to a prior *smoothing* step, which consists in the reconstruction of a functional signal supposed to be close to the observed points. The most common way to define a function from the data points is to use basis expansion. A basis of functions is a set $\{\phi_1, \dots, \phi_B\}$ coming from a functional space $\mathcal{S}$, such as each element of $\mathcal{S}$ can be defined as a linear combination of the $\{\phi_b\}_{b=1,\dots,B}$. Formally, we can define the basis expansion of $x(t)$, $\forall t \in \mathcal{T}$, as:

$$\begin{aligned} x(t) &= \sum_{b=1}^{B} \alpha_b \ \phi_b(t) \\ &= \boldsymbol{\alpha}^\intercal \boldsymbol{\phi}, \end{aligned}$$

where $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_B)$ is a vector of real valued coefficients, and $\boldsymbol{\phi}$ is the vector of basis functions evaluated at timestamp t. If we assume to observe $(\mathbf{y}, \mathbf{t})$ and fix a basis of functions,

we may derive a least squares estimation (LSE) for $\boldsymbol{\alpha}$ by minimising:

$$LSE(\mathbf{y} \mid \boldsymbol{\alpha}) = \sum_{j=1}^{N} \left[y^j - \sum_{b=1}^{B} \alpha_b \phi_b \left(t_j \right) \right]^2 \tag{1.1}$$

$$= (\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\alpha})^\intercal (\mathbf{y} - \boldsymbol{\Phi}\boldsymbol{\alpha}), \tag{1.2}$$

where $\boldsymbol{\Phi}$ is the $N \times B$ matrix containing the values $\phi_b(t_j), \forall b, \forall j$. If we assume Gaussian white noise residuals, then the estimate $\hat{\boldsymbol{\alpha}}$ is given by:

$$\hat{\boldsymbol{\alpha}} = \left(\boldsymbol{\Phi}^\intercal \boldsymbol{\Phi} \right)^{-1} \boldsymbol{\Phi}^\intercal \mathbf{y},$$

and we can derive the vector of fitted values:

$$\hat{\mathbf{y}} = \boldsymbol{\Phi}\hat{\boldsymbol{\alpha}} = \boldsymbol{\Phi} \left(\boldsymbol{\Phi}^\intercal \boldsymbol{\Phi} \right)^{-1} \boldsymbol{\Phi}^\intercal \mathbf{y}.$$

Several other solutions may be derived for more sophisticated assumptions on the residuals or the model. In the presence of noisy data, the over-fitting/under-fitting issues may generally occur. According to the smoothness we expect for the resulting curves, it can be well advised to use *regularisation* through the introduction of a penalty term in (1.1). Thereby, a smoothing parameter has to be determined to define how regular the resulting functions are assumed, which might be estimated with cross-validation techniques, for instance. Although defining a consistent value for the smoothing parameter is an initial task in itself, it also enables to control the signal-on-noise ratio of the data explicitly. This topic being far beyond the scope of the present thesis, we refer to Ramsay and Silverman (2005, Chapter 4&5) for details. In the case where we collect M different sources (*individuals* or *batches*) of functional data $\{\mathbf{y}_1, \ldots, \mathbf{y}_M\}$, observed on a common grid of timestamps $\mathbf{t} = \{t_1, \ldots, t_N\}$, we have the overall formulation for the associated $M \times N$ matrix $\mathbf{x}(\mathbf{t})$:

$$\mathbf{x}(\mathbf{t}) = \begin{pmatrix} x_1(t_1) & \cdots & x_1(t_N) \\ \vdots & \ddots & \vdots \\ x_M(t_1) & \cdots & x_M(t_N) \end{pmatrix} = \begin{pmatrix} \alpha_{11} & \cdots & \alpha_{1B} \\ \vdots & \ddots & \vdots \\ \alpha_{M1} & \cdots & \alpha_{MB} \end{pmatrix} \begin{pmatrix} \phi_1(t_1) & \cdots & \phi_1(t_N) \\ \vdots & \ddots & \vdots \\ \phi_B(t_1) & \cdots & \phi_B(t_N) \end{pmatrix},$$

and we may still derive analytical solutions to estimate the matrix of coefficients through straightforward linear algebra. Intuitively, when a common basis $\boldsymbol{\Phi}$ is fixed to fit multiple functions, the information that is specific to the i-th individual is contained in the vector of coefficients $\{\alpha_{i1} \ldots, \alpha_{iB}\}$, hence providing a parametric representation for infinite dimensional objects. Therefore, a naive approach consists in performing classical multivariate methods directly on these coefficients (Abraham et al., 2003). Among the most common basis used in practice, we can cite Fourier basis and wavelets, which are well suited to model periodic data (Ramsay and Silverman, 2002). Fourier basis is a widespread choice that works well when data present a repetitive pattern (such as day/night cycles for example) since the basis functions are sinusoids. However, their efficiency decreases when data are less regular, especially on the modelling of derivatives. Wavelet basis are designed to settle this sensibility to irregular signals. Although coefficients are slightly longer to compute, this basis has really nice mathematical properties and progressively replaced Fourier basis in several applications (Giacofci et al., 2013; Ramsay et al., 2009). For non-periodic data, a somehow classical choice is to use spline basis (de Boor, 1972) in practice. For example, B-splines are piecewise polynomial functions that require few coefficients to define a good

approximation, which makes them especially suited when observations are sparse on the input domain. B-splines allows the approximation of a wide range of shapes with rather good smoothness. From a practical point of view, the comprehensive R package *fda* contains methods to fit observations into functional data and way more tools for FDA. An overview of the *fda* package can be found in Ramsay and Silverman (2002). Once the functional signal has been reconstructed for the data, some classical statistical tools have been extended in the functional context to pursue analysis. One of the first and crucial method that has been adapted in the early literature on FDA (Rao, 1958; Tucker, 1958) was the functional principal component analysis (FPCA).

Functional PCA

Several reasons are explaining the central role that FPCA plays when it comes to analysing functional data. This method provides a precious explanatory tool for detecting the main features and modes of variation in a dataset as well as a dimensionality reduction technique. Moreover, as for the covariance matrix in the multivariate standard case, the covariance functions characterising functional variables remain difficult to interpret, and FPCA aims at expressing data as a linear combination of uncorrelated functions. If we consider an $\mathbf{L}^2$ stochastic process $X(t), t \in \mathcal{T}$, its mean function is defined as:

$$\mu(t) = \mathbb{E}(X(t)),$$

which can be estimated in practice by averaging over a set of M observations of the process

$$\hat{\mu}(t) = \frac{1}{M} \sum_{i=1}^{M} x_i(t), \ \forall t \in \{t_1, \ldots, t_N\},$$

if all the functional data are observed on a dense regular grid (one may first use basis expansion otherwise). Moreover, we can express the covariance function as:

$$\mathrm{Cov}(X(s), X(t)) = \mathbb{E}[(X(s) - \mu(s))(X(t) - \mu(t))],$$

which can be naturally estimated with empirical data as well. From the Karhunen-Loève theorem (Loève, 1946; Karhunen, 1947), we can express the centred process as an eigenfunction expansion:

$$X(t) - \mu(t) = \sum_{q=1}^{\infty} \xi_q \varphi_q(t),$$

with

$$\xi_q = \int_{\mathcal{T}} (X(t) - \mu(t)) \varphi_q(t) \, \mathrm{d}t,$$

where $\{\varphi_q\}_{q \geq 1}$ are the orthonormal eigenfunctions of the autocovariance operator, associated with the non-negative and decreasing eigenvalues $\{\lambda_q\}_{q \geq 1}$. The random variables $\{\xi_q\}_{q \geq 1}$, called *principal component scores*, are also centred and $\mathbb{E}[\xi_q \xi_l] = \delta_{ql} \lambda_q$. Note that the FPCA leads to the best empirical basis expansion for functional data in terms of mean integrated square error. Moreover, each eigenfunction φ_q is supposed to represent the main mode of variation, under the constraint to be uncorrelated to the previous ones:

$$\varphi_q = \underset{\|\varphi\|=1, \langle \varphi, \varphi_j \rangle = 0, j=1,\ldots,q-1}{\arg\max} \left\{ \mathbb{V}\left(\int_{\mathcal{T}} (X(t) - \mu(t)) \varphi(t) dt \right) \right\}.$$

In practice, a truncation of the Karhunen-Loève expansion is used to extract the p most important modes of variation associated with the Q greatest eigenvalues:

$$X(t) \approx X_Q(t) = \mu(t) + \sum_{q=1}^{Q} \xi_q \varphi_q(t).$$

Thereby, the remaining terms of the sum are assumed to be negligible as it corresponds to low informative features with eigenvalues usually close to 0. This approximation is used in many other methods of FDA since it may be considered as the most parsimonious way to correctly represent a functional data with a given number of basis functions. Let us stress that, although the variables in multivariate PCA can be permuted without effect on the analysis, the order in functional data matters, and their underlying smoothness allows FPCA to still perform well in high dimensions context. Many other standard statistical tools have also been adapted to the functional context such as *functional canonical correlation*, *discriminant analysis* and *functional linear models* (Ramsay and Silverman, 2005). For the sake of completeness, let us mention another non-linear generalisation that has been proposed for PCA, namely the principal curves (Hastie and Stuetzle, 1989), which can even be learnt sequentially from data streams (Guedj and Li, 2019).

Non-parametric FDA

So far, we have presented approaches generally relying on the assumption that we can characterise infinite-dimensional functions with a set of relevant parameters. However, the field of non-parametric statistics has tried for a long time to weaken such an assumption, and the approach focusing on distribution-free or parameters-free methods has been adapted to the functional case. According to Ferraty and Vieu (2006), a functional non-parametric model consists in the introduction of constraints on the form of infinite-dimensional processes that cannot be indexed by a finite number of elements. This framework is sometimes called *double infinite-dimensional* because the infinite aspect occurs both from the functional and the non-parametric context. Many notions and methods developed in this case lie on the key notion of semi-metrics. Contrarily to the finite-dimensional norms that are all equivalent, this property disappears in the functional framework, and the choice of the preliminary norm becomes crucial. As some of the non-parametric methods make use of usual heuristics, although replacing the classical metrics by an adequate functional notion of closeness, we present below three different semi-norms and their scope of use. First, the PCA-type semi-norm is defined by using the FPCA expansion presented before on a centred functional data $x(t)$, for which we only retain the Q first eigenfunctions. Formally:

$$\|x\|_Q^{PCA} = \sqrt{\sum_{q=1}^{Q} \left(\int x(t)\varphi_q(t)\,\mathrm{d}t \right)^2}.$$

We can naturally deduce the associated semi-metric:

$$\mathcal{D}_Q^{PCA}(x_1, x_2) = \sqrt{\sum_{q=1}^{Q} \left(\int [x_1(t) - x_2(t)]\varphi_q(t)\,\mathrm{d}t \right)^2}.$$

This kind of semi-metric is expected to provide interesting results when studying rough datasets. Secondly, the partial least square (PLS) semi-metrics (Wold, 1966) rely on the same kind of definition, except we need to use output variables as well to define it. By computing a

decomposition with p components that maximises the covariance between input and output variables, we can again use it to define a semi-norm based on the L^2 norm. The PLS semi-metric that is constructed from it is generally recommended in the case of multivariate output variables. Finally, the derivatives-type uses, as indicated by its name, the derivatives of an arbitrary order $u \in \mathbb{N}$ to define a semi-norm and the corresponding semi-metric as such:

$$\mathcal{D}^{(u)}(x_1, x_2) = \int \left(x_1^{(u)}(t) - x_2^{(u)}(t) \right)^2 \mathrm{d}t.$$

This type is usually well adapted for relatively smooth datasets. On another aspect, the notion of small ball probabilities (Delaigle and Hall, 2010) also plays a major role in the definition and practical use of statistical objects such as mean, median, quantiles as well as their conditional counterpart. The quantities also lie at the centre of the non-parametric regression and clustering methods. From this notion, we can also introduce the concept of kernel local smoothing in the functional case. As often in non-parametric statistics, kernels also play a central role in non-parametric FDA, and a paragraph is dedicated to a more thorough inspection of these objects in the sequel.

We previously mentioned that an interesting property arises in the functional framework with the possible cancelling of the curse of dimensionality. This particularity only happens if the correlation between the points on the curve is important enough. Contrarily to the multidimensional framework, and even in the case of many data points, an important smoothness of the underlying function indicates a behaviour that can be considered as almost unidimensional (because naturally constrained). For weak correlations though, or functional data with too abrupt leaps, the curse of dimensionality reappears. It is also important to note that the non-parametric framework remains poorly adapted to treat the case of irregular measurements in the input space. In particular, most methods fall down in the case of sparse datasets, which makes this approach difficult to use for the applicative problems in our scope.

1.1.1.d Clustering functional data

Families of curve clustering methods

In this section, we focus on the *clustering* problem, which provides insights about the eventual group structures in a functional dataset or may serve as a starting point to more elaborate analyses. Non-supervised learning in the functional case focuses on the definition of sub-groups of curves that make sense according to an appropriate measure of similarity. Given K the number of clusters, a clustering algorithm would apply one or several rules to allocate the functional data presenting common properties into the same group. This problem has been largely explored for the past years in the functional context, and a few elements summarising the state-of-the-art are provided in this paragraph. According to the survey Jacques and Preda (2014), functional data clustering algorithms can be sorted into three distinct families, detailed below. We do not develop on direct clustering on raw data points that does not take into account the functional nature of the data and usually leads to poor results.

(i) *2-steps methods.* The first step aims at fitting functions from data as detailed previously, by choosing a common basis for all curves. Then, a clustering algorithm, such as k-means (Abraham et al., 2003) for instance, is applied on the basis coefficients. If this vector of observations is high-dimensional, we can proceed first to an FPCA step, before

using the clustering method on the scores coming from the first eigenfunctions of the FPCA.

(*ii*) *Non-parametric clustering.* As previously discussed, some approaches of curve clustering can be adapted in the context of non-parametric FDA (Ferraty and Vieu, 2006). From the notions of functional semi-metrics and small-ball probabilities, we can define a notion of heterogeneity within a set of functions. By using such a quantity, a descending hierarchical heuristic can be defined by considering a *parent* set of functions and build splitting scores based on the gain or loss of heterogeneity in the resulting split sets. Another heuristic that has been adapted is k-means. The authors in Cuesta-Albertos and Fraiman (2007) propose a closeness notion based on the trimmed mean for function data, whereas derivative-based distances are used in Ieva et al. (2013) instead.

(*iii*) *Model-based clustering.* As the 2-step approach, model-based methods often use basis expansion and/or FPCA to fit the data. However, rather than proceeding in two steps, the clustering procedure is performed simultaneously. As we apply a model-based clustering algorithm in our exploratory study in Chapter 2, more details on this subject are provided in a subsequent dedicated paragraph.

Note that the literature does not provide specific indications about the family of methods that should be used according to the context. Nevertheless, one should keep in mind that the appropriate way to handle functions often depends on the nature of the original data. Additional references are provided in Chapter 2, where we present and compare several usual curve clustering algorithms implemented in the *R* package *funcy*. Below, Figure 1.1, inspired by a representation from Jacques and Preda (2014), summarizes these different families and the clustering process in a functional context.

Model-based curve clustering

As briefly mentioned above, model-based clustering aims at defining probabilistic techniques to deal with the appropriate grouping of functional data. The representation of a functional data still relies on finite-dimensional coefficients that can be obtained through a basis of functions expansion or from the resulting scores of an FPCA. However, those coefficients are now considered as realisations of random variables instead of as simple parameters. In such context, it becomes possible to make assumptions on the probability distribution they are sampled from, and in particular, cluster-specific distributions can be defined. For example, as proposed in James and Sugar (2003), we can assume that a spline basis coefficient α_k comes from a Gaussian distribution, specific to the k-th cluster:

$$\alpha_k \sim \mathcal{N}(\mu_k, \Sigma),$$

where μ_k is a cluster-specific mean and Σ a common variance for all groups. Such an assumption implies an underlying Gaussian mixture model, and the inference procedures then rely on the simultaneous estimation of the basis coefficients and the proportions of the mixture. The use of spline basis remains convenient for smooth data, although wavelet-based approaches have also been adapted (Giacofci et al., 2013) when a wider range of functional shapes are needed.

On the other hand, Delaigle and Hall (2010) introduced an approximation of the notion of probability density for random functions, based on the truncated FPCA expansion and the density of the resulting principal components. A similar approximation is used in Bouveyron and Jacques (2011) to establish a mixture model where principal components are supposed

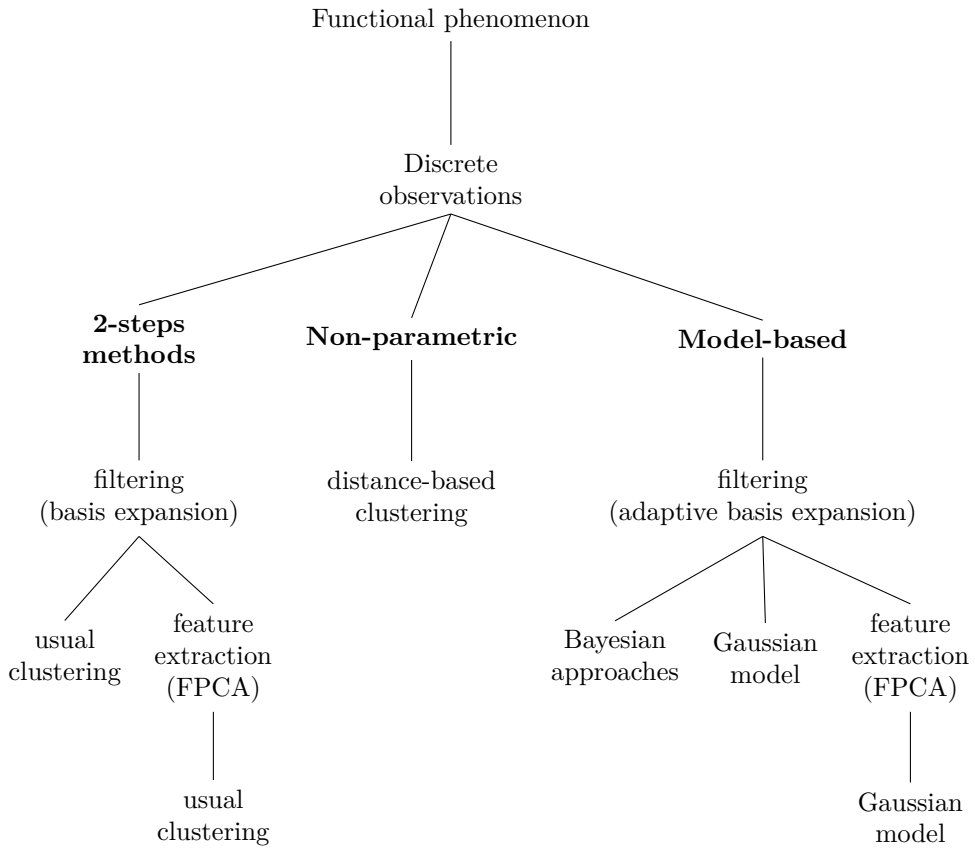

Figure 1.1 – Summary of the different approaches to perform clustering on functional data, from raw data (top) to final clusters (bottom).

to come from a Gaussian distribution. The idea relies on the application of an FPCA to each cluster separately and the derivation of an EM algorithm (see Section 1.1.3.a) to alternatively compute the component scores and the proportion of the mixture. Adapting ideas previously introduced in the multivariate context (Bouveyron et al., 2007) to functional data, the authors propose different parsimonious assumptions on the variance structure, resulting in various sub-models. The overall clustering algorithm is called *funHDDC*, and this method is applied for a preliminary analysis of the sports dataset used as illustrative example in this thesis (see Section 2.3). Another way to introduce some parsimony in the model is suggested in Jacques and Preda (2013b) by defining a truncation order for the Karhunen-Loève expansion that is cluster-specific. Finally, an extension of *funHDDC* to the case of multivariate functional data have been described in Schmutz et al. (2018). In this approach, the practical choice on which sub-models to use is handled by model selection, thanks to criteria such as *BIC* (Schwarz, 1978) or the *slope heuristic* (Birgé and Massart, 2006; Arlot, 2019).

Bayesian FDA

Let us finally raise recent theoretical advances on the matter of online clustering (Li et al., 2018) that make use of another perspective, namely, generalised Bayesian learning

algorithms and the PAC-Bayesian framework (Guedj, 2019). Besides, Chapter 3 and Chapter 4 focus on multi-task time series forecasting, which may present connections to the study of multiple curves proposed in FDA. Contrarily to the approaches described above, we aim at proposing a probabilistic framework in these chapters. Some Bayesian FDA methods have been developed in Thompson and Rosen (2008) or Crainiceanu and Goldsmith (2010), which inspired some aspects our models. Moreover, as initially proposed in Rice and Silverman (1991), it remains possible to model and learn mean and covariance structures simultaneously from a non-parametric probabilistic point of view, even when data are curves. Such an approach leads to the introduction of a popular object in the machine learning field: Gaussian processes (GPs), which are at the core of discussions in the subsequent section.

1.1.2 Gaussian processes

In statistics, the classical *supervised learning* problem implies the estimation from data of an underlying function f that maps an input x onto an output y such as:

$$y = f(x).$$

This learning problem is called *inductive*, as we infer on the value of f from observed data, in order to make predictions on y values for any new input x_* afterwards. In practice, it seems vain to expect this relation to be verified exactly in real-world observations (because of genuine randomness or measurement errors for example), and the introduction of an additive noise term would indicate that we seek an appropriate approximation. Besides, the infinite number of candidates offering a solution to this problem generally leads to consider additional hypotheses on the class of functions we should pick from. Although we account for another approach in the subsequent paragraphs, a classical assumption involves the restriction to a reasonable class of functions that constrains the form of f. A usual trade-off then occurs, since we expect enough flexibility to provide a good fit of the data whereas avoiding to run into the danger of overfitting, which deteriorates the generalisation performance. Among the many forms we may consider, the linear model often offers a straightforward choice by assuming that the output value changes proportionally to the input. If one could fairly argue that the linear model remains too restrictive, it only serves here as a convenient motivation to introduce Gaussian processes as an answer to the regression problem (we purposefully set aside the case of classification, see Rasmussen and Williams (2006, Chapter 3) for details).

1.1.2.a From linear model to Gaussian Processes

Let us assume that $x \in \mathbb{R}^d$ and $y \in \mathbb{R}$. Defining a linear model implies to express the relationship between theses variables through the following equation:

$$y = \beta^\intercal x + \epsilon,$$

where β would typically be a d-dimensional real-valued vector in a frequentist context. However, from a Bayesian point-of-view, we shall assume a prior distribution over β, like an isotropic Gaussian $\beta \mid \alpha \sim \mathcal{N}\left(0, \alpha^{-1} I_d\right)$ for example. Moreover, let $\epsilon \mid \sigma^2 \sim \mathcal{N}\left(0, \sigma^2\right)$ be a Gaussian white noise as well. In this case, when provided with an iid training sample of data $\mathcal{D} = \{(y_1, x_1), \ldots, (y_N, x_N)\}$, let us define $\mathbf{y}$ the corresponding output vector and $\mathbf{X}$ the $N \times d$ design matrix, such as:

$$\mathbf{y} = \mathbf{X}\beta + \boldsymbol{\epsilon},$$

where $\boldsymbol{\epsilon} \mid \sigma^2 \sim \mathcal{N}\left(0, \sigma^2 I_N\right)$. If we assume β and $\boldsymbol{\epsilon}$ to be independent, $\mathbf{y}$ is defined as a linear combination of Gaussian variables and thus remains Gaussian. Therefore, we only need to specify its mean parameter:

$$\mathbb{E}\left[\, \mathbf{y} \mid \mathbf{X}, \sigma^2, \alpha \,\right] = \mathbf{X}\mathbb{E}\left[\, \beta \mid \alpha \,\right] + \mathbb{E}\left[\, \boldsymbol{\epsilon} \mid \sigma^2 \,\right] = 0,$$

as well as its covariance matrix:

$$\begin{aligned}
\mathbb{V}\left[\, \mathbf{y} \mid \mathbf{X}, \sigma^2, \alpha \,\right] &= \mathbb{E}\left[\, \mathbf{y}\mathbf{y}^\intercal \mid \mathbf{X}, \sigma^2, \alpha \,\right] \\
&= \mathbb{E}\left[\, \mathbf{X}\beta\beta^\intercal\mathbf{X}^\intercal + \mathbf{X}\beta\boldsymbol{\epsilon}^\intercal + \boldsymbol{\epsilon}\beta^\intercal\mathbf{X}^\intercal + \boldsymbol{\epsilon}\boldsymbol{\epsilon}^\intercal \mid \mathbf{X}, \sigma^2, \alpha \,\right] \\
&= \mathbf{X}\mathbb{E}\left[\, \beta\beta^\intercal \mid \alpha \,\right]\mathbf{X}^\intercal + \mathbb{E}\left[\, \boldsymbol{\epsilon}\boldsymbol{\epsilon}^\intercal \mid \sigma^2 \,\right] \\
&= \frac{\mathbf{X}\mathbf{X}^\intercal}{\alpha} + \sigma^2 I_N.
\end{aligned}$$

Hence, the conditional likelihood of the model is given by:

$$p\left(\mathbf{y} \mid \mathbf{X}, \sigma^2, \alpha\right) = \mathcal{N}\left(\mathbf{y}; 0, \boldsymbol{K} + \sigma^2 I_N\right), \tag{1.3}$$

where we purposefully define $\boldsymbol{K} = \frac{1}{\alpha}\mathbf{X}\mathbf{X}^\intercal$ to be a $N \times N$ matrix, which elements are $\left[\, \boldsymbol{K} \,\right]_{uv} = k(x_u, x_v) = \frac{1}{\alpha} x_u^\intercal x_v, \ \forall u, v = 1, \dots, N$. Hence, while the observations are assumed to be iid, integrating over β induces a dependence structure between the predictions. This conditional integrated likelihood can be used to learn the (hyper-)parameters of the model, whether by defining adequate priors and computing their posterior distributions in a fully Bayesian treatment or with an *empirical Bayes* approach by directly maximising (1.3) (more details in Section 1.1.2.c). As we shall demonstrate further on, this model defines a first and particular example of Gaussian process regression. Letting this aspect aside, let us look more closely at the function $k(\cdot, \cdot)$ defined above, which only depends on the inner product between the observed inputs. We have seen that the linear model often remains too restrictive in the form of relationship it defines. However, let us recall that Section 1.1.1.c introduced the expansion of a non-linear function as a linear combination of basis functions. This representation can be seen as a linear model in the space of the basis functions $\phi(x)$, i.e. the (often) higher-dimensional space on which the input variables x are projected by the map $\phi(\cdot)$. Hence, we can derive the same model as previously except $\boldsymbol{\Phi}(\mathbf{X})$ substitutes $\mathbf{X}$ everywhere, and thus imply non-linear relationships between x and y. In practice though, the explicit choice of an adequate mapping $\phi(\cdot)$ remains complicated. However, we can notice that the mapping only appears in the linear model through the calculus of an inner product $\phi(x_u)^\intercal\phi(x_v)$ in the corresponding space. Fortunately, this quantity might be expressed as a function of the original input points $k_\phi(x_u, x_v) = \phi(x_u)^\intercal\phi(x_v)$, namely a *kernel function*. For the identity mapping $\phi(x) = x$ and $\alpha = 1$, we would retrieve $k_\phi(x_u, x_v) = k(x_u, x_v) = x_u^\intercal x_v$, which we shall refer to as a *linear kernel*. Conversely, by using a more elaborate kernel function and thus specifying a different covariance structure in (1.3), we would imply a transformation of the input representation leading to non-linear interactions with the output. This approach is known as the *kernel trick* (Aiserman et al., 1964) and constitutes the key idea behind *kernel methods*, for which the Gaussian process regression is a particular example.

1.1.2.b Kernel methods

First of all, we were able to define $k_\phi(\cdot, \cdot)$ thanks to the Mercer theorem (Mercer and Forsyth, 1909), which states that a positive-definite kernel is a symmetric function that can be ex-

pressed as the inner product between the evaluations of a fixed mapping $\phi(\cdot)$ at points $x, x' \in \mathbb{R}^N$. Roughly speaking, a kernel is often seen as a way to measure the similarity between two inputs (or a notion of closeness), although it more exactly computes the similarity between two evaluations of a map. Moreover, the $N \times N$ Gram matrix (simply called *covariance matrix* in the GP context) $\mathbf{K}_\phi$, associated with the kernel $k_\phi(\cdot, \cdot)$, must be positive semi-definite (Shawe-Taylor and Cristianini, 2004) as well. The term *kernel methods* then refers to a class of algorithms making use of those objects to solve learning tasks. Among the most well-known kernel methods, we can cite support vector machines (SVM)(Boser et al., 1992; Cortes and Vapnik, 1995) that allow, inter alia, the definition of non-linear frontiers in classification problems. Using kernel functions enables to work implicitly in high-dimensional feature spaces without explicitly computing the coordinate of data in that space.

There exists a variety of different kernels, regrouped in families (Genton, 2002) according to the properties they imply on the method using them. First, we call *stationary* a kernel that is defined as a function of $x - x'$. This type of kernel is then invariant to translation and only depends on the lag separating two inputs. Furthermore, if we now consider a function of $|x - x'|$, the kernel becomes *isotropic* and remains insensitive to motions in any direction. Conversely, the name *nonstationary* refers to the most general class of kernels, which depend explicitly on the values of the inputs. In addition to the interaction between inputs, most of the kernels depend upon a set of parameters that specify their exact forms. In the context of GPs, those are often called hyper-parameters, since they are related to a distribution over functions instead of a function directly. Let us give the examples of a few classical kernels we may encounter, especially when it comes to defining a GP model. In the first place, we already saw an example of *linear kernels* that can be expressed as such:

$$k_{Lin}(x, x') = \sigma_s^2 (x - c)(x' - c),$$

with $c \in \mathbb{R}^N$, and $\sigma_s^2 \in \mathbb{R}$ a scale factor that appears in every kernel. The linear kernel is non-stationary, thus the value of the inputs matters even if its parameters are kept fixed. This kernel is often used in combination with others to indicate an increasing or decreasing tendency in the data. A second example can be seen with the *periodic kernel*, which is isotropic and generally serves to represent phenomenons with repeating structures:

$$k_{Per}(x, x') = \sigma_s^2 \exp\left(-\frac{2}{\ell^2} \sin^2\left(\pi \frac{|x - x'|}{\omega}\right)\right),$$

where ω indicates the period between two repetitions of the pattern, and ℓ is a lengthscale parameter that controls the smoothness of the kernel. Among the most common choices of kernel for GPs or SVM lies the *exponentiated quadratic* (EQ) (sometimes called *squared exponential* or *Gaussian kernel*):

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

where the parameters are the same as previously. This isotropic kernel presents nice properties and is well-suited for interpolating smooth functions since it is infinitely differentiable. If the EQ has somehow become a go-to choice for many applications, some authors (see Stein (1999) for instance) argue that such smoothness assumption might be unrealistic for modelling many real-life phenomenons and recommend the following Matérn class. The

Matérn kernel (Matérn, 2013) can be seen as the generalisation of the previous EQ and is defined as:

$$k_{Matern}(x, x') = \sigma_s^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu} \frac{|x - x'|}{\ell} \right)^{\nu} K_\nu \left(\sqrt{2\nu} \frac{|x - x'|}{\ell} \right),$$

where Γ is the gamma function, K_ν is the modified Bessel function of the second kind, and ν and ℓ are non-negative parameters controlling the smoothness of the kernel. The Matérn kernel, which converges to an EQ when $\nu \to \infty$, provides a flexible choice that has been proved useful for many applications (Minasny and McBratney, 2005; Yang et al., 2016).

Once we know the characteristics of some simple kernels, we may take advantage of several convenient properties for combining them into more elaborate ones. Let us assume $k_1(x, x')$ and $k_2(x, x')$ to be valid kernels. Then the following kernels are also valid (Bishop, 2006, Chapter 6):

$$\begin{aligned}
k(\mathbf{x}, \mathbf{x}') &= c k_1(\mathbf{x}, \mathbf{x}'), \\
k(\mathbf{x}, \mathbf{x}') &= g(\mathbf{x}) k_1(\mathbf{x}, \mathbf{x}') g(\mathbf{x}'), \\
k(\mathbf{x}, \mathbf{x}') &= q(k_1(\mathbf{x}, \mathbf{x}')), \\
k(\mathbf{x}, \mathbf{x}') &= \exp(k_1(\mathbf{x}, \mathbf{x}')), \\
k(\mathbf{x}, \mathbf{x}') &= k_1(\mathbf{x}, \mathbf{x}') + k_2(\mathbf{x}, \mathbf{x}'), \\
k(\mathbf{x}, \mathbf{x}') &= k_1(\mathbf{x}, \mathbf{x}') k_2(\mathbf{x}, \mathbf{x}'), \\
k(\mathbf{x}, \mathbf{x}') &= k_3(\phi(\mathbf{x}), \phi(\mathbf{x}')), \\
k(\mathbf{x}, \mathbf{x}') &= \mathbf{x}^{\mathrm{T}} \mathbf{A} \mathbf{x}',
\end{aligned}$$

with $c \in \mathbb{R}^+$, $g(\cdot)$ an arbitrary function, $q(\cdot) \in \mathbb{R}[X]$ a polynomial, $k_3(\cdot, \cdot)$ a valid kernel of the features space, and $\mathbf{A}$ a symmetric positive semi-definite matrix. As presented above, there exists a large variety of operations that can be used to manipulate kernels and enhance the set of desired properties for our model. However, the specific effect of those operations on the resulting kernels constitutes a topic that is beyond the scope of the present thesis. For a comprehensive review on the subject along with automated methods to construct them, we can refer to Duvenaud (2014). By specifying an appropriate kernel for a GP, it is possible to derive a variety of models such as linear regression, splines or Kalman filters, those just being the most commons among a wide choice. In the sequel, the terms *kernel*, *covariance function*, *kernel function*, or *covariance kernel* refers all to the same object when working in context of GPs.

1.1.2.c Gaussian process regression

Nonparametric probabilistic framework

According to Rasmussen and Williams (2006), a Gaussian process can be defined as a collection of random variables (indexed over a continuum), any finite number of which having a joint Gaussian distribution. Hence, a GP provides a generalisation to the notion of multivariate Gaussian distribution in the infinite-dimensional context. Although GPs have been introduced (Wiener, 1949) and studied (Thompson, 1956; Matheron, 1973; Cressie, 1993) for quite a long time, they regained much interested with the development of Bayesian

learning methods for regression problems (O'Hagan, 1978; Williams and Rasmussen, 1996). We have previously seen that the class in which the regression function f belongs often needs to be restricted by hypotheses on its form. However, there is an alternative to this strategy where we keep the functional space large enough while favouring some properties in a probabilistic fashion. Roughly speaking, assuming a prior distribution over f would specify an occurrence probability to every admissible choice of functions. Thereby, we could attribute higher probabilities to functions that we consider more likely to occur in the context of the modelling. While appealing, this approach seems unrealistic in the first place, considering the uncountable infinite set of functions to deal with. Gaussian processes, however, offer an elegant framework that combines both a distribution over a functional space and a modelling that remains tractable. Although we would only work in practice with finite numbers of evaluations of the function f, the properties that are deduced on the full process are the same whether we consider the infinitely remaining input points or not (they are implicitly integrated out). Let us assume $t \in \mathcal{T}$, with $\mathcal{T} \subset \mathbb{R}$ (we still use a temporal vocabulary for convenience though the following remains true for other input spaces such as $\mathcal{T} = \mathbb{R}^d$ for instance). Saying that f is a Gaussian process, i.e. $f(\cdot) \sim \mathcal{GP}\left(\mu(\cdot), k(\cdot,\cdot)\right)$, implies that its distribution is entirely specified through its mean function:

$$\mu(t) = \mathbb{E}\left[\, f(t) \,\right], \ \forall t \in \mathcal{T},$$

and its covariance function, expressed as a kernel:

$$k(t,t') = cov\left(f(t), f(t')\right) = \mathbb{E}\left[\, (f(t) - \mu(t))(f(t') - \mu(t')) \,\right], \ \forall t, t' \in \mathcal{T}.$$

Generally, the mean function is supposed to equal zero everywhere since it can simply be integrated into the kernel as an additional term. However, this quantity being of critical importance in our forthcoming contributions (see Chapters 3 and 4), we shall keep the term $\mu(t)$ explicit in the following expressions. By doing so, we emphasise on its role (or absence of role actually) in the classical approaches, which will serve our purpose later. Regardless of this remark, the mean function μ is at this stage a modelling choice, considered as fixed and known in the remainder of the section. On the other hand, most of the attention is usually paid to the covariance structure. The choice of the kernel determines a relationship between input values that induces the covariance between the outputs. Hence, setting an appropriate kernel is of major influence on the generalisation performances of the model, since the covariance structure controls the way we should extrapolate from new data. Once the parameters of the prior distribution are stated, we may derive the model likelihood that lies in the centre of the inference procedure.

Marginal prior distribution

First, let us recall the expression of the GP regression model:

$$y(t) = f(t) + \epsilon, \ t \in \mathcal{T},$$

where $f(\cdot) \sim \mathcal{GP}\left(\mu(\cdot), k(\cdot,\cdot)\right)$ is a GP as previously described, and $\epsilon \sim \mathcal{N}\left(0, \sigma^2\right)$ is a Gaussian white noise independent from f. If we consider the finite-dimensional evaluations of the output $\mathbf{y} = \{y(t_1), \ldots, y(t_N)\}$ and the GP $\mathbf{f} = \{f(t_1), \ldots, f(t_N)\}$ at timestamps $\mathbf{t} = \{t_1, \ldots, t_N\}$, the following conditional distribution can be derived:

$$p(\mathbf{y} \mid \mathbf{f}, \sigma^2) = \mathcal{N}\left(\mathbf{y}; \mathbf{f}, \sigma^2 I_N\right), \tag{1.4}$$

along with the finite-dimensional distribution over $\mathbf{f}$:

$$p(\mathbf{f} \mid \mathbf{t}) = \mathcal{N}(\mathbf{f}; \mu(\mathbf{t}), \mathbf{K}), \tag{1.5}$$

where $\mathbf{K}$ is the $N \times N$ Gram matrix associated with the kernel $k(\cdot, \cdot)$, and $\mu(\mathbf{t})$ the N-dimensional vector of the mean function, both being evaluated at the observed timestamps $\mathbf{t}$. Making use of (1.4) and (1.5), we can derive the marginal prior distribution $p(\mathbf{y} \mid \mathbf{t}, \sigma^2)$ explicitly (Bishop, 2006, Section 2.3.3) by integration over all possible $\mathbf{f}$:

$$\begin{aligned} p(\mathbf{y} \mid \mathbf{t}, \sigma^2) &= \int p(\mathbf{y}, \mathbf{f} \mid \mathbf{t}, \sigma^2) \, \mathrm{d}\mathbf{f} \\ &= \int p(\mathbf{y} \mid \mathbf{f}, \sigma^2) p(\mathbf{f} \mid \mathbf{t}) \, \mathrm{d}\mathbf{f} \\ &= \int \mathcal{N}\left(\mathbf{y}; \mathbf{f}, \sigma^2 I_N\right) \mathcal{N}(\mathbf{f}, \mu(\mathbf{t}), \mathbf{K}) \, \mathrm{d}\mathbf{f} \\ &= \mathcal{N}(\mathbf{y}; \mu(\mathbf{t}), \mathbf{\Psi}) \end{aligned}$$

where $\mathbf{\Psi} = \mathbf{K} + \sigma^2 I_N$. This distribution can be derived for any finite-dimensional vector $\mathbf{y}$, implying that $y(\cdot)$ is a GP as well, with a mean function $\mu(\cdot)$ and a covariance kernel $\psi(t, t') = k(t, t') + \delta_{t,t'}\sigma^2, \ \forall t, t' \in \mathcal{T}$. Speaking rather loosely, we notice that $\mathbf{f}$ has been 'replaced' by its mean parameter when integrated out. Moreover, this expression highlights that two independent Gaussian sources of randomness simply add to each other, and thus define a new covariance structure for the resulting GP. This kind of convenient marginalisation, with a closed-form in the Gaussian case, also serves us to derive one of our key result (Proposition 3.5) in Chapter 3.

Learning the hyper-parameters

Let us stress that we purposefully omitted to mention the parametric nature of the kernel $k(\cdot, \cdot)$ in the previous expressions for the sake of simplicity. However, although the form of the kernel is set as a modelling choice, we generally want to keep some fitting flexibility by learning kernel's hyper-parameters from data. Let us note θ the set of hyper-parameters that specify the kernel $k_\theta(\cdot, \cdot)$, which will be used in the current paragraph. The inference procedure in GP regression then requires the estimation of $\Theta = \{\theta, \sigma^2\}$. Let us assume now that we observe an N-dimensional dataset $(\mathbf{y}, \mathbf{t})$, defined as previously. From the marginal distribution, we may establish the likelihood of the model $p(\mathbf{y} \mid \mathbf{t}, \Theta)$ as a function of Θ. There exist several approaches to make use of this likelihood function for the learning. In a fully Bayesian treatment, we would need to introduce a prior over each element of Θ and use Bayes theorem to derive their posterior distributions. However, the normalisation term in the formula often remains intractable in practice, and we shall make use of approximations such as MCMC methods, which might be time-consuming. On the other hand, another hybrid approach called *empirical Bayes* allows us to derive point estimates through direct optimisation. Whether we seek a *maximum likelihood* or a *maximum a posteriori* estimate, we can maximise respectively the likelihood with respect to Θ or the posterior distributions. For the latter, we would still need to define prior distributions over the hyper-parameters, whereas the former provides a simpler approach on which we focus. In practice, we generally prefer to manipulate the log-likelihood $\mathcal{L}(\Theta; \mathbf{y}, \mathbf{t}) = \log p(\mathbf{y} \mid \mathbf{t}, \Theta)$, which is more convenient to handle (in numerical implementations as well since it avoids instability when dealing with small probabilities):

$$\mathcal{L}(\Theta; \mathbf{y}, \mathbf{t}) = -\frac{1}{2}\left(\log|\mathbf{\Psi}_\Theta| + N\log(2\pi) + (\mathbf{y}-\mu(\mathbf{t}))^\intercal \mathbf{\Psi}_\Theta^{-1}(\mathbf{y}-\mu(\mathbf{t}))\right). \tag{1.6}$$

To efficiently maximise this function, we would often take advantage of gradient-based algorithms, such as conjugate gradients (Hestenes and Stiefel, 1952) or BFGS methods (Nocedal, 1980), for instance. Hence, if we assume the evaluation of $\mathbf{\Psi}_\theta$'s derivatives to be straightforward (which is true for the kernels we presented before), we can simply express the log-likelihood derivatives as:

$$\frac{\mathrm{d}\mathcal{L}(\Theta; \mathbf{y}, \mathbf{t})}{\mathrm{d}\theta_i} = \frac{1}{2}\left((\mathbf{y}-\mu(\mathbf{t}))^\intercal \mathbf{\Psi}_\Theta^{-1}\frac{\mathrm{d}\mathbf{\Psi}_\Theta}{\mathrm{d}\theta_i}\mathbf{\Psi}_\Theta^{-1}(\mathbf{y}-\mu(\mathbf{t})) - \mathrm{tr}\left(\mathbf{\Psi}_\Theta^{-1}\frac{\mathrm{d}\mathbf{\Psi}_\Theta}{\mathrm{d}\theta_i}\right)\right), \ \forall\theta_i \in \Theta.$$

PREDICTION

In this paragraph, we consider that the hyper-parameters $\hat{\Theta}$ have been learned and we thus omit the dependencies on them for the sake of clarity. As usual in regression problems, we now observe a new input t_* (that could be a vector of timestamps as well), for which we want to predict the associated output y_*. Since $y(\cdot)$ is assumed to be a GP, its evaluation on the $(N+1)$-dimensional vector of timestamps $(\mathbf{t}, t_*)^\intercal$ remains a joint Gaussian distribution:

$$p\left(\mathbf{y}, y_* \mid \mathbf{t}, t_*\right) = \mathcal{N}\left(\begin{bmatrix}\mathbf{y}\\ y_*\end{bmatrix}; \begin{bmatrix}\mu(\mathbf{t})\\ \mu(t_*)\end{bmatrix}, \begin{pmatrix}\mathbf{\Psi} & \psi(t_*, \mathbf{t})\\ \psi(\mathbf{t}, t_*) & \psi(t_*, t_*)\end{pmatrix}\right)$$

where $\psi(t_*, \mathbf{t})$ denotes the column-vector of all kernel evaluations $\psi(t_*, t)$, $\forall t \in \mathbf{t}$. By displaying the expression in this detailed form, we can keep track of all elements that appear in the predictive distribution. It can be easily demonstrated (Bishop, 2006, Section 2.3.1) that if we condition y_* over the variables $\mathbf{y}$ that have been observed, the conditional remains Gaussian, thus providing a predictive distribution for y_*:

$$p(y_* \mid \mathbf{y}, \mathbf{t}, t_*) = \mathcal{N}\left(y_*; \hat{\mu}(t_*), \hat{\Psi}_*\right) \tag{1.7}$$

with:

- $\hat{\mu}(t_*) = \mu(t_*) + \psi(t_*, \mathbf{t})\mathbf{\Psi}^{-1}(\mathbf{y}-\mu(\mathbf{t}))$,
- $\hat{\Psi}_* = \psi(t_*, t_*) - \psi(t_*, \mathbf{t})\mathbf{\Psi}^{-1}\psi(\mathbf{t}, t_*)$.

As $\mathcal{D} = \{\mathbf{y}, \mathbf{t}, t_*\}$ constitutes the set of data that we assume observed, we can call (1.7) the posterior distribution of y_*. This quantity offers a thorough probabilistic prediction for the output value at timestamp t_*. In order to display this results more easily, we often extract a *maximum a posteriori* estimate $\hat{y}_*^{MAP} = \mathbb{E}\left[y_* \mid \mathcal{D}\right] = \hat{\mu}(t_*)$ along with credible intervals, constructed from the posterior variance $\mathbb{V}\left[y_* \mid \mathcal{D}\right] = \hat{\Psi}_*$. Let us stress that this result stands if t_* represents an arbitrary vector of timestamps, providing a multivariate Gaussian distribution as a prediction. In this case, we would even acquire a full covariance matrix at the target timestamps instead of simple point-wise variances. In practical applications, it is in common usage to compute the predictions on a fine grid for displaying purpose, as we propose on Figure 1.2. This comparison highlights the evolution of the GP modelling as we add information through data, and provides examples of functional realisations of the prior and posterior processes.

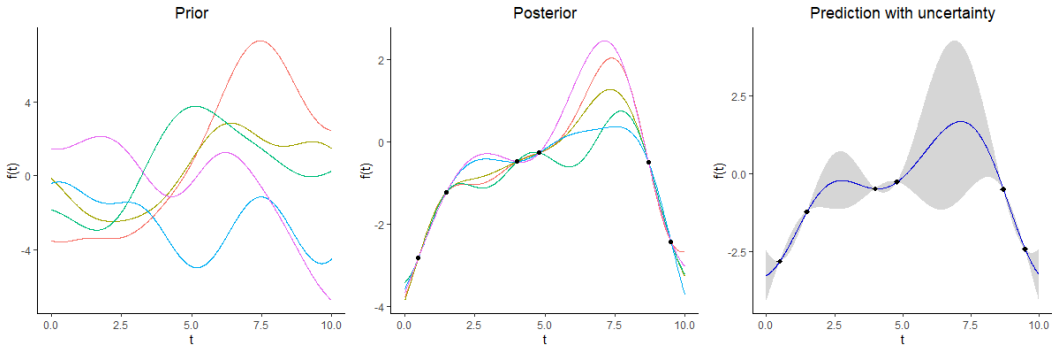

Figure 1.2 – Left: Example of curves sampled from the same GP prior distribution. **Middle**: Example of curves sampled from the associated posterior distribution with respect to the points displayed in black. **Right**: GP prediction (blue) with the associated 95% credible interval (grey band) coming from the posterior distribution

Let us give some additional intuitions on the parameters of this predictive distribution. We can note that the posterior mean $\hat{\mu}(t_*)$ is expressed as a linear combination of the data points $\mathbf{y}$, re-centred around the corresponding prior values of the mean function $\mu(\mathbf{t})$, added with the prior mean value at the target timestamps $\mu(t_*)$. Moreover, the weights of the linear combination appear to be proportional to the covariance between the observed and target outputs, through the kernel evaluations $\psi_{\hat{\Theta}}(t_*, \mathbf{t})$. Roughly speaking, this indicates that the closer a data point lies to the target, the higher it contributes to the prediction, and conversely. Such a behaviour remains quite intuitive and implies that a prediction far from observed data almost exclusively depends upon the prior mean $\mu(t_*)$. We would advise the reader to keep these aspects in mind for the subsequent chapters as it highlights the often underrated potential of specifying an adequate mean function for the unobserved timestamps. Regarding the covariance term, it still consists of a prior (co-)variance term $\psi_{\hat{\Theta}}(t_*, t_*)$, at which we subtract a positive quantity that decreases as we move away from data points. This means that the prediction uncertainty is gradually adapting to the distance between observed and target values, thus quantifying rather smartly the increasing difficulty in performing reliable long-term forecasts.

Advantages of Gaussian processes

The GP framework presents several useful properties, which makes it particularly well-suited to handle the regression problem. First, it offers an analytical procedure with a closed-form predictive posterior expressed as a Gaussian distribution. Moreover, GPs provide convenient building blocks for constructing more elaborate models. In the same way as for linear models, for which GPs can be seen as an extension (Rasmussen and Williams, 2006, Chapter 2), it seems easier in the first place to derive sophisticated methods from a tractable and well-understood skeleton. Furthermore, we can express a broad range of hypotheses (Duvenaud, 2014) solely through the definition of the covariance function. These assumptions being encoded in an adequate kernel with a usually limited number of parameters to estimate. This particularity reduces the requirement for regularisation schemes or complex optimisation and often allows us to avoid the pitfall of over-fitting. Nonetheless, this elegant GP framework and its modelling advantages also have a price when it comes to practical implementation.

1.1.2.d Limits and extensions

We exhibited in the previous paragraphs the convenient manner for deriving probabilistic predictions that made GP regression a popular method in the machine learning community. Despite their undeniable qualities, GPs also suffer from limitations that prevent wider practical applicability. As we previously mentioned, the various range of properties that GPs can model is strongly related to the choice of the kernel. This flexibility also raises the practical question of what kernel's form is the most adapted to each particular structure of data. This issue remains often addressed by human experts although automatic kernel construction methods have recently been proposed (Duvenaud, 2014; Gómez-Bombarelli et al., 2018).

Besides, as we may notice in (1.6), the learning stage in GP requires a matrix inversion that may lead to a computational bottleneck. This inversion of the $N \times N$ covariance matrix induces a $\mathcal{O}\left(N^3\right)$ complexity for learning, whereas the vector-matrix products in (1.7) at the prediction step necessitate $\mathcal{O}\left(N^2\right)$ operations. Hence, GPs scale up quickly with the number of data points, and we generally consider that it may only be applied directly to moderate data sets (up to 10^4 observations approximately). As this aspect constitutes the main obstacle to GP usage in modern applications, abundant literature has concentrated in the past two decades on the diverse ways to tackle this issue. A classical idea behind the sparse approximations of GPs rely on the introduction of *pseudo-inputs* $\mathbf{u} = \{u_1, \dots, u_n\}$ as latent evaluations of the GP at locations $\mathbf{t_u}$, which are assumed to be in number $n << N$. By this mean, the learning procedure only lies on a covariance matrix of dimension $n \times n$ and induces a computational cost reduced to $\mathcal{O}(Nn^2)$ (and $\mathcal{O}(n^2)$ for prediction). We only present here some ideas behind this approach and a few examples that have been the successive state-of-the-art on this question. For a more thorough analysis and comparisons we can refer to Rasmussen and Williams (2006, Chapter 8), Quiñonero-Candela and Rasmussen (2005), or Bauer et al. (2016).

Keeping the previous notation, let us yet rewrite the covariance matrices $\mathbf{K_{a,a}} = k(\mathbf{t_a}, \mathbf{t_a})$, with $\mathbf{a} \in \{\mathbf{f}, \mathbf{u}\}$ corresponding respectively to the vectors of observed and latent inputs. First note that we can always write the prior distribution $p\left(\mathbf{f}_*, \mathbf{f}\right)$ by integrating over the pseudo-inputs:

$$p\left(\mathbf{f}_*, \mathbf{f}\right) = \int p\left(\mathbf{f}_*, \mathbf{f}, \mathbf{u}\right) \mathrm{d}\mathbf{u} = \int p\left(\mathbf{f}_*, \mathbf{f} \mid \mathbf{u}\right) p(\mathbf{u}) \mathrm{d}\mathbf{u},$$

where $p(\mathbf{u}) = \mathcal{N}\left(\mathbf{0}, K_{\mathbf{u},\mathbf{u}}\right)$ by definition, for the any finite-dimensional evaluations of the GP. Then, we can make the approximation assumption that $\mathbf{f}_*$ and $\mathbf{f}$ are independent conditionally to $\mathbf{u}$, and thus derive an approximate joint posterior:

$$p\left(\mathbf{f}_*, \mathbf{f}\right) \simeq q\left(\mathbf{f}_*, \mathbf{f}\right) = \int q\left(\mathbf{f}_* \mid \mathbf{u}\right) q(\mathbf{f} \mid \mathbf{u}) p(\mathbf{u}) \mathrm{d}\mathbf{u}.$$

We often refer to $\mathbf{u}$ as the *inducing variables*, since all the dependencies between $\mathbf{f}$ and $\mathbf{f}_*$ are then induced through $\mathbf{u}$. Note that the location of those *pseudo-inputs* could be chosen randomly as a subset of the initial observations, but this approach is generally suboptimal compared to data-driven methods. We now present three of the principal approximations for $q(\cdot, \cdot)$ that have been proposed in the literature. Following the formalism introduced by Quiñonero-Candela and Rasmussen (2005), these methods can be expressed instead in regards to the approximation they provide for the exact conditional likelihood $p\left(\mathbf{y} \mid \mathbf{f}\right) = \mathcal{N}\left(\mathbf{y}; \mathbf{f}, \sigma^2 I_N\right)$. First, the deterministic training conditional (DTC) approximation (Seeger et al., 2003) is based on the projection $\mathbf{f} \simeq K_{\mathbf{f},\mathbf{u}} K_{\mathbf{u},\mathbf{u}}^{-1} \mathbf{u}$:

$$p(\mathbf{y} \mid \mathbf{f}) \simeq q(\mathbf{y} \mid \mathbf{u}) = \mathcal{N}\left(\mathbf{y}; K_{\mathbf{f},\mathbf{u}} K_{\mathbf{u},\mathbf{u}}^{-1} \mathbf{u}, \sigma^2 I_N\right).$$

Secondly, the fully independent training conditional (FITC) approximation (Csató and Opper, 2002; Snelson and Ghahramani, 2006) proposes a more elaborate expression of the covariance:

$$p(\mathbf{y} \mid \mathbf{f}) \simeq q(\mathbf{y} \mid \mathbf{u}) = \mathcal{N}\left(\mathbf{y}; K_{\mathbf{f},\mathbf{u}} K_{\mathbf{u},\mathbf{u}}^{-1} \mathbf{u}, \operatorname{diag}\left[K_{\mathbf{f},\mathbf{f}} - K_{\mathbf{f},\mathbf{u}} K_{\mathbf{u},\mathbf{u}}^{-1} K_{\mathbf{u},\mathbf{f}}\right] + \sigma^2 I_N\right).$$

Let us mention that there exist many different ways to come up with this method: from approximation of the conditional likelihood as above; as a result of the expectation-propagation algorithm; or as an initial modification of the prior over $\mathbf{f}$. Finally, the method that might probably be considered as the current state-of-the-art (Bauer et al., 2016) is called variational free energy (VFE) (Titsias, 2009). This approach jointly infers the inducing inputs along with the kernel hyper-parameters. VFE makes use of variational inference to derive and maximise a lower bound for the true marginal likelihood $p(\mathbf{y})$ (and then approximates the true posterior distribution):

$$\log p(\mathbf{y}) \geq \int q(\mathbf{f}, \mathbf{u}) \log \frac{p(\mathbf{y} \mid \mathbf{f}) p(\mathbf{u})}{q(\mathbf{u})} \mathrm{d}\mathbf{u}\mathrm{d}\mathbf{f},$$

where the variational distribution is assumed to factorize as such $q(\mathbf{f}, \mathbf{u}) = p(\mathbf{f} \mid \mathbf{u}) q(\mathbf{u})$, and an optimal distribution can be derived for $q(\mathbf{u})$ (see Section 1.1.3.b for details on variational inference).

Although a large part of the early literature addressed the computational issue, several other problems and extensions that been studied in the GP context are worth a mention. First, we would often want to enable the assumption of non-Gaussian likelihoods, since it may sometimes be an unsuitable modelling of the phenomenons, for example in the presence of outliers or heavy tails. If we generally lose the explicit expressions by working out of the Gaussian framework, several inference approximations have been proposed (Neal, 1997) and implemented (Rasmussen and Nickisch, 2010; Vanhatalo et al., 2013) to extent GPs to a broad variety of likelihoods. In line with more elaborate likelihoods, we may also want to construct a model based on a mixture of GPs in an analogous way that Gaussian mixtures provide a prolific framework to handle diverse problems, such as unsupervised learning. Initial contributions (Tresp, 2001; Rasmussen and Ghahramani, 2002) proposed mixture of GP experts models to deal with local region of the input space. In the context of curve clustering, other approaches (Shi and Wang, 2008) lie on the concept of latent clusters and a mixture of GPs to retrieve the group structure in a dataset. Chapter 4 actually wraps this idea around the initial method presented in Chapter 3 to provide a cluster-specific extension. It might also appear too restrictive to manipulate a unique output variable in several applications, for instance in geostatistics that first focused on this problem. However, it remains possible to construct adequate covariance matrices to deal with this issue, and several models and approximations of multiple output GPs have been introduced (Boyle and Frean, 2005; Álvarez and Lawrence, 2011). A recent discussion on the subject can be found in Liu et al. (2018). With the boom of connected devices, the machine learning field has recently paid much attention to *online* applications and most of the classical algorithms have been extended to enable adding data and updating results on the flow. The GP framework does not make exception and several recent works (Bijl et al., 2015; Clingerman and Eaton, 2017; Moreno-Muñoz et al., 2019) proposed online extensions. Besides, the multi-task learning paradigm and its adaptation to GPs, which constitutes the main contribution of the present

thesis, shall be discussed more thoroughly in Section 1.1.4.b. Beforehand, although we gave an overview of the classical learning procedure in GPs, we did not introduce yet the practical methods that serve the inference purpose in our work. Moreover, we discussed several extensions proposed in the literature, and some of them lie on variational approximations, which has not been presented yet. To this end, we now take a quick detour through EM-like algorithms and variational inference before diving into deeper aspects of the multi-task GP framework.

1.1.3 Inference with Expectation-Maximisation procedures

1.1.3.a EM algorithm and Gaussian mixture models

The Expectation-Maximisation (EM) algorithm (Dempster et al., 1977; McLachlan and Krishnan, 2007) has initially been introduced to compute maximum likelihood estimators in missing data models. However, this approach covers today a broader range of problems where the equations cannot be solved directly, for example, when latent variables are involved in the model. A typical example of the use of EM occurs in mixture models where a latent variable is associated with each data point to define from which component this observation comes. Notice that the EM algorithm can also be used to find maximum a posteriori (MAP) estimates for the sought parameters. In this section, we first describe the EM algorithm in a general setting, before illustrating its practical application in the classical context of the Gaussian mixture models.

General EM

Let us define $\mathbf{X}$ a set of N observations generated from a given statistical model, and $\mathbf{Z}$ a set of latent (or missing) variables. Note that the present section is illustrated by using continuous latent variables, although discrete ones could be considered as well. We also note $\boldsymbol{\theta}$ the vector of parameters to be estimated and $L(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z}) = p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta})$ the complete-data likelihood function. In order to compute the maximum likelihood estimates (MLE) for $\boldsymbol{\theta}$, we would usually maximise the following marginal likelihood of the observed data:

$$L(\boldsymbol{\theta}; \mathbf{X}) = p(\mathbf{X} \mid \boldsymbol{\theta}) = \int p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta}) d\mathbf{Z}.$$

However, optimising over this quantity directly might be difficult or, as we shall see in Chapter 3, we may want to take advantage of the introduction of the latent variables Z. For any $\boldsymbol{\theta}^{old}$, let us write the following relation:

$$\begin{aligned}
\log p(\mathbf{X} \mid \boldsymbol{\theta}) &= \log \int p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta}) \, \mathrm{d}\mathbf{Z} && (1.8)\\
&= \log \int \frac{p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta})}{p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{old})} p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{old}) \, \mathrm{d}\mathbf{Z} && (1.9)\\
&= \log \mathbb{E}_{p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{old})} \left[\frac{p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta})}{p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{old})} \right] && (1.10)\\
&\geq \mathbb{E}_{p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{old})} \left[\log \frac{p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta})}{p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{old})} \right] && (1.11)\\
&= \mathbb{E}_{p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{old})} \left[\log p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta}) \right] - \mathbb{E}_{\log p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{old})} \left[p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{old}) \right] && (1.12)\\
&= \mathbb{E}_{p(\mathbf{Z}|\mathbf{X}, \boldsymbol{\theta}^{old})} \left[\log L(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z}) \right] + C. && (1.13)
\end{aligned}$$

The expression above comes from Jensen's inequality associated with the concavity of log function, and stands for any latent variables $\mathbf{Z}$ (for discrete distributions we may simply replace integrals by sums). Because the expectation would be taken over a constant in (1.11), the inequality turns into equality for $\theta^{old} = \theta$. From this result, we can deduce that if a simple formulation exists for $L(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z})$, it might be used for the optimisation of $\boldsymbol{\theta}$ instead of the marginal likelihood, and we shall see that the EM algorithm takes advantage of this property. In practice, by assuming that $\boldsymbol{\theta}^{old}$ is known, we can usually derive a closed-form expression for $\mathbb{E}_{p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{old})}[\log L(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z})]$. The idea behind the EM algorithm lies in the computing of this quantity alternatively with the optimisation of $\boldsymbol{\theta}$. More precisely, for an arbitrary threshold ϵ, we would repeat the two following steps until $\left|\theta^{new} - \theta^{old}\right| < \epsilon$:

Expectation step (E step): Compute $Q\left(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{old}\right)$ to be the expected value of the data-complete log-likelihood with respect to posterior distribution of $\mathbf{Z}$ with known $\boldsymbol{\theta}^{old}$:

$$Q\left(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{old}\right) = \mathbb{E}_{p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{old})}[\log L(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z})]. \tag{1.14}$$

Maximization step (M step): Find the parameters that maximise the previous quantity:

$$\boldsymbol{\theta}^{new} = \arg\max_{\boldsymbol{\theta}} Q\left(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{old}\right)$$

Interestingly, increasing $Q\left(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{old}\right)$ implicitly indicates that the marginal log-likelihood $\log p(\mathbf{X} \mid \boldsymbol{\theta})$ improves at least by the same quantity at each iteration (Little and Rubin, 2019). At the beginning of the procedure, $\boldsymbol{\theta}$ is set to a random value or initialised using smarter strategies according to the context. In general, the EM algorithm is only assured to converge to local maxima (Wu, 1983; Hathaway, 1986) of the likelihood function. However, many heuristics have been developed over the years to overcome this issue, such as a stochastic version (Celeux et al., 1992), simulated annealing (Ueda and Nakano, 1998) or repeated short runs (Biernacki et al., 2003).

Gaussian mixture models

A typical situation where the learning procedure is handled by an EM algorithm happens to be the inference in Gaussian mixture models. Often used in clustering problems, a Gaussian mixture assumes the presence of K different sets of mean vector and covariance matrix $\{\mu_k, \Sigma_k\}$ in the generative model of the data. Moreover, an associated vector $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)$, satisfying $\pi_k > 0, \forall k$, and $\sum_{k=1}^{K} \pi_k = 1$, defines the proportion of each component of the mixture. Formally, we say that an observation $\mathbf{x}_i$ comes from a Gaussian mixture distribution if:

$$p\left(\mathbf{x}_i \mid \boldsymbol{\pi}, \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1,\ldots,K}\right) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}\left(\mathbf{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right).$$

A way to come up with such a distribution lies on the definition and marginalisation over a latent variable $\mathbf{Z}_i = (Z_{i1}, \ldots, Z_{ik})$, where $Z_{ik} = 1$ if the i-th observations belongs to the cluster k, and 0 otherwise. We first assume that $\mathbf{Z}_i$ is sampled from a multinomial distribution:

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

Moreover, the i-th data point is sampled from a Gaussian distribution according to its class k:

$$p\left(\mathbf{x}_i \mid Z_{ik}=1, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)=\mathcal{N}\left(\mathbf{x}_i ; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right).$$

Hence, from a sample of independent observations $\mathbf{X}=\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ and by writing $\boldsymbol{\theta}=\{\boldsymbol{\pi}, \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1,\ldots,K}\}$, we have the following complete-data likelihood:

$$\begin{aligned}
L(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z}) &= p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta}) \\
&= \prod_{i=1}^{N} p(\mathbf{x}_i, \mathbf{Z}_i \mid \boldsymbol{\theta}) \\
&= \prod_{i=1}^{N} p(\mathbf{x}_i \mid \mathbf{Z}_i, \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1,\ldots,K}) p(\mathbf{Z}_i \mid \boldsymbol{\pi}) \\
&= \prod_{i=1}^{N} \prod_{k=1}^{K} p\left(\mathbf{x}_i \mid Z_{ik}=1, \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)^{Z_{ik}} \pi_k^{Z_{ik}} \\
&= \prod_{i=1}^{N} \prod_{k=1}^{K} \Big(\pi_k \, \mathcal{N}\left(\mathbf{x}_i ; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)\Big)^{Z_{ik}}.
\end{aligned}$$

We can then deduce the corresponding log-likelihood:

$$\log L(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z}) = \sum_{i=1}^{N} \sum_{k=1}^{K} Z_{ik} \Big(\log \pi_k + \log \mathcal{N}\left(\mathbf{x}_i ; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) \Big). \tag{1.15}$$

Recalling that (1.14) requires to take the expectation of the above expression with respect to the posterior distribution of $\mathbf{Z}$, we see that in this context, the *E step* consists in the computation of $\tau_{ik} := \mathbb{E}_{p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{old})}[\, Z_{ik} \,]$, $\forall i, \forall k$. From the Bayes rule, we can retrieve a factorized form for the posterior distribution of $\mathbf{Z}$:

$$\begin{aligned}
p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}) &= \frac{p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta})}{p(\mathbf{X} \mid \boldsymbol{\theta})} \\
&= \prod_{i=1}^{N} \frac{p\left(\mathbf{x}_i, \mathbf{Z}_i \mid \boldsymbol{\theta}\right)}{p\left(\mathbf{x}_i \mid \boldsymbol{\theta}\right)} \\
&= \prod_{i=1}^{N} \prod_{k=1}^{K} \left(\frac{\pi_k \mathcal{N}\left(\mathbf{x}_i ; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)}{\sum_{l=1}^{K} \pi_l \mathcal{N}\left(\mathbf{x}_i ; \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l\right)} \right)^{Z_{ik}} \\
&= \prod_{i=1}^{N} \mathcal{M}\left(\mathbf{Z}_i ; 1, \boldsymbol{\tau}_i = (\tau_{i1}, \ldots, \tau_{iK})^{\intercal}\right),
\end{aligned}$$

with:

$$\tau_{ik} = \frac{\pi_k \mathcal{N}\left(\mathbf{x}_i ; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right)}{\sum_{l=1}^{K} \pi_l \mathcal{N}\left(\mathbf{x}_i ; \boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l\right)}, \ \forall i, \forall k.$$

Thus, making use of the parameters θ^{old} coming from a former *M step* or the initialisation, we have a closed-form expression to update the value of τ_{ik}. This posterior probability for

the i-th observations to belong k-th class, sometimes called *responsibility*, shall now be plugged into (1.15) to define the optimisation function:

$$\begin{aligned} Q\left(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{old}\right) &= \mathbb{E}_{p(\mathbf{Z}|\mathbf{X},\boldsymbol{\theta}^{old})}[\log L(\boldsymbol{\theta}; \mathbf{X}, \mathbf{Z})] \\ &= \sum_{i=1}^{N}\sum_{k=1}^{K} \tau_{ik}\Big(\log \pi_k + \log \mathcal{N}\left(\mathbf{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) \Big). \end{aligned}$$

The M step requires to maximise the function $Q\left(\boldsymbol{\theta} \mid \boldsymbol{\theta}^{old}\right)$ with respect to the parameters $\boldsymbol{\theta} = \{\boldsymbol{\pi}, \{\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\}_{k=1,\dots,K}\}$, which are noticed to occur in independent terms of the overall sum. In this case, by setting the corresponding gradients to 0, we can first retrieve explicit weighted MLE for the mean and covariance of a Gaussian distribution:

$$\boldsymbol{\mu}_k^{new} = \frac{\sum_{i=i}^{N} \tau_{ik}\boldsymbol{x}_i}{\sum_{i=i}^{N} \tau_{ik}}, \qquad \boldsymbol{\Sigma}_k^{new} = \frac{\sum_{i=i}^{N} \tau_{ik}(\boldsymbol{x}_i - \boldsymbol{\mu}_k)(\boldsymbol{x}_i - \boldsymbol{\mu}_k)^\intercal}{\sum_{i=i}^{N} \tau_{ik}}, \ \forall k = 1, \dots, K.$$

Intuitively, we see that these expressions correspond to weighted averages using every data points. Since the weights $\{\tau_{ik}\}_{k=1,\dots,K}$ represent the posterior probabilities of lying in a cluster, an observation can be thought as contributing to each class proportionally to its belonging probability. To conclude, the optimisation over $\boldsymbol{\pi}$ requires some cautions regarding the constraint $\sum_{k=1}^{K} \pi_k = 1$. By using a Lagrange multiplier and setting the gradients to 0, it remains pretty straightforward to retrieve the MLE for the multinomial distribution:

$$\pi_k^{new} = \frac{1}{N}\sum_{i=i}^{N} \tau_{ik}, \ \forall k = 1, \dots, K.$$

A somehow standard procedure to initialise the classes relies on the use of results from a previous clustering algorithm such as k-means for example. We terminate the alternated updates when $\left|\boldsymbol{\theta}^{new} - \boldsymbol{\theta}^{old}\right| < \epsilon$, with $\epsilon > 0$ an arbitrary threshold, or when the likelihood has converged. The model selection problem of finding the right number of clusters K comes with extensive literature in itself. However, we may recall some classic criteria such as the *AIC* (Akaike, 1974), *BIC* (Schwarz, 1978), and *ICL* (Biernacki et al., 2000), or efficient heuristics like the *slope heuristic* (Birgé and Massart, 2006; Baudry et al., 2012) for instance. Comprehensive comparisons and discussions on the subject can also be found in Biernacki and Govaert (1999) or McLachlan and Peel (2004).

1.1.3.b Variational EM

We introduce in the following some tools from the *calculus of variations* (Bishop, 2006, Chapter 10) that lead to an extension of the EM algorithm called Variational EM (VEM). The inference procedure in Gaussian mixture models still serves as an illustrative example throughout the section. As we mentioned before, the *E step* of the algorithm necessitates the explicit computing of the posterior distribution of the latent variables thanks to the previously estimated parameters θ^{old}. However, let us recall that this distribution is defined

as:

$$p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta}^{old}) = \frac{p(\mathbf{X} \mid \mathbf{Z}, \boldsymbol{\theta}^{old}) p(\mathbf{Z} \mid \boldsymbol{\theta}^{old})}{p(\mathbf{X} \mid \boldsymbol{\theta}^{old})} = \frac{p(\mathbf{X} \mid \mathbf{Z}, \boldsymbol{\theta}^{old}) p(\mathbf{Z}, \boldsymbol{\theta}^{old})}{\int_{\mathbf{Z}} p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta}^{old}) \, \mathrm{d}\mathbf{Z}}.$$

In many cases, this posterior distribution is intractable whether because of the marginalisation at the denominator or, as we shall see in our model in Chapter 4, because some interdependent latent variables prevent from an exact formulation. In such a context, we may still want to derive an efficient algorithm by computing an approximation $q(\mathbf{Z}) \approx p(\mathbf{Z} \mid \mathbf{X})$ of the desired quantity. The idea behind this approximation would be to define an analytical expression, typically by restricting the choice to an usual family of distributions and picking the one that minimises the Kullback-Leibler (KL) divergence to the true posterior. In order to get the closest possible approximation, the appropriate family of distributions to consider depends on the model since we should take it as rich and flexible as possible once the minimal requirements for a tractable expression are fulfilled. Let us expand a new formulation for $\log p(\mathbf{X} \mid \theta)$, expressed from the KL divergence between $q(\mathbf{Z})$ and $p(\mathbf{Z} \mid \mathbf{X}, \theta)$:

$$\begin{aligned}
\mathrm{KL}(q \| p) &= \int q(\mathbf{Z}) \log \frac{q(\mathbf{Z})}{p(\mathbf{Z} \mid \mathbf{X}, \theta)} \, \mathrm{d}\mathbf{Z} \\
&= \int q(\mathbf{Z}) \left[\log \frac{q(\mathbf{Z})}{p(\mathbf{Z}, \mathbf{X} \mid \theta)} + \log p(\mathbf{X} \mid \theta) \right] \mathrm{d}\mathbf{Z} \\
&= \int q(\mathbf{Z}) [\log q(\mathbf{Z}) - \log p(\mathbf{Z}, \mathbf{X} \mid \theta)] \, \mathrm{d}\mathbf{Z} + \int q(\mathbf{Z}) [\log p(\mathbf{X} \mid \theta)] \, \mathrm{d}\mathbf{Z} \\
&= \int q(\mathbf{Z}) [\log q(\mathbf{Z}) - \log p(\mathbf{Z}, \mathbf{X} \mid \theta)] \, \mathrm{d}\mathbf{Z} + \log p(\mathbf{X} \mid \theta).
\end{aligned}$$

From the formulation above, we can rewrite the observed-data log-likelihood of the model as:

$$\begin{aligned}
\log p(\mathbf{X} \mid \boldsymbol{\theta}) &= \mathrm{KL}(q(\cdot) \| p(\cdot \mid \mathbf{X}, \boldsymbol{\theta})) - \mathbb{E}_{q(\mathbf{Z})}[\log q(\mathbf{Z}) - \log p(\mathbf{Z}, \mathbf{X} \mid \boldsymbol{\theta})] \\
&= \mathrm{KL}(q(\cdot) \| p(\cdot \mid \mathbf{X}, \boldsymbol{\theta})) + \mathcal{L}(q; \boldsymbol{\theta}),
\end{aligned}$$

where $\mathcal{L}(q; \boldsymbol{\theta})$ is a functional both of the distribution $q(\mathbf{Z})$ and the parameters $\boldsymbol{\theta}$. As we know that a KL divergence is always non-negative, $\mathcal{L}(q; \boldsymbol{\theta})$ provides a lower bound for $\log p(\mathbf{X} \mid \boldsymbol{\theta})$, with equality when $q(\mathbf{Z}) = p(\mathbf{Z} \mid \mathbf{X}, \boldsymbol{\theta})$ (i.e., $\mathrm{KL}(q(\cdot) \| p(\cdot \mid \mathbf{X}, \boldsymbol{\theta})) = 0$). If we assume $\boldsymbol{\theta}^{old}$ known, we may derive the *E step* of the VEM algorithm from the maximisation of the lower bound $\mathcal{L}(q; \boldsymbol{\theta})$ with respect to $q(\mathbf{Z})$, which is equivalent to minimising the KL divergence. A typical choice for the approximation $q(\mathbf{Z})$ is to consider a family of usual parametric distributions (e.g. Gaussian), and the problem of maximisation over a distribution reduces to the optimisation of the associated parameters (mean and covariance in the Gaussian case). In practice, we often do not need to state this explicitly, and we may simply induce an adequate family by assuming that the distribution over the latent variables factorizes over some convenient partition $\mathbf{Z}_1, \ldots, \mathbf{Z}_J$:

$$q(\mathbf{Z}) = \prod_{j=1}^{J} q_j(\mathbf{Z}_j).$$

For example, if we switch back to the Gaussian mixture problem where we add some additional latent variables $\mathbf{U}$ to the model, we now want to compute the posterior distribution of $\mathbf{Z}^+ = \{\mathbf{Z}_1, \dots, \mathbf{Z}_N, \mathbf{U}\}$. However, if the posteriors of $\mathbf{U}$ and $\mathbf{Z} = \{\mathbf{Z}_1, \dots, \mathbf{Z}_N\}$ depend upon each other (as in our model in Chapter 4), the explicit computation becomes intractable and we need to define an appropriate approximation. A clever approach in this context is to seek an analytical approximation that belongs to the family of distributions induced by the assumption of independence we lack: $q(\mathbf{Z}^+) = q_{\mathbf{U}}(\mathbf{U})q_{\mathbf{Z}}(\mathbf{Z})$. Such an assumption is the minimum we need to derive explicit formulations, and it naturally implies that the posterior approximations remain of the same form as the prior distributions whenever these prior distributions are members of the exponential family. In our example, we would even get a 'free' additional factorisation over the classes $q(\mathbf{Z}^+) = q_{\mathbf{U}}(\mathbf{U}) \prod_{i=1}^{N} q_i(\mathbf{Z}_i)$, and the approximate posterior over each $\mathbf{Z}_i$ would remain a multinomial distribution.

With some tools from the calculus of variations, it can be proved (Bishop, 2006, Chapter 10.1.1) that the optimal distributions $q_j^*(\mathbf{Z}_j)$ (minimising the KL divergence and maximising the lower bound) are given by:

$$q_j^*(\mathbf{Z}_j) = \mathbb{E}_{l \neq j} \left[\log p(\mathbf{X}, \mathbf{Z}) \right] + \text{ constant} , \tag{1.16}$$

where $\mathbb{E}_{l \neq j}$ indicates that the expectation is taken over all the variables of the partition $\mathbf{Z}_1, \dots, \mathbf{Z}_J$ except the j-th. The constant is an explicit term of normalisation that we can retrieve by inspection of the distribution in practice. In our example, it would mean that for any i, if we consider $\{q_1, \dots, q_{i-1}, q_{i+1}, \dots, q_N\}$ and $q_{\mathbf{U}}$ fixed and known, we can compute the optimal distribution $q_i^*(\mathbf{Z}_i)$ by taking the expectation of the complete-data log-likelihood with respect to them. Conversely, the solution for the additional latent variable $\mathbf{U}$ is expressed as:

$$q_{\mathbf{U}}^*(\mathbf{U}) = \mathbb{E}_{i=1,\dots,N} \left[\log p(\mathbf{X}, \mathbf{Z}) \right] + \text{ constant} , \tag{1.17}$$

where $\mathbb{E}_{i=1,\dots,N}$ is the expectation associated with the distributions $q_i(\mathbf{Z}_i)$, $\forall i = 1, \dots, N$.

As the expressions (1.16) and (1.17) provide solutions that depend upon each other, it suggests the derivation of a cyclic estimation procedure where we first initialise all the factors appropriately, and then iteratively compute each optimal distribution thanks to the others. For each $\{\mathbf{Z}_i\}_{i=1,\dots,N}$, we would identify the form of the multinomial distribution for $q_i^*(\mathbf{Z}_i)$ and deduce an explicit updating formula for its parameters τ_{ik}, as in the EM. Moreover, if the prior over $\mathbf{U}$ belongs to the exponential family, like the Gaussian distribution for example, we could also retrieve a Gaussian approximate posterior by inspection and derive the associated analytical expressions for the mean and covariance parameters. Such a procedure is assured to converge by convexity of the lower bound with respect to each factor (Boyd and Vandenberghe, 2004). Once $q^*(\mathbf{Z})$ is held fixed, the *M step* still requires to maximise the lower bound $\mathcal{L}(q; \boldsymbol{\theta})$ but this time with respect to $\boldsymbol{\theta}$, necessarily causing the observed-data log-likelihood $\log p(\mathbf{X} \mid \boldsymbol{\theta})$ to increases as well. The overall algorithm also converges to (local) maxima, and we provide the pseudo-code of a generic VEM algorithm in Algorithm 1 to clarify the different steps.

Let us stress that we only present here EM and VEM versions designed to find MLE of the parameters, as these methods are respectively used in Chapter 3 and Chapter 4 in the learning procedures of the algorithms we develop. However, the philosophy behind these approaches has a wider area of application and may be used to find MAP estimates or even handle inference in a fully Bayesian model. In order to treat the parameters in a Bayesian

Algorithm 1 The Variational EM algorithm

// INITIALISATION

Initialise the set of (hyper-)parameters $\boldsymbol{\theta}$.
Initialise the distributions $q_j(\mathbf{Z}_j),\ \forall j = 1, \dots, J$.

// OPTIMISATION

while did not converge **do**

E step: Maximise $\mathcal{L}(q; \boldsymbol{\theta}^{old})$ with respect to $q(\mathbf{Z})$:

$$q_j^*(\mathbf{Z}_j) = \mathbb{E}_{j \neq l}\left[\log p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta}^{old})\right] + \text{ constant }, \ \forall j = 1, \dots, J.$$

M step: Maximise $\mathcal{L}(q^*; \boldsymbol{\theta})$ with respect to $\boldsymbol{\theta}$:

$$\boldsymbol{\theta}^{new} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, \mathbb{E}_{q^*(\mathbf{Z})}\left[p(\mathbf{X}, \mathbf{Z} \mid \boldsymbol{\theta})\right].$$

end while

fashion, we would also need to define corresponding prior distributions and compute their posteriors for inference, as we did with latent variables. Such a model is somehow more general and powerful since it offers a full distribution, and thus uncertainty quantification, instead of point-wise estimates for the parameters. It also tackles the problem of the singularities that may arise in maximum likelihood estimation (Attias, 1999). However, this approach comes with an additional computational cost if an MCMC algorithm handles the inference, or with extra calculus and potentially complex derivations if variational inference is used instead. The variational Bayes EM (VBEM) algorithm offers a powerful way to provide analytical approximations of the posterior distributions in such models and has raised much interest in the past two decades (Beal and Ghahramani, 2003; Latouche et al., 2012).

1.1.4 Multi-task learning

1.1.4.a The multi-task paradigm

According to the comprehensive survey Zhang and Yang (2018), the multi-task learning (MTL) is "*a learning paradigm in machine learning that aims to leverage useful information contained in multiple related tasks to help improve the generalisation performance of all the tasks*". The term MTL has been introduced by Caruana (1997) with an illustration on shared hidden units in neural networks, and the initial ideas, as well as new ones, are since then adapted in many fields of machine learning. For a clear understanding, let us stress that we call *task* a process generating a batch of output data from the associated inputs. In this regard, different tasks may result in different outputs for the same set of input values, and we can see tasks as a higher level of hierarchy or structure in a dataset. Regardless of the specific models and algorithms that are used in practice, the MTL covers a wide range of designs supposed to be meaningful when it comes to sharing knowledge between multiple tasks. Among the important concepts, we can report the *feature learning approach* (Argyriou et al., 2007, 2008), where we assume that all tasks share the same feature representation. The main work then involves constructing this relevant new expression from the original data. For example, Maurer et al. (2013) proposes to derive a sparse representation using a linear transformation of the initial features. On the other hand, the well-named *task clustering*

approach (Thrun and O'Sullivan, 1996) aims at defining groups of similar tasks in the same fashion that classical clustering algorithms define clusters of resembling data points. In this framework, the answered question is more about which tasks should share information than in the way it is actually shared (Kang et al., 2011), and task clustering might be thought in combination with other approaches. By loosening the strict belonging to a cluster, we would fall into the spectrum of *task relation learning* (Evgeniou et al., 2005) where the relation between tasks is rather quantified through a similarity or a covariance measure. In this sense, most of the multi-task models expressed in terms of Gaussian processes, which we discuss further on, live in this category. Last but not least, the *decomposition approach* is somehow a meta-view on the multi-task paradigm that associates different regularisation strategies on the model parameters. Jalali et al. (2010), for instance, make use of both ℓ_1 and ℓ_∞ regularisation to offer a sparse representation in two different aspects, but this idea may be enhanced, even to tree-structured levels of regularisation (Jawanpuria and Nath, 2012).

1.1.4.b Multi-task Gaussian processes models

There have been several different approaches in the literature claiming for the title of *multi-task Gaussian process* models. We have seen in the previous section that this paradigm can be developed from many perspectives and offers more of a general philosophy on the learning process than a well-defined model or heuristic. When it comes to GP, we can still highlight the work developed in Bonilla et al. (2008) that has had much influence over the years, leading to further developments (Zhang and Yeung, 2012; Chen et al., 2018) and numerous applications (Williams et al., 2009; Ghassemi et al., 2015). To introduce some aspects of this model, we need to adjust a bit the notation introduced in Section 1.1.2.d to the multi-task framework. As the data are supposed to come from different sources, we introduce the index $i = 1, \dots, M$ to differentiate the *individuals* (we use this word instead of *tasks*, *batches*, or others for consistency when it comes to our main illustrative example). As before however, we observe the outputs $\mathbf{y}_i = \{y_i(t_1), \dots, y_i(t_{N_i})\}$ for the i-th individual at timestamps $\mathbf{t}_i = \{t_1, \dots, t_{N_i}\}$. Let us stress that the number of timestamps N_i as well as their location might be different from one individual to another in general. We also note $\mathbf{t} = \bigcup_{i=1}^{M} \mathbf{t}_i$, the N-dimensional set of pooled timestamps. With our notation, the case where all individuals are observed on the same grid exactly is equivalent to assume $N = N_i, \ \forall i = 1, \dots, M$, and we suppose it to be the case in the following for simplicity. The regression model in this context becomes:

$$y_i(t) = f_i(t) + \epsilon_i, \ \forall i = 1, \dots, M, \forall t \in \mathcal{T}.$$

The idea behind the so-called *multi-task Gaussian process* (Bonilla et al., 2008) model is based on the introduction of a specific kernel form with two factors. As we want to introduce some task-related influence between the individuals, the function f is supposed to be a GP (with zero mean for simplicity) with an evaluation of its covariance function on observed inputs defined such as:

$$cov\left(f_i(t_k), f_j(t_l)\right) = \left[\mathbf{K}^f\right]_{ij} k^t(t_k, t_l), \ \forall i, j = 1, \dots, M, \forall t_k, t_l \in \mathbf{t},$$

where $\mathbf{K}^f$ is a $M \times M$ positive semi-definite matrix specifying the inter-task similarities, and the notation $\left[\cdot\right]_{ij}$ stands for the extraction of the i-th line and j-th column of a matrix. The kernel $k^t(\cdot, \cdot)$ defines, as for usual covariance structures in GPs, the relationships between the

inputs. Assuming a noise variance σ_i^2 for the i-th individual, the conditional distributions remain:

$$p(\mathbf{y}_i \mid \mathbf{f}_i, \sigma_i^2) = \mathcal{N}\left(\mathbf{y}_i; \mathbf{f}_i, \sigma_i^2 I_N\right), \ \forall i = 1, \dots M.$$

Then, we way still derive the marginal joint distribution over $\{\mathbf{y}_i\}_i = \{\mathbf{y}_1, \dots, \mathbf{y}_M\}$, by integrating out the $\{\mathbf{f}_i\}_i$:

$$p(\{\mathbf{y}_i\}_i \mid \Theta, \mathbf{t}) = \mathcal{N}\left(\{\mathbf{y}_i\}_i; 0, \mathbf{\Sigma}\right),$$

where Θ is the set of all hyper-parameters and $\mathbf{\Sigma} = \mathbf{K}^f \otimes \mathbf{K}^t + \mathbf{D} \otimes I_N$ is an $MN \times MN$ matrix. Moreover, $\mathbf{K}^t$ is the $N \times N$ Gram matrix associated with $k^t(\cdot, \cdot)$, $\mathbf{D}$ is the $M \times M$ diagonal matrix of task noises such as $[\mathbf{D}]_{ij} = \delta_{ij}\sigma_i^2$, and $\otimes$ stands for the Kronecker product. Considering this joint distribution and what we said about GP's prediction formula in Section 1.1.2.c, it appears obvious that we can derive the posterior distribution as usual. This prediction would take into account the observations of every individual, and weight them both in regards to the covariance between the timestamps, as usual, but also to the covariance between tasks. This way, we have a contribution of all individuals and a way of sharing information across them that we hope to enhance the results. This model provides an elegant framework to integrate the philosophy of multi-task learning into the GP regression while keeping closed-form expression throughout.

However, we can notice a major drawback, especially in the light of our previous remarks on the scalability of GPs. The covariance matrix thereby defined $\mathbf{\Sigma}$ has a dimension $MN \times MN$, and then a computational cost in $\mathcal{O}(M^3N^3)$ for inversion, which may discard its applicability in most cases. Moreover, the inference procedure necessitates estimating the hyper-parameters $\Theta = \{\theta^t, \mathbf{D}, \mathbf{K}^f\}$, where the $M \times M$ inter-task covariance matrix may become unrealistic to learn when the number of individual increases. It remains possible, however, to maximise the marginal likelihood directly through gradient-based methods and using Cholesky decomposition for $\mathbf{K}^f$ to remain positive semi-definite, but authors in Bonilla et al. (2008) proposed another solution. Indeed, by exploiting the Kronecker product structure of $\mathbf{\Sigma}$, it is possible to derive an EM algorithm with closed-form updates for $\mathbf{K}^f$ (which will remain positive semi-definite as well) and $\mathbf{D}$, to decouple their learning to θ_t's. Hence, by computing a new likelihood formula with current values $\hat{\Theta}$ at the E-step, and updating matrices after θ_t's optimisation at the M-step, we can iterate until convergence to complete the inference. This approach tackles the issue of optimising too many parameters while leaving unchanged the computational cost. As we have previously seen, there yet exist many approaches to provide sparse GP approximations (see Section 1.1.2.d) and authors propose an adaptation of these methods to reduce the computational burden to $\mathcal{O}(MNP^2Q^2)$, where $P < M$ and $Q < N$. Once more, let us advise to keep these values in mind for comparison purpose when we discuss the computational cost of our multi-task GP proposal in Chapter 2 (which remains in $\mathcal{O}(MN_i^3 + N^3)$ before any sparse approximation). As an additional remark, if we consider the noise free case, i.e. $p(\{\mathbf{y}_i\}_i \mid \Theta, \mathbf{t}) = \mathcal{N}\left(\{\mathbf{y}_i\}_i; 0, \mathbf{K}^f \otimes \mathbf{K}^t\right)$ and a block-design of the covariance matrix, a decorrelation phenomenon appears and $\mathbf{K}^f$ does not influence the prediction any more. Hence, although this property may seem appealing regarding the computational cost reduction, it remains pointless in practice since this would cancel all inter-task transfer, which is the whole purpose of the model.

Many direct extensions have been proposed (Hayashi et al., 2012; Rakitsch et al., 2013) in the literature to provide additional features to this approach. There also exist alternative models (Teh et al., 2005) that are worth a mention, such as models with conditionally independent $\{\mathbf{y}_i\}_i$, sharing the same covariance function across individuals (Schwaighofer et al.,

2005; Yu et al., 2006). In particular, the geostatistics field have developed (Zhang, 2007; Genton and Kleiber, 2015) many aspects on related problems although using a different vocabulary, namely the *intrinsic model of coregionalization*. An interesting application of the multi-task Gaussian process models occurs in the framework of Bayesian optimisation (Swersky et al., 2013). The problem of finding an appropriate setting for hyper-parameters being unfortunately often neglected, this work proposes to leverage previously trained models in order to quickly tune new ones, resulting in valuable enhancements of the optimisation process. Another notable application can be reported with the use of multi-task GP for causal inference (Alaa and van der Schaar, 2017), which aims at helping to define individualised treatments for medical purpose. Besides, as the issue of computing cost always remains an important topic, Zhu and Sun (2014) introduce a multi-task sparsity regulariser for the subset selection of multiple Gaussian processes. Finally, the recent work of Clingerman and Eaton (2017) develops a *lifelong learning* approach that enables online transfer between GP models in order to learn multiple tasks consecutively.

On the other hand, several other methods have been named using the term *multi-task GP* over time while referring to different strategies. In particular, some models have been developed (Yu et al., 2005; Shi et al., 2007; Yang et al., 2016, 2017) not only focusing on the covariance structure, but also considering the mean function in the multi-task strategy. Since our forthcoming developments have been inspired by some of these ideas, let us provide insights on their work and thus motivate the interest of sharing information through the mean function in GP models for multi-task learning purpose. Originally, Yu et al. (2005) offered an extensive study of the relationships between the Bayesian hierarchical linear models and GPs to develop a corresponding multi-task GP formulation. Hierarchical Bayesian modelling provides a natural way of specifying a relation between tasks by assuming that model parameters are drawn from a common hyper-prior distribution. Let us introduce their ideas by recalling the linear model with $\mathbf{y} \in \mathbb{R}^N$ and $\mathbf{X} \in \mathbb{R}^{N \times d}$, in a Bayesian point-of-view:

$$\mathbf{y} = \mathbf{X}\beta + \epsilon,$$

where β is a d-dimensional random vector such as $\beta \sim \mathcal{N}\left(\beta; \boldsymbol{\mu}_\beta, \mathbf{C}_\beta\right)$, and ϵ a Gaussian white noise N-dimensional vector. In this hierarchical approach, the aim is to obtain *type II* maximum likelihood estimates for the parameters $\{\boldsymbol{\mu}_\beta, \mathbf{C}_\beta\}$ by assuming the following hyper-prior distribution, which is conjugated with the multivariate Gaussian likelihood of the model:

$$\begin{aligned} p\left(\boldsymbol{\mu}_\beta, \mathbf{C}_\beta\right) &= p\left(\boldsymbol{\mu}_\beta \mid \mathbf{C}_\beta\right) p\left(\mathbf{C}_\beta\right) \\ &= \mathcal{N}\left(\boldsymbol{\mu}_\beta; \boldsymbol{\mu}_{\beta_0}, \frac{1}{\pi}\mathbf{C}_\beta\right) \mathcal{IW}\left(\mathbf{C}_\beta; \tau, \mathbf{C}_{\beta_0}\right), \end{aligned}$$

where π, τ, $\boldsymbol{\mu}_{\beta_0}$ and $\mathbf{C}_{\beta_0}$ are fixed scalars, vector and matrix to set. By this mean, an EM algorithm can be derived with closed-form updates for $\boldsymbol{\mu}_\beta$, $\mathbf{C}_\beta$ and the noise variance. In the case where we only focus on the function values and the covariance matrix on a finite set of data, and we explicitly know coordinates of the inputs in the feature space (quite unusual in the GP context), an adaptation to the GP framework is proposed with:

$$p\left(\boldsymbol{\mu}_\mathbf{f}, \mathbf{K}\right) = \mathcal{N}\left(\boldsymbol{\mu}_\mathbf{f} \mid 0, \frac{1}{\pi}\mathbf{K}\right) \mathcal{IW}(\mathbf{K} \mid \tau, \boldsymbol{\kappa}), \tag{1.18}$$

where we consider $\mathbf{f} \sim \mathcal{N}(\boldsymbol{\mu}_\mathbf{f}, \mathbf{K})$ a GP evaluated on the set of inputs, π still is a fixed scalar, and $\boldsymbol{\kappa}$ is the kernel defined by the inner product between inputs. The Normal-inverse-Wishart hyper-prior distribution above still allows for the derivation of an EM algorithm for

inference as before. This so-called *transductive multi-task GPs* approach is, in general, not convenient in practice because each observation of new test data forces to re-run the EM algorithm. An *inductive multi-task GP* model is finally proposed, thanks to the definition of parametric representations of the target mean $\boldsymbol{\mu}_{\mathbf{f}}$ and covariance $\mathbf{K}$. An EM algorithm and explicit updates formulas are again provided for both those quantities and the newly introduced parameters. As this approach remains a bit unclear on the way the parametric approximation is derived though, we prefer to remain careful on the interpretations of this model. However, this approach inspired the further work of Yang et al. (2016), which gives a generalisation of this model for tackling the FDA problem of simultaneously smooth multiple curves. By assuming general covariances structures and a fully Bayesian hierarchical model, an MCMC algorithm has to be derived for inference. As a consequence, this very flexible model may suffer from a computational burden that motivated the introduction (Yang et al., 2017) of a more efficient approximation using a basis functions representation.

Finally, let us conclude this state-of-the-art chapter by reporting the series of articles (Shi et al., 2005, 2007; Shi and Wang, 2008; Wang and Shi, 2014) and the gathering book (Shi and Choi, 2011) presenting the so-called *Gaussian process functional regression* framework. The kind of tackled problems and the point-of-view of this approach somehow resembles some aspects of the present thesis, both in sharing information through the mean function in GP regression (Shi et al., 2007) (Chapter 3) and in the further extension to a GPs mixture model for cluster-specific predictions (Shi and Wang, 2008) (Chapter 4). Although these models are not introduced as multi-task learning methods, we can retrieve a common philosophy in the will of handling simultaneously multiple curves to improve each individual modelling. This work particularly focuses on data that are considered as functional and thus makes use of several models we previously introduced, such as basis functions or GP regression. Let us highlight only a few aspects of these methods as a transition to the following chapters of the present manuscript while avoiding too many overlaps. The considered regression model is somehow more general than the previous ones since we still use $t \in \mathcal{T}$ as an input, but also introduce a set of additional functional covariates $\mathbf{x}_i(t) = \{x_i^1(t), \ldots, x_i^r(t)\}$ along with scalar ones $\mathbf{u}_i = \{u_i^1, \ldots, u_i^h\}$, such as:

$$y_i(t) = f_i\left(t, \mathbf{x}_i(t), \mathbf{u}_i\right) + \epsilon, \ \forall i = 1, \ldots, M, \forall t \in \mathcal{T}.$$

The form of the function f_i is chosen to be, for all individuals, the sum of a mean deterministic function μ_i and a centred Gaussian process $\tau_i(\cdot)$, such as:

$$f_i(t) = \mu_i(\mathbf{u}_i, t) + \tau_i(\boldsymbol{x}_i(t)), \ \forall i = 1, \ldots, M, \forall t \in \mathcal{T},$$

with $\tau_i(\cdot) \sim \mathcal{GP}\left(0, k(\cdot, \cdot)\right)$ and $\mu_i(\mathbf{u}_i, t) = \mathbf{u}_i^\intercal \beta(t)$ being a linear combination of the scalar inputs with a functional term to estimate (a modelling choice proposed in Ramsay and Silverman, 2005). This functional term is common to all individuals and estimated thanks to basis function expansion (see Section 1.1.1.c). We can thus deduce an estimation $\hat{\mu}_i$ for the mean function:

$$\hat{\mu}_i(t) = \mathbf{u}_i^\intercal \hat{\mathbf{B}} \boldsymbol{\phi}(t), \ \forall i = 1, \ldots, M, \forall t \in \mathcal{T},$$

where $\boldsymbol{\phi}(t) = \{\phi_1, \ldots, \phi_B(t)\}$ is a vector of fixed basis functions and $\hat{\mathbf{B}}$ the corresponding $h \times B$ matrix of coefficients estimated from the dataset. Plugging this estimate into the model evaluated on the input points, the authors define the estimated evaluation $\hat{\boldsymbol{\tau}}_i$ of the centred GP:

$$\hat{\boldsymbol{\tau}}_i = \mathbf{y}_i - \hat{\mu}_i(\mathbf{t}), \ \forall i = 1, \ldots, M.$$

These values are then used in the likelihood of the GP model to estimate the hyper-parameters of the kernel $k(\cdot, \cdot)$ as usual. Finally, the classical prediction formulas for GP can be derived for prediction purpose. As mentioned initially, this model raised the idea of a shared mean function estimated with data from all individual as a way of borrowing information across individuals. However, some modelling choices remain rather arbitrary, and the use of a parametric basis function expansion for the mean function may seem odd in a GP framework, which we have presented here as a convenient extension of this approach. Moreover, the basis function modelling drags in the model its usual practical inconveniences such as smoothing parameter selection, explicit choice of the basis, difficulties in handling uncommon grids of inputs. Although we became aware of this work after the beginning of our own developments, it remains accurate to consider the models presented in Chapter 3 and Chapter 4 as an extension to a fully non-parametric GP framework. In this consideration, our *multi-task GPs with common mean* approach offers both modelling and practical improvements, as detailed in the subsequent section.

1.2 Contributions

1.2.1 Context

Functional data analysis (FDA) has been an active area of research in statistics for the past two decades. Many real-world data that are currently collected can be considered as intrinsically functional as they come from time-dependent phenomenons or are observed over a continuum. Among the diversity of applicative fields, sports science still appear as mainly unexplored, although promising in regards to the diversity of available datasets and interesting related questions. This thesis initially took its roots from the collaboration with sports federations around the problem of talent detection among young athletes. Recent studies (Boccia et al., 2017; Kearney and Hayes, 2018) reporting careers of thousands of athletes exhibit only a weak relationship between the performances at young and adult age. In order to enlighten some aspects of the talent identification decision-making process, our work focuses on two main questions: Are there such things as typical patterns of progression among athletes? Can we imagine using similar features between individuals to improve the forecast of future performances? The French Swimming Federation (FFN) provided several datasets that serve as a leading thread application to illustrate the methodologies developed throughout this manuscript. In the sequel, we mainly focus on two datasets (one for women, one for men) that gather the performances of the FFN members between 2000 and 2016 in competitions of 100m freestyle (50m pool). During its career, we assume that the athlete's performance level over time can be represented by a function called *progression curve*, which constitutes our main object of interest. The point-wise observations of these curves being provided by the competition results, our data consist of a set of time series, observed irregularly among individuals. As illustrated on Figure 1.3, the sparsity and irregularity in the observations offer a significant challenge when it comes to comparing individual from one another and proceed to thorough analyses.

The three main chapters of this thesis propose methodological developments and solutions of increasing efficiency to our two main questions. Chapter 2 offers an initial exploration of the datasets and highlights the existence of different patterns of progression thanks to the application of curve clustering methods. Then, we introduce in Chapter 3 a novel multi-task Gaussian processes model along with the associated learning algorithm and prediction formulas, providing a well suited probabilistic modelling. Finally, reusing the idea of group

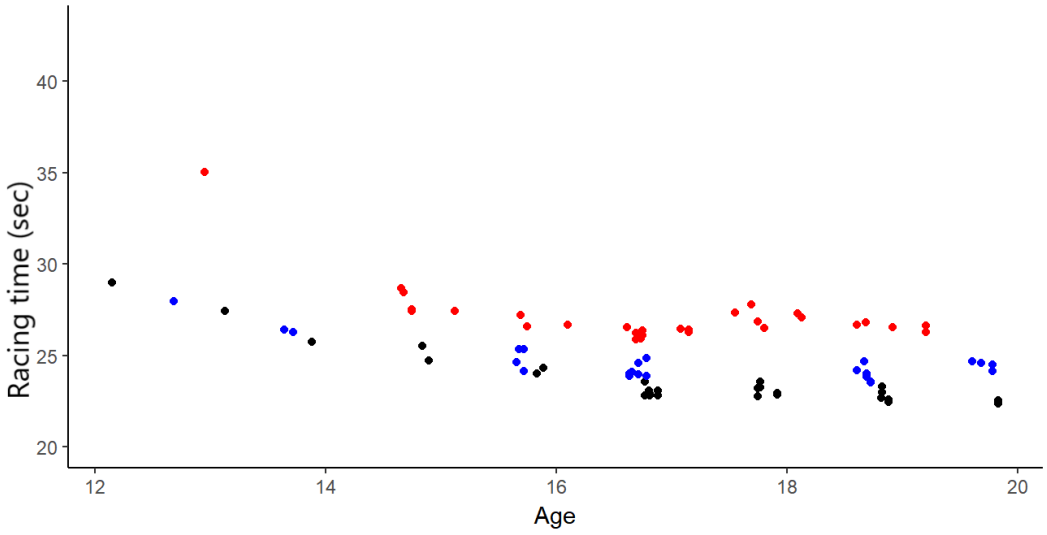

Figure 1.3 – Example of data points associated with 3 different swimmers (respectively in blue, red and black). Inputs are the age at the competition's day. Outputs are the racing time in competition (here for the male 50m freestyle).

structures in the data, Chapter 4 proposes a generalisation of the previous model by developing a multi-task GPs mixture model, allowing for weighted cluster-specific predictions.

1.2.2 Exploration and clustering of the swimmers' progression curves

Following the path of the quick review proposed in Section 1.1.1.d about curves clustering methods, we start this chapter with a comparison on synthetic data between several state-of-the-art algorithms, gathered within the R package *funcy*. Three different situations of increasing difficulty are proposed, and the algorithms are compared (Table 2.2) both on their capacity to retrieve the correct cluster for each curve and on their running time. The performances of each method are analysed in regards to their ability to deal with the different problems, and the advantages and drawbacks of the various approaches are discussed.

Subsequently, we propose a first attempt to model and analyse the swimmers' progression curves as functional data. As we illustrate our methods on time series, we generally use the corresponding vocabulary and refers to the input values as *timestamps*. Moreover, we also call *individual* each entity possessing its own batch or set of data, to remain consistent with the swimmers' application. A dataset is composed of M individuals, each of them observed on N_i (potentially different from one individual to another) data points. Thus, for all $i = 1, \dots, M$, we are provided with a set $\{(t_i^1, y_i(t_i^1)), \dots, (t_i^{N_i}, y_i(t_i^{N_i}))\}$ of inputs and associated outputs. Since many objects are defined for all individuals in the sequel, we shorten our notation as such: for any object x existing for all i, we denote $\{x_i\}_i = \{x_1, \dots, x_M\}$. Convenient notation follows:

- $\mathbf{t}_i = \{t_i^1, \dots, t_i^{N_i}\}$, the set of timestamps for the i-th individual,
- $\mathbf{y}_i = y_i(\mathbf{t}_i)$, the vector of outputs for the i-th individual,

- $\mathbf{t} = \bigcup_{i=1}^{M} \mathbf{t}_i$, the pooled set of timestamps among individuals,
- $N = \#\{\mathbf{t}\}$, the total number of observed timestamps.

For each individual, a common B-spline basis $\left\{\mathcal{B}^1, \ldots, \mathcal{B}^B\right\}$ is defined for a decomposition such as:

$$y_i(t) = \sum_{b=1}^{B} \alpha_i^b \mathcal{B}^b(t), \ \forall i, \forall t.$$

This way, the individual-specific information of each functional data is represented by the set $\left\{\alpha_i^1, \ldots, \alpha_i^B\right\} \in \mathbb{R}^B$ of the B-spline coefficients. This set of coefficients being of equal dimension for each curve, any classical curve clustering algorithm can now be applied. We choose to use the *funHDDC* algorithm proposed in (Bouveyron and Jacques, 2011; Schmutz et al., 2018) both for its performance and the ability to deal with multidimensional functions. As we exhibit it in a prior exploration of the dataset through an FPCA, the different modes of variation among curves and their derivatives suggest that they bring essential information altogether. By performing a curve clustering solely using the splines coefficients, the resulting groups appear as uninformative, mainly using the relative position of curves from one another on the y-axis and ignoring more subtle modes of variation. Therefore, we include the coefficients of the curves' derivative as an additional functional variable, in order to bring complementary features about the progression dynamics into the clustering. Such an approach gives satisfactory results since the compromise between the level of performance and progression patterns offers a broader view on the way swimmers improve over time. The resulting groups proved to be coherent with the knowledge of experts from the swimming federation. In particular, we identify specific patterns of early or late progression, highlighting that many swimmers can fill a gap at older ages with a higher rate of improvement (see Figure 1.4). Although this approach allows us to enlighten some important features of the dataset and group structures, it also suffers from several modelling issues. On the one hand, the lack of available data points for some individuals makes the global parametric B-splines decomposition difficult, sometimes leading to unsatisfactory individual modelling. On the other hand, this frequentist approach does not offer uncertainty quantification either for modelling or prediction purpose, which would yet be valuable in such a decision-making problem. These many obstacles lead us to the methodological developments at the heart of this thesis, taking place in the non-parametric and probabilistic framework of Gaussian processes.

1.2.3 Multi-task Gaussian processes with common mean process

The Gaussian process framework offers an elegant way to model the underlying function, mapping an input variable onto the output in a supervised context. Despite many nice properties, GPs suffer from their computational cost in $\mathcal{O}(N^3)$, which we will not focus on here, and from generalisation issues when data points are whether sparsely or poorly distributed over the input domain. As our applicative datasets contain many individuals ($\simeq 10^4$) each one observed on only a few locations ($\simeq 10^1$), the definition of a multi-task model would offer the opportunity of sharing information across the individuals to tackle this issue. The novelty of this approach lies in the introduction of a mean process, common to all individuals. This mean function is defined as a Gaussian process, for which the hyper-posterior distribution is tractable, and provides to each individual a prior mean value that contains information over the whole domain and thus improves the predictive capacities.

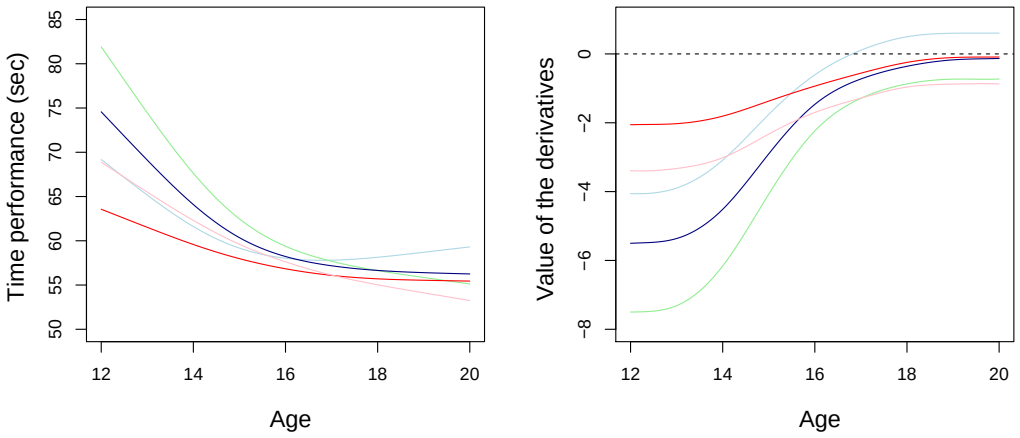

Figure 1.4 – Mean curves (left) and mean derivatives (right) resulting from the clustering of the swimmers' progression curves into 5 groups.

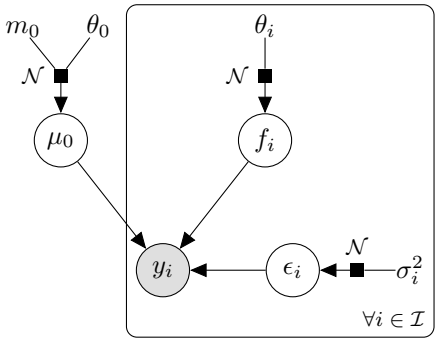

Figure 1.5 – Graphical model of dependencies between variables in our Multi-task GPs model.

The functional data $y_i(t)$ associated with the i-th individual is assumed to be generated from the following model:

$$y_i(t) = \mu_0(t) + f_i(t) + \epsilon_i(t), \ \forall i, \forall t,$$

where μ_0 is the common mean GP, f_i is a centred individual-specific GP, and ϵ_i a noise term. The assumptions on the model are summarized in the graphical model displayed on Figure 1.5. Thanks to a sample of data $\{\mathbf{t}_i, \mathbf{y}_i\}_i$, the inference of such model requires the estimation of the hyper-parameters associated with the GPs' covariance kernel and the computing of μ_0's hyper-posterior distribution.

Those quantities being interdependent, an Expectation-Maximisation (EM) algorithm is derived for this purpose. In the *E-step*, the hyper-posterior distribution of μ_0 can be computed explicitly using the current values of the hyper-parameters, as detailed in Propo-

	Prediction		Estimation μ_0	
	MSE	CI_{95}	MSE	CI_{95}
MAGMA	**18.7 (31.4)**	**93.8 (13.5)**	**1.3 (2)**	**94.3 (11.3)**
GPFDA	31.8 (49.4)	90.4 (18.1)	2.4 (3.6)	$\star$
GP	87.5 (151.9)	74.0 (32.7)		

Table 1.1 – Average MSE (sd) and average CI_{95} coverage (sd) on 100 runs for GP, GPFDA and MAGMA. ($\star$: 99.6 (2.8), the measure of incertitude from the GPFDA package is not a genuine credible interval)

sition 3.1. Conversely, Proposition 3.2 and Proposition 3.3 detail the hyper-parameters optimisation formulas that operate in the *M-step* of the algorithm. Explicit gradients associated with the functions to maximise can be derived for facilitating the optimisation procedure in practice. By alternatively repeating those two steps, we converge to local optima of the likelihood and reach appropriate estimates for the desired quantities.

The subsequent prediction procedure is operated in a few different steps. If we observe a new individual, called $*$, at timestamps $\mathbf{t}_*$ for whom we want to predict its output values at timestamps $\mathbf{t}^p$, we shall define the corresponding pooled grid $\mathbf{t}_*^p$. Depending on the assumption on the model, it might also be necessary to compute the hyper-parameters associated with the covariance kernel of the new individual. Then, the μ_0's hyper-posterior is computed on the grid of timestamps $\mathbf{t}_*^p$ (Proposition 3.4) and integrated out in the proposition below, expressing the prior multi-task distribution for the new individual:

Proposition 1.1. *For a set of timestamps $\mathbf{t}_*^p$, the multi-task prior distribution of y_* is given by:*

$$p(y_*(\mathbf{t}_*^p) \mid \{\mathbf{y}_i\}_i) = \mathcal{N}\Big(y_*(\mathbf{t}_*^p); \hat{m}_0(\mathbf{t}_*^p), \Gamma_*^p\Big).$$

The quantities $\hat{m}_0(\mathbf{t}_*^p)$ and Γ_*^p involved in this expression are all known from the learning or the previous steps. Finally, we establish the posterior distribution as usual in GP regression:

$$p(y_*(\mathbf{t}^p) \mid y_*(\mathbf{t}_*), \{\mathbf{y}_i\}_i) = \mathcal{N}\left(y_*(\mathbf{t}^p); \hat{\mu}_0^p, \hat{\Gamma}^p\right),$$

where:

- $\hat{\mu}_0^p = \hat{m}_0(\mathbf{t}^p) + \Gamma_{p*}\Gamma_{**}^{-1}\left(y_*(\mathbf{t}_*) - \hat{m}_0(\mathbf{t}_*)\right),$
- $\hat{\Gamma}^p = \Gamma_{pp} - \Gamma_{p*}\Gamma_{**}^{-1}\Gamma_{*p}.$

The final predictive formula above integrates both the information and the uncertainty over the mean process. This posterior multi-task distribution provides a significant improvement in the predictive performances over a wide domain of timestamps, even in the absence of individual-specific observations.

The algorithmic complexity of the whole method is discussed as we need to pay the price $\mathcal{O}(M \times N_i^3 + N^3)$ because of the multi-task framework, although once the training step is performed in advance, the on-the-fly prediction remains equivalent to a single task GP. The overall algorithm implementing this method is called MAGMA (standing for Multi-tAsk Gaussian processes with common MeAn). We propose an extensive simulation study to illustrate several properties of our approach as well as to compare its performance to competing state-of-the-art methods. Contrarily to the alternative algorithm GPFDA described

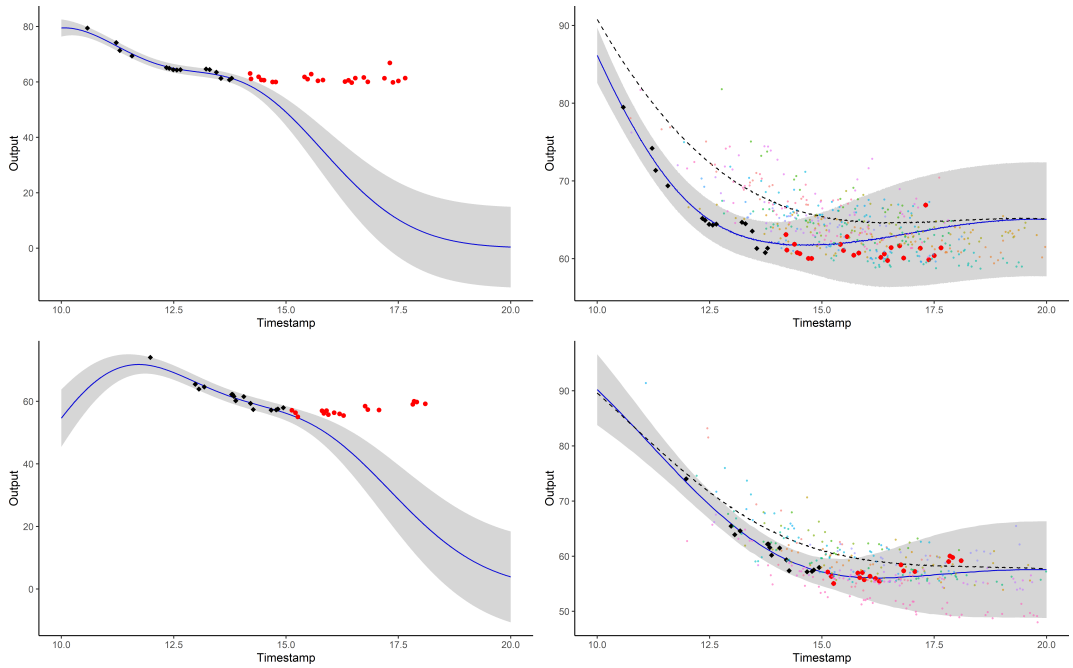

Figure 1.6 – Prediction curves (blue) for a testing individual with associated 95% credible intervals (grey) for GP regression (left) and MAGMA (right), for both women (top) and men (bottom). The dashed lines represent the mean functions of the hyper-posterior mean process. Observed data points are in black, testing data points are in red. The colourful backward points are observations from the training dataset, each colour corresponding to a different individual.

in Shi and Choi (2011), MAGMA accounts for μ_0's uncertainty and handle uncommon grids of observations while maintaining explicit formulations. The comparisons both in μ_0's estimation and predictive performances between GPFDA, usual GP regression, and MAGMA are summarised in Table 1.1. We observe that our method outperforms the alternatives both in mean squared error (MSE) and in the ratio of 95% credible interval (CI_{95}) coverage, a quantity measuring the adequacy of the uncertainty quantification. Sharing information across individuals through the process μ_0 also proves to be efficient in the context of swimmers' progression curves, on which the predictive results remain highly satisfactory. An illustration of the predictive advantage of our multi-task approach in this context is displayed on Figure 1.6 where we compare GP regression to MAGMA on data coming from a random individual for both men and women.

1.2.4 Multi-task Gaussian processes mixture and curve clustering

The idea behind this chapter comes from our leading thread example, as we recall that some datasets present group structures and we may take advantage of such a feature. Therefore, we propose an extension of the previous model by defining a multi-task mixture of GPs model. To tackle the problem of a unique underlying mean process hypothesis that might appear as too restrictive, we introduce a set of K mean processes, each one being associated with a specific cluster. Assuming that the i-th individual belongs to the k-th cluster, we

define the generative model for $y_i(t)$ as:

$$y_i(t) = \mu_k(t) + f_i(t) + \epsilon_i(t), \forall t,$$

where μ_k is the k-th cluster-specific mean GP while f_i and ϵ_i remain respectively the i-th individual-specific GP and the noise term. This new model also depends on latent multinomial variables $\{Z_i\}_i$ controlling the memberships of the mixture. The overall interactions between those quantities are summarised in the graphical model displayed on Figure 1.7.

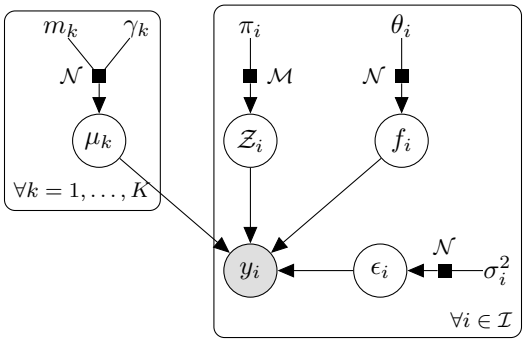

Figure 1.7 – Graphical model of dependencies between variables in our multi-task GPs mixture model.

This novel approach requires to estimate the hyper-parameters of each GP's covariance kernel jointly with the hyper-posterior distributions over the $\{\mu_k(\cdot)\}_k$ processes and $\{\mathbf{Z}_i\}_i$ variables. The posterior dependencies between those last quantities now force us to develop a variational EM (VEM) algorithm to handle the inference procedure. In the *E-step*, as we consider the current values of the hyper-parameters known, we establish a variational formulation by assuming the factorisation $q(\{\mathbf{Z}_i\}_i, \{\mu_k(\cdot)\}_k) = q_{\mathbf{Z}}(\{\mathbf{Z}_i\}_i) q_{\boldsymbol{\mu}}(\{\mu_k(\cdot)\}_k)$, where $q(\cdot)$ represents the approximation to the true hyper-posterior. For all $k = 1, \ldots, K$, we deduce in Proposition 4.2 the variational distribution for μ_k, which remains analogous to the expression established in Chapter 3. These quantities are computed iteratively along with the variational distributions for $\{\mathbf{Z}_i\}_i$ variables, providing for all individuals $i = 1, \ldots, M$, the updated probabilities to belong to each cluster, defined as:

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

The *M-step* remains roughly similar as before, and we derive in Proposition 4.3 four different optimisation formulas for each corresponding model assumptions. Explicit gradients associated with the functions to maximise can be derived for facilitating the optimisation procedure in practice. Alternatively repeating E and M steps until convergence we still obtain estimates for the desired quantities, enabling the subsequent derivation of approximated versions of the multi-task GP prediction formulas previously introduced.

Once more, if we observe a new individual, called $*$, at timestamps $\mathbf{t}_*$ for whom we want to predict its output values at timestamps $\mathbf{t}^p$, we shall define the corresponding pooled grid $\mathbf{t}^p_*$. Depending on the model assumptions, an EM algorithm may need to be derived to estimate

the hyper-parameters associated to the new individual along with its probabilities $\{\tau_{*k}\}_k$ of belonging in each cluster. In some particular cases, we may only have to compute the updated proportions of the mixture from explicit expressions. Afterwards, we first integrate out the mean processes $\{\mu_k(\cdot)\}_k$ in order to get the multi-task prior distributions for the new individual, namely:

Proposition 1.2. *For a set of timestamps $\mathbf{t}_*^p$, the multi-task prior distribution of y_* knowing its clustering latent variable is given by:*

$$p(y_*(\mathbf{t}_*^p) \mid \mathbf{Z}_*, \{\mathbf{y}_i\}_i) = \prod_{k=1}^{K} \mathcal{N}\left(y_*(\mathbf{t}_*^p); \hat{m}_k(\mathbf{t}_*^p), \widehat{\mathbf{\Gamma}}_{*k}^{\mathbf{t}_*^p}\right)^{Z_{*k}}.$$

Then, the corresponding cluster-specific multi-task posteriors can be derived in the same manner as previously:

$$p(y_*(\mathbf{t}^p) \mid Z_{*k} = 1, y_*(\mathbf{t}_*), \{\mathbf{y}_i\}_i) = \mathcal{N}\left(y_*(\mathbf{t}^p); \hat{\mu}_{*k}(\mathbf{t}^p), \widehat{\mathbf{\Gamma}}_{*k}^{\mathbf{t}^p}\right), \ \forall k,$$

where:

- $\hat{\mu}_{*k}(\mathbf{t}^p) = \hat{m}_k(\mathbf{t}^p) + \mathbf{\Gamma}_k^{\mathbf{t}^p\mathbf{t}_*}{\mathbf{\Gamma}_k^{\mathbf{t}_*\mathbf{t}_*}}^{-1}\left(y_*(\mathbf{t}_*) - \hat{m}_k(\mathbf{t}_*)\right), \ \forall k,$
- $\widehat{\mathbf{\Gamma}}_{*k}^{\mathbf{t}^p} = \mathbf{\Gamma}_k^{\mathbf{t}^p\mathbf{t}^p} - \mathbf{\Gamma}_k^{\mathbf{t}^p\mathbf{t}_*}{\mathbf{\Gamma}_k^{\mathbf{t}_*\mathbf{t}_*}}^{-1}\mathbf{\Gamma}_k^{\mathbf{t}_*\mathbf{t}^p}, \ \forall k.$

To conclude, by integrating out the variable Z_* we can formulate the final posterior distribution as a weighted sum of the cluster-specific predictions:

Proposition 1.3. *The multi-task GPs mixture distribution for $y_*(\mathbf{t}^p)$ takes the form below:*

$$p(y_*(\mathbf{t}^p) \mid, y_*(\mathbf{t}_*), \{\mathbf{y}_i\}_i) = \sum_{k=1}^{K} \tau_{*k} \ \mathcal{N}\left(y_*(\mathbf{t}^p); \hat{\mu}_{*k}(\mathbf{t}^p), \widehat{\mathbf{\Gamma}}_{*k}^{\mathbf{t}^p}\right).$$

We called MagmaClust the overall algorithm implementing this method.

	MSE	$WCIC_{95}$	Training time	Prediction time
GP	138 (174)	78.4 (31.1)	0 (0)	0.6 (0.1)
Magma	31.7 (45)	84.4 (27.9)	61.1 (25.7)	0.5 (0.2)
MagmaClust	**3.7 (8.1)**	**95 (13.2)**	132 (55.6)	0.6 (0.2)

Table 1.2 – Average (sd) values of MSE, $WCIC_{95}$, training and prediction times (in secs) on 100 runs for GP, Magma and MagmaClust.

This approach takes advantage of the multiple mean processes for handling datasets presenting group structures. We provide a comparison to alternatives both in regards to the clustering ability (Figure 1.8) and the predictive performances (Table 1.2). Finally, we bring a final touch to the swimmers' progression curves problem, the associated clustering, and the probabilistic forecast of future performances. Our approach proves to be particularly efficient, and an example of application on both men and women is displayed on Figure 1.9. This contribution gathers in one method the different aspects scanned during the thesis, providing both a satisfactory answer to the initial applicative issue and a significant methodological contribution we hope to be useful for working on related problems.

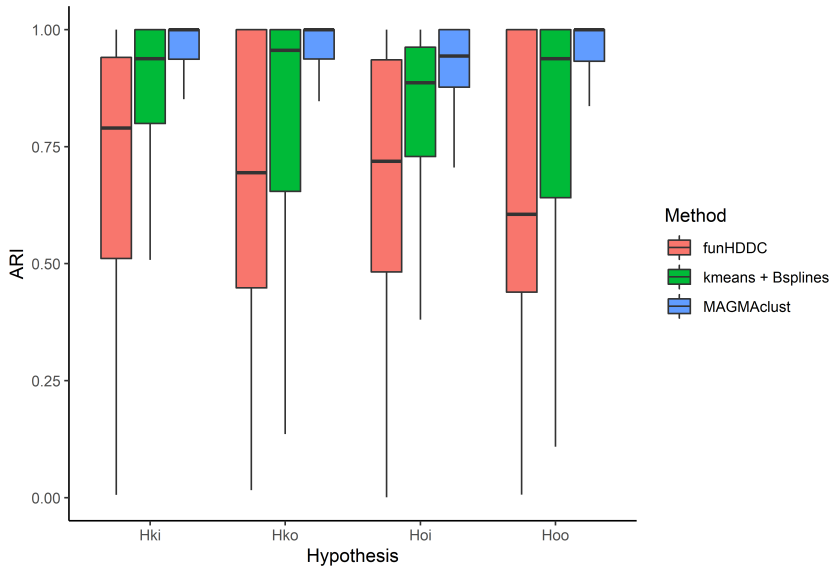

Figure 1.8 – Rand Index values between the true clusters and the partitions estimated by kmeans, funHDDC, and MagmaClust. The value of K is set to the true number of clusters for all methods. The RI is computed on 100 datasets for different generating model's assumptions $\mathcal{H}_{ki}, \mathcal{H}_{k0}, \mathcal{H}_{0i}$, and $\mathcal{H}_{00}$.

1.2.5 Published articles and preprints

The work presented in this manuscript led to one publication (Leroy et al., 2018) (Chapter 2), and two articles currently under review (Leroy et al., 2020b,a) (Chapters 3 and 4). Besides, two additional papers (Moussa et al., 2019; Pla et al., 2019) have been co-written and published during this thesis on sports-science topics, distant from those of the present document. Let us provide below the detailed list of publications:

- A. Leroy, A. Marc, O. Dupas, J. L. Rey, and S. Gey. Functional Data Analysis in Sport Science: Example of Swimmers' Progression Curves Clustering. *Applied Sciences*, 8 (10):1766, Oct. 2018. doi: 10.3390/app8101766

- A. Leroy, P. Latouche, B. Guedj, and S. Gey. MAGMA: Inference and Prediction with Multi-Task Gaussian Processes. *PREPRINT arXiv:2007.10731 [cs, stat]*, July 2020b

- A. Leroy, P. Latouche, B. Guedj, and S. Gey. Cluster-Specific Predictions with Multi-Task Gaussian Processes. *PREPRINT arXiv:2011.07866 [cs, LG]*, Nov. 2020a

- I. Moussa, A. Leroy, G. Sauliere, J. Schipman, J.-F. Toussaint, and A. Sedeaud. Robust Exponential Decreasing Index (REDI): Adaptive and robust method for computing cumulated workload. *BMJ Open Sport & Exercise Medicine*, 5(1):e000573, Oct. 2019. ISSN 2055-7647. doi: 10.1136/bmjsem-2019-000573

- R. Pla, A. Leroy, R. Massal, M. Bellami, F. Kaillani, P. Hellard, J.-F. Toussaint, and A. Sedeaud. Bayesian approach to quantify morphological impact on performance in

international elite freestyle swimming. *BMJ Open Sport & Exercise Medicine*, 5(1): e000543, Oct. 2019. ISSN 2055-7647. doi: 10.1136/bmjsem-2019-000543

1.2.6 Implementations

The algorithms described in Chapter 3 and Chapter 4 have been implemented into R packages that constitute the practical contributions of the present thesis. The current versions of the codes are freely available at the following addresses:

- MAGMA: https://github.com/ArthurLeroy/MAGMA,
- MAGMACLUST: https://github.com/ArthurLeroy/MAGMAclust.

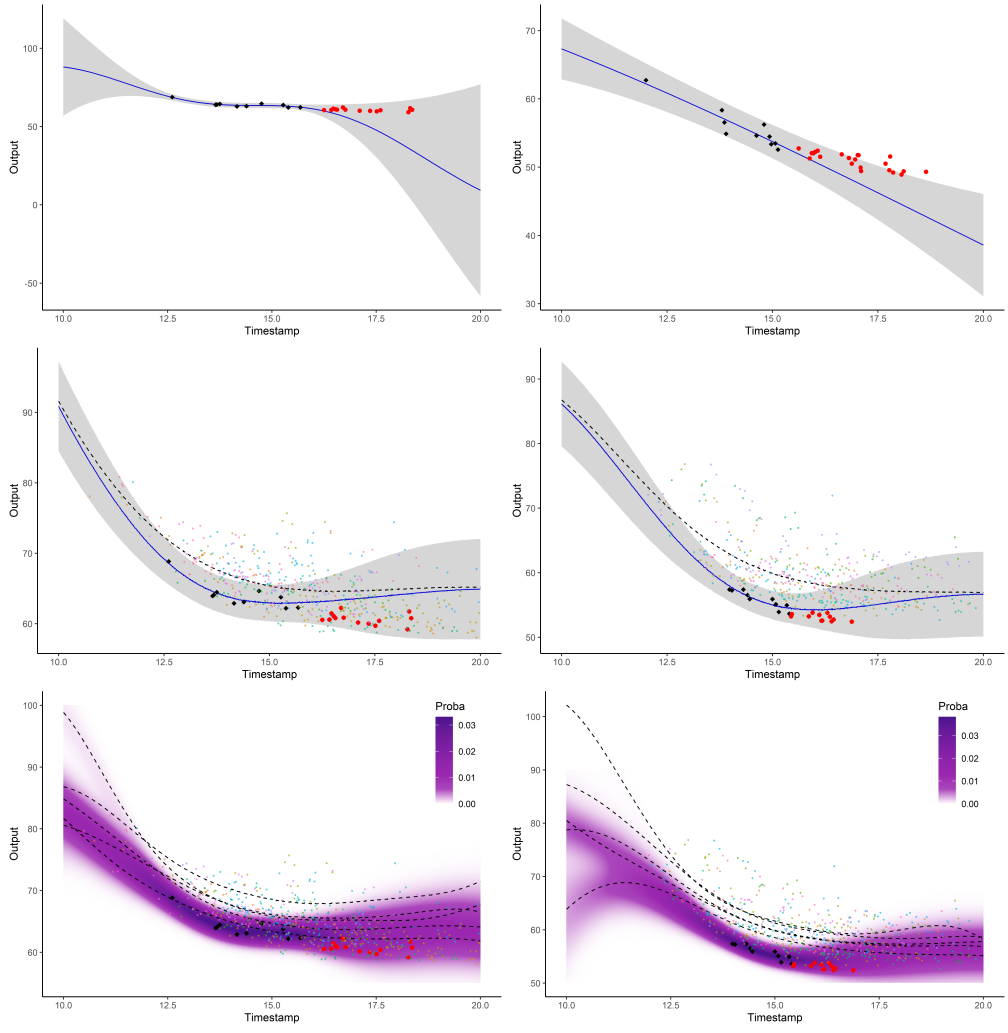

Figure 1.9 – Left: women dataset. **Right:** men dataset. Prediction and uncertainty obtained through GP (top), MAGMA (middle), and MAGMACLUST (bottom) for a random swimmer. The dashed lines represent the mean parameters from the mean processes estimates. Observed data points are in black, testing data points are in red. Backward points are the observations from the training dataset.

1.3 Perspectives

Extension to more general input variables

Whereas the models introduced in Shi et al. (2007) and Shi and Choi (2011) do not apply with irregular timestamps and lack an uncertainty quantification, they somehow deal with a more general framework in terms of input variables. In many medical applications proposed by the authors, enabling the introduction of different types of functional or scalar input variables often constitutes a useful improvement. As we currently only account for the influence of timestamps on the output values, our multi-task approach might naturally take advantage of sharing more information across the individuals.

Online version

In our approach, most of the computing time concentrates on learning the mean process. As we previously evoked the appetite for online algorithms in recent learning applications, it would be interesting to derive formulas allowing for a fast update of μ_0's hyper-posterior when a new individual is added to the training set. Since computing the hyper-posterior can already be seen as iterative with respects to the individuals, this issue seems reasonably manageable in the case where we would let the hyper-parameters unchanged. However, for the exact trade-off that we would need to set between computing time and memory usage, and the possibility to update the hyper-parameters quickly as well, the door remains open to further developments.

Sparse approximations

Although we do not study this matter thoroughly, the scalability of GPs with large datasets remains of paramount importance when it comes to practical implementation. In order to widen the applicability of our algorithms, it would seem valuable to adapt at least one of the sparse approximations that have been proposed in the literature (Snelson and Ghahramani, 2006; Bauer et al., 2016). In particular, the variational approach introduced in Titsias (2009) to selects the pseudo-inputs along with hyper-parameters learning would prove useful in the estimation of our mean processes $\{\mu_k(\cdot)\}_k$. The inference procedure in the case of MagmaClust already relying on a variational approach, it might be tempting to combine both approximations. The more recent works about variational stochastic inference (Hensman et al., 2013) would also be worth a look for this purpose.

Multi-task modelling of the covariance structure

As previously mentioned, the *multi-task* term in the GP literature mostly refers to the way of modelling the covariance structure. In our work, we keep the covariance functions as simple as possible to focus on the mean process, but using a more elaborate structure as in Bonilla et al. (2008) might bring the best of both worlds in a multi-task perspective. Defining explicit relationships across individuals through a covariance matrix would add a shared-features aspect to our underlying common mean framework. However, this method suffers from a high computational complexity and might slow down our current approach. Besides, the different assumptions proposed in our model for the hyper-parameters already define a multi-task aspect regarding the covariance, and we could easily imagine to extend this feature to sharing more general structures of covariance across individuals.

Dedicated model selection tools

The matter of model selection happens to be of primary importance in practical implemen-

tations, for instance when it comes to choose the number of groups in clustering problems. Tackling this issue, which is required in our GPs mixture model, is generally non-trivial and many approaches have been developed in this sense over the years. As we mainly work with tractable likelihoods, the adaptation of efficient heuristics to develop specific model selection tools seems achievable, although the presence of multiple latent processes needs to be carefully dealt with. In particular, under the assumption of iid individuals within the clusters (hypotheses $\mathcal{H}_{00}$ and $\mathcal{H}_{k0}$ in Chapter 4), we could probably expect to reach an *ICL* (Biernacki et al., 2000) formulation. However, when lacking this property (hypotheses $\mathcal{H}_{0i}$ and $\mathcal{H}_{ki}$ in Chapter 4), this approach seems off the table, and we shall certainly seek an efficient heuristic in the kind of the *slope heuristic* (Birgé and Massart, 2006).

Study of swimmers' progression curves

The collaboration with the French Swimming Federation (FFN) has proved fruitful over the past three years and should continue with the implementation of our algorithm within their performance analysis software. Some work remains necessary to allow for fast online predictions once the model is pre-trained on a fine grid of timestamps. Furthermore, the automatic update of the training database and the subsequent re-computation of the model's parameters still need to be implemented. Regardless of such technical details, we are currently discussing whether to organise prospective forecasts with long-term follow-up of young swimmers, for real-life validation purpose.

2

Clustering of the swimmers' progression curves

2.1 Introduction

The following work proposes a motivational introduction to the applied problematic of the present thesis. It aims at providing a first explanatory study of the swimmers' performance dataset that led to the further methodological contributions, developed afterwards. This chapter is based on the article Leroy et al. (2018) that was written as an introduction to functional data analysis (FDA) methods, with a focus on curve clustering, for sport-scientist practitioners. The purpose of this work is twofold. First, we propose a presentation of several curve clustering algorithms along with a practical comparison on simulated data within a convenient implemented framework. Secondly, we illustrate in more details the preliminary work of data exploration and the complete procedure of curve clustering on our sport-related applicative example.

2.2 Comparison of curves clustering algorithms

2.2.1 Presentation of the methods

As an introduction to the matter of clustering functional data in practice, a comparative study between several usual algorithms is proposed on simulated data. Below, we provide a few information on the methods being compared, along with references to the corresponding papers, and recall into which family they belong according to the classification evoked in Section 1.1.1.d. For this work, the *R* package *funcy*, which compiles seven state-of-the-art algorithms, has been used for comparison purpose. This implementation gives a common syntax to perform clustering on data with functional structure, and the different approaches for managing such a task follows.

Distance-based algorithms:

- *distclust*(Peng and Müller, 2008): An approximation of the L^2 distance between curves is defined, and a k-means heuristic makes use this measure to handle the functional data. This method is well designed in the context of sparsely observed functions with irregular measurements.

Model-based algorithms:

- *fitfclust*(James and Sugar, 2003): One of the first algorithm to use a Gaussian mixture model for clustering univariate functions. This heuristic holds for all following algorithms described as Gaussian mixture methods. Functions are represented using basis functions, and the associated coefficients are supposed to come from Gaussian distributions. Given a number K of different means and covariances parameters corresponding to the K clusters, an EM algorithm is used to estimate the probability of each observational curves to belong to a cluster. After convergence (various stopping criteria exist), an individual is affected to its most likely cluster. A preliminary step of FPCA can be added to work on lower-dimensional vectors and thus offering a sparse representation of the data. The algorithm *fitfclust* is assumed to perform well in the context of sparsely observed functions.

- *iterSubspace*(Chiou and Li, 2007): A non-parametric method based on a random-effect model. This approach uses the Karhunen-Loeve expansion of the curves, and perform a k-centres algorithm on the scores of FPCA and the mean process. This method can be useful when the Gaussian assumption does not hold, but k-centres approaches are reported to lead to unstable results.

- *funclust* (Jacques and Preda, 2013a): An algorithm based on Gaussian mixture model. This method uses the Karhunen-Loeve expansion of the curves as well. Moreover, it allows for different sizes of expansion's coefficients vector across clusters, according to the quantity of variance expressed by the corresponding FPCA. The algorithm also enables different covariance structures between clusters and thus generalizes some methods such as *iterSubspace*.

- *funHDDC*(Bouveyron and Jacques, 2011): An algorithm based on a Gaussian mixture model. This method presents many common characteristics with *funclust* but additionally allows for clustering multivariate functions. The algorithm also provides six assumptions on the modelling of covariates structures, especially to deal with the extra dimension of the FPCA.

- *fscm* (Jiang and Serban, 2012): A non-parametric model-based algorithm. Each cluster is modelled by a Markov random field, and functions are clustered by shape regardless of the scale. Observation curves are considered as locally-dependent, and a K-nearest neighbours heuristic is proposed to define the proximity structure. Then, an EM algorithm estimates the parameters of the model. This method is well designed when the assumption of independence between curves does not hold.

- *waveclust* (Giacofci et al., 2013): An algorithm designed from a linear Gaussian mixed effect model. This approach performs dimension reduction using wavelet decomposition (rather than classic FPCA). An EM algorithm is derived to compute parameters of the model and probabilities to belong to each cluster. This method is well-suited for high-dimensional curves when variations such as peaks appear in data, and thus wavelets perform better than splines.

Unfortunately, the available version (1.0.0) of the *funcy* package encounters troubles with the *funHDDC* implementation, which is not currently supported. All the remaining algorithms were applied on three simulated data sets, with $K = 4$ groups. The resulting clusterings are compared to real group distributions using the Rand Index (RI)(Rand, 1971). This measure, given as a real number between 0 and 1, accounts for the according pairs of individuals between the different partitions of a data set. The RI values are provided to highlight the ability of each procedure to retrieve the actual groups. Then, graphs of centres of each curve clusters are displayed to analyze the consistency of the resulting groups according to the ones of synthetic data.

2.2.2 Description of the simulated datasets

Three different datasets have been simulated to test the algorithms of the *funcy* package on different contexts. We used the included function *sampleFuncy* that provides a convenient way to simulate datasets suited for direct application of the aforementioned methods. The synthetic datasets are sampled from four different processes of the form $f(t)+\varepsilon$, with f and ϵ detailed in Table 2.1 below. For each process, 25 curves are simulated, thereby leading to 100 curves in each sample. The following clustering procedure aims to gather themselves curves that correspond to the same underlying process. An additional goal would be to retrieve, at least approximately, the shapes of the underlying functions f that generated each data curves within a cluster. Speaking rather loosely, *Sample 1* depicts a straightforward situation with low noise and well-separated processes, whereas *Sample 2* represents the same processes in a higher variance context. Finally, *Sample 3* corresponds to a high-noise and crossing processes context, which is designed to be trickier. Moreover, in the case of *Sample 3*, observations of the curves are irregular on t-axis, and thus, we had to proceed to a previous fitting step for three out of six algorithms of the package that are not implemented in this case. The function *regFuncy* of the package is used for this purpose.

Data set	Functions	Noise	Grid on t-axis
Sample 1	$t \longmapsto t-1$	$\varepsilon \sim \mathcal{N}(0, 0.05)$	10 *regular* points
	$t \longmapsto t^2$		
	$t \longmapsto t^3$		
	$t \longmapsto \sqrt{t}$		
Sample 2	$t \longmapsto t-1$	$\varepsilon \sim \mathcal{N}(0, 0.1)$	10 *regular* points
	$t \longmapsto t^2$		
	$t \longmapsto t^3$		
	$t \longmapsto \sqrt{t}$		
Sample 3	$t \longmapsto t-1$	$\varepsilon \sim \mathcal{N}(0, 0.5)$	$\leq$ 10 *irregular* points
	$t \longmapsto -t^2$		
	$t \longmapsto t^3$		
	$t \longmapsto sin(2\pi t)$		

Table 2.1 – Details on the simulated samples. Processes are defined as $f(t) + \varepsilon$ with 4 different functions f in each sample and a varying noise ϵ.

2.2.3 Comparative study

The Table 2.2 below displays the results of the comparison between the six studied algorithms. These values remain mainly illustrative, and we acknowledge that the quality of a clustering algorithm cannot fully be addressed through simulation. However, it can give some clues on the type of situations where algorithms seem to perform properly or not. The *Sample 1* was designed to be easy to manage, and most model-based algorithms perform well as expected. Nevertheless, they are outperformed by the only distance-based method *distclust* gives almost perfect results. As *Sample 2* proposes a noisier version of *Sample 1*, the problem becomes harder and results slightly decrease. Let us notice that, although the stochastic processes we sampled from are identical to *Sample 1*, the relative performances between methods change. This might indicate differences at noise robustness between the methods. For example, performances of the *fcsm* algorithm decrease only slightly compared to *distclust*. Finally, as expected, the results deteriorate greatly on the fuzzy situation of *Sample 3*. Only three methods achieve moderate performances, and we may note that there is an algorithm of both families among them. Although Table 2.2 informs us of the clustering performances, it does not give information on the ability of the methods to retrieve the actual shape of the underlying functions. To this end, the following graphs provide some an illustration on this aspect according to the best performing method in each context.

Method	Sample 1	Sample 2	Sample 3	Running speed
fitfclust	0.945 (0.14)	0.857 (0.01)	0.307 (0.06)	2.8
distclust	**0.996 (0.01)**	0.888 (0.05)	0.523 (0.07)	19.2
iterSubspace	0.938 (0.14)	0.850 (0.12)	**0.527 (0.07)**	**1**
funclust	0.450 (0.17)	0.418 (0.16)	0.084 (0.07)	**1**
fscm	0.948 (0.12)	**0.902 (0.01)**	**0.527 (0.07)**	7
waveclust	0.920 (0.12)	0.810 (0.01)	0.324 (0.13)	34

Table 2.2 – Mean Rand Index and (Standard Deviation) on 100 simulations of the tree samples. Each algorithm runs in at most few seconds on our simulated data sets. Comparison in speed between algorithms is given as a multiple of the fastest which is set arbitrarily to 1.

Figure 2.1 proposes a representation of the *Sample 1* curves along with cluster centres coming from the *distclust* algorithm. As expected, *Sample 1* remains quite simple to deal with, since curves of different groups are well separated. Moreover, the *distclust* clustering algorithm satisfyingly figures out the actual shape of each underlying function.

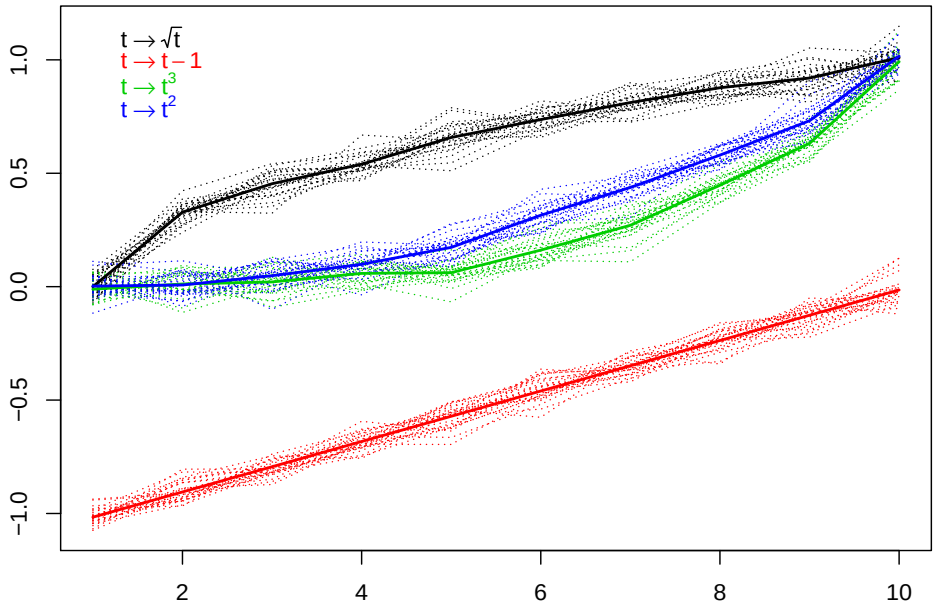

Figure 2.1 – All curves (dotted lines) and cluster centres curves (plain lines) obtained with *distclust* algorithm for *Sample 1*. The algorithm correctly clusters curves and retrieves the underlying shapes of generating functions.

Furthermore, we can notice on Figure 2.2 that, although the noisier situation of *Sample 2* affects the RI scores, the shapes of the generating functions remain correctly approximated by clusters centres of *fscm*.

The *Sample 3* was designed to be trickier since curves cross each other and the overall signal thus appears rather noisy. In this context, Figure 2.3 reveals that the *iterSubspace* algorithm still retrieve approximately the true shapes of the underlying functions. However, while the sinus function (in black) seems correctly identified, the method struggles to

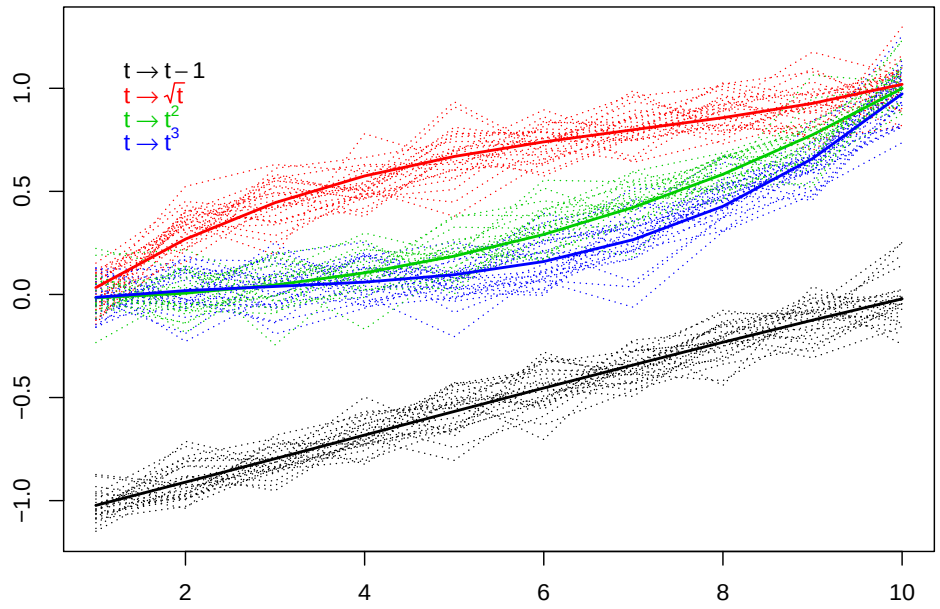

Figure 2.2 – All curves (dotted lines) and cluster centres curves (plain lines) obtained with *fscm* algorithm for the simulated *Sample 2*. Clustering becomes more difficult between curves (e.g. blue and green curves) but the algorithm still performs well to figure out the underlying shapes.

separate the polynomial functions.

2.3 Clustering swimmers' progression curves

2.3.1 Description of the dataset

The real data presented in this work have been collected by the French Swimming Federation. It gathers all the performances in competition of french male swimmers, since 2002, for the 100m freestyle in a 50m pool. Because of confidentiality issues, the names of athletes are replaced by identifying numbers. The data set is composed of 46115 values of performance and age for 1468 different swimmers. Raw data consists of time series where the racing time constitutes the output variable associated with the age of the swimmer as input. The number of competitions and the age at which swimmers participate differs from one to another, leading to strongly uncommon grids of timestamps. This particularity of the dataset (as well as the ability to work on derivatives) led to model the observations as functions rather than time series. Thus, a preliminary step of fitting is performed to extract the functional nature of the data and deal with the random fluctuations in the observations. All the algorithms were run on the *R* software, and the corresponding packages are named in the sequel.

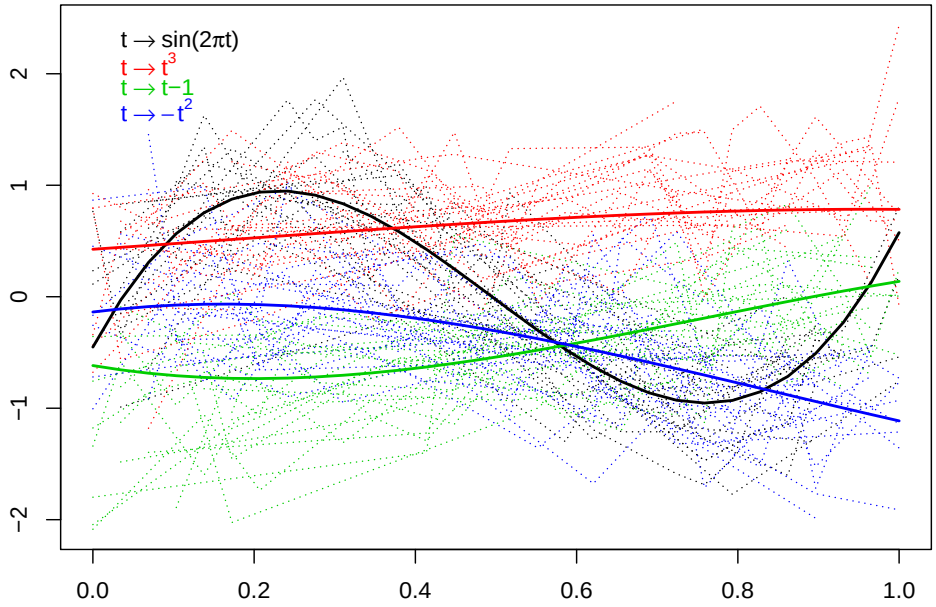

Figure 2.3 – All curves (dotted lines) and cluster centres curves (plain lines) obtained with *iterSubspace* algorithm for the simulated *Sample 3*. Both clustering and detecting underlying shapes become difficult. The high noise makes the clustering fuzzy, which is affecting the central cluster curves.

2.3.2 Methodology

As mentioned above, the real data set is very irregular, with no accordance in time and number of measurements between athletes. Thus, the first step of the analysis aims at defining a common representation for data through a smoothing procedure. According to the non-periodic form of the curves and the relatively low sampling of observational points (around 30) for each athlete, a B-spline basis is chosen. The study focuses on the age period from 12 to 20 years old, which is crucial in the progression phenomenon that we aimed at studying. A basis of seven B-splines of order 4 was defined so that the derivatives remain smooth enough to work on derivatives. Since we do not wish to focus on a specific time period, the knots are equally placed on ages 13 to 19. Let us note that data are considered as realizations of underlying stochastic processes, and thus raw data are assumed to contain random fluctuations. The function that is fitted using the B-spline basis should represent the true signal properly, although the well known over/under-fitting issue may appear in this case. In order to differentiate the true signal from the noise, several methods can be used, knowing that there is always a trade-off between smoothness of the function and closeness to data points. A classical approach consists of using a penalization term in the least-square fitting calculation, and the signal-on-noise ratio would be controlled by a unique hyper-parameter. In our case, a cross-validation criterion is used to compute an optimal value for this hyper-parameter, and the resulting functional data were considered as coherent by swimming experts. This whole fitting procedure was performed thanks to *R* (version 3.5.0) software, and especially the *fda* (version 2.4.8) package. To efficiently manage a real dataset, a thorough exploration step generally helps to figure out the more suited algorithms for the

analysis, in particular when dealing with infinite-dimensional objects like functions. For this purpose, an FPCA is performed on the progression curves and their derivatives separately. By inspection of the percentage of variance explained by each eigenfunction and the shapes of them, the main modes of variations of the curves can be highlighted. One can see on Figure 2.6 in the appendix that main variations among the levels of performance appear at young ages and a clustering procedure on the sole progression curves tends to simply group individuals according to this criterion. As displayed on Figure 2.7, first eigenfunctions of the derivatives represent three different modes of variations localized at young, middle, and older ages. These characteristics of data would be relevant to include to the clustering procedure besides the level of performance information. To this end, the *funHDDC* algorithm is used as clustering procedure, since this is one of the rare implemented methods that works in a multivariate case and thus allow us to consider both curves and their derivatives simultaneously. Let us refer to the subsequent section for more details about the reasons for this choice. Although implemented in the *funcy* package, we choose to work with the original *funHDDC R* package, because of the current implementation issues. Several features of the package are used, such as Bayesian Information Criterion (BIC), Integrated Classification Likelihood (ICL) and slope heuristic, to deal with the problems of model selection and choice of the number K of clusters. Since no particular assumptions were made on the covariance structure or the number of clusters form a sports expert point of view, the hyper-parameters of the model have been optimized from data. All available models for *funHDDC* are computed for different values of K and the best one (the sense of the term *best* is developed in Section 2.3.3) is retained as our result clustering. In the *funHDDC* algorithm, each cluster is considered to be fully characterized by a Gaussian vector, from which scores on eigenfunctions of the FPCA are assumed to come. Thus, the clustering procedure becomes a likelihood maximization problem that aims at finding the adequate values of means and covariance matrices fitting the best to data, along with probabilities for each of data curves to belong to a cluster. Since all parameters influence the values of each other, this classical issue is addressed thanks to an Expectation-Maximization (EM) algorithm that computes approximations of optimal parameters efficiently. At the end of the procedure, a data curve is considered to belong to the cluster within which it has the highest probability to come from. The clustering is performed on the curves and their derivatives separately at first. Then, the resulting clusters are compared thanks to the Adjusted Rand Index (ARI) Rand (1971), which is an extended version of the RI to partitions with a different number of clusters. This measure allows us to quantify the adequacy between groups defined whether by a clustering the progression curves or the derivatives. Note that many other indices exist, such as Silhouette index (Rousseeuw, 1987) or Jaccard index (Rogers and Tanimoto, 1960) for example. Although our results were quite comparable using one or another, an extensive comparative study of the different indexes can be found in Arbelaitz et al. (2013). Noticing that athletes are clustered differently according to the situation, providing two types of information, a third clustering procedure is proposed. This time, the multivariate clustering version on the *funHDDC* algorithm is used. The term *multivariate* refers here to a clustering algorithm that deals with multidimensional functions. The progression curves are defined as a functional variable, while the derivatives are another. Finally, the results were analyzed and discussed with swimming experts to confront the computed clusters with practical knowledge on this matter.

2.3.3 Results of the curve clustering

The choice of the *funHDDC* algorithm was motivated by two main arguments. First, it provides a flexible method that has been shown efficient in various cases (Bouveyron et al., 2018; Martínez-Álvarez et al., 2019). Secondly, because of the results of the FPCA performed to explore the data set. As presented on the top graph on Figure 2.6, we notice that the underlying dimension of the data seems lower than the original one: most of the variance in the dataset can be expressed according to only two eigenfunctions. An analogous result with three underlying modes on variation for the derivatives is displayed on Figure 2.7. Thus, it seems natural to work with an FPCA-based method to efficiently account for this sparsity property. Furthermore, *FunHDDC* provides a flexible way to deal with the "extra-dimensions", proposing six sub-models, corresponding to different assumptions on the structure of covariance matrices. As advised by the authors in Schmutz et al. (2018), the BIC (Schwarz, 1978) is used for model selection, whereas the choice of the appropriate number of clusters relies on the slope heuristic (Birgé and Massart, 2006; Arlot, 2019). According to these criteria, the most suited models are composed of 5 clusters for the progression curves alone, and 4 clusters when working solely on derivatives. Resulting cluster centres are represented on Figure 2.8 and Figure 2.9 respectively. At this stage, the Adjusted Rand Index (ARI) is used to compare theses two different ways to regroup athletes and give a value of 0.41. The value of ARI would be around 0.20 for a completely random clustering procedure. This result, far from an ARI equals to 1 that would indicate complete adequacy, lets us think that the curves and derivatives imply different data features when it comes to building the clusters in each context. Swimming experts highlighted that the clustering on progression curves mainly regroup the athletes according to their final level of performance. In contrast, the derivatives clustering seems to gather individuals presenting similar trends of progression (at a particular age, or with the same dynamic, for example). These conclusions guided us to the multivariate clustering procedure, for which results are presented on Figure 2.4 and Figure 2.5. Each athlete is represented thanks to its performance curve, where the colour indicates in which cluster it belongs. A close look at the groups and their trends on Figure 2.4 seems to indicate that multivariate clustering clusters combine information both on level of performance and trends of evolution as expected. As the full display of all curves can be tough to analyze, the Figure 2.5 presents the cluster centre curves for a clearer view of the main tendencies.

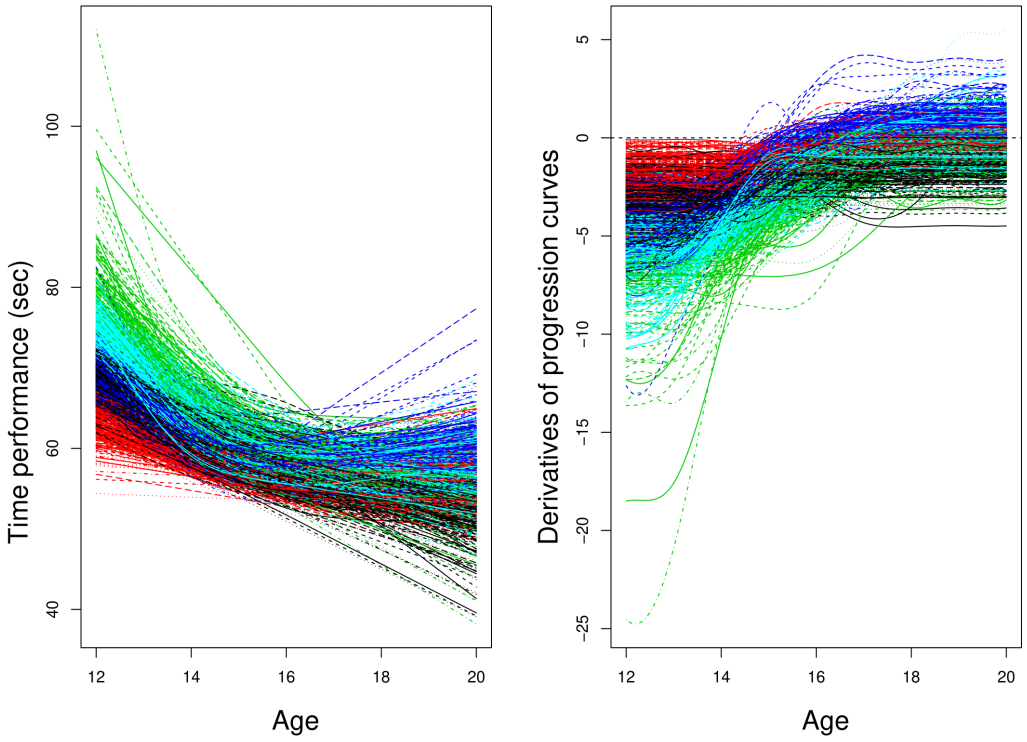

Figure 2.4 – All progression curves of swimmers (left) and derivatives (right) coloured by clusters, obtained with the multivariate *funHDDC* algorithm.

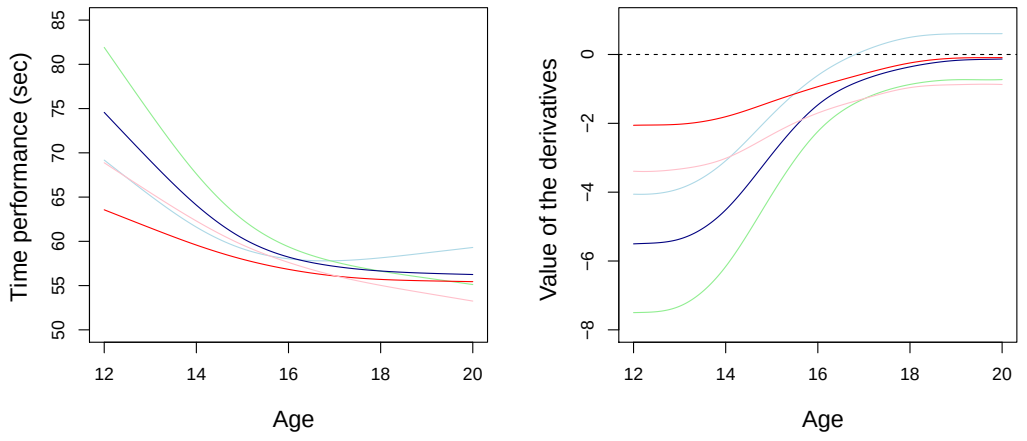

Figure 2.5 – Cluster centres curves of swimmers (left) and derivatives (right) coloured by clusters, obtained with the multivariate clustering *funHDDC* algorithm.

2.4 Discussion

As mentioned in the simulated dataset context, we shall emphasize that no objective criterion might reflect the quality of a clustering procedure correctly. The authors of von Luxburg et al. (2012) recall that all clustering algorithms remain in some way, subjective regarding how they gather individuals or which metric they use. However, it generally offers a new point of view to potentially detect perspectives that might be hidden in the raw data. In this work, we presented some classical methods for curve clustering along with some handy practical implementations. Moreover, the comparative simulation study provides some insights on the particularities of the different algorithms. We may notice that distance-based methods generally lie on simple heuristics and offer rather good results in simple contexts. On the other hand, model-based algorithms are built on more complicated designs, although remaining well suited on a wider range of problems. These valuable performances offer to this family of methods to be of high consideration in the current literature on curve clustering Jacques and Preda (2014) and might partly explain why hey constitutes most of the algorithms implemented in *funcy*. Algorithms using Gaussian mixtures are naturally more flexible than methods like k-means since they might be considered as a generalization with elliptic clusters rather than circular ones. However, let us also stress that this flexibility sometimes comes with additional running time. Even if the EM-based inference provides efficient implementations to manage Gaussian mixture models, the multiplicity of models and the number of clusters to test might results in non-negligible computing time (a few hours in our case). For our purpose, which is to help the swimming federation with the detection of young promising athletes, computational time was not an issue since the aim was more about the long term decision making. Nevertheless, many current sport-related problems need to be solved quickly, or even online, and our methodological choices would have been different under such constraints.

About the results on the swimming dataset, we observe consistent outcomes from both mathematical and sports point of views. Moreover, although our work does not provide predictive results on the progression phenomenon of young swimmers, it still offers some enlightenment of its general pattern along with a practical tool to gather similar profiles. Moreover, using functional data analysis tools, we were able to figure out valuable information from strongly irregular time series. Using smooth functions instead of raw data points provides a first understanding of the main trends and the continuous nature of the progression phenomenon. In order to improve the quality of the approximation, though, the collection of additional data such as training performances would be valuable. Nevertheless, these results might help the detection of promising young athletes with both a better understanding and graphical outcomes to support the decision process. Notice that this work remains descriptive and thus preliminary, but proposes a first step for further predictive analysis. Although we do not discuss here findings concerning any particular swimmer for confidentiality concerns, let us still stress some points that seem interesting to swimming experts. First, as previously mentioned in Boccia et al. (2017) and Kearney and Hayes (2018), it does not seem easy to precisely detect young talents before 16 years because of the high-speed improvement at these ages. However, we can observe between 14 and 16 years old a significant decrease in the value of the derivatives and thus of the speed of progression. Moreover, athletes that seem to perform better at 20 years old are often those who continue to progress, even slightly, after 16 years old. A classical pattern, confirmed with swimming experts, is the presence of a cluster of swimmers who are always among best performers. These athletes are typically often detected and can benefit from

the best conditions to improve their performances. However, two clusters of athletes, often slightly slower than previous ones at young ages, present opposite behaviours. As one group stops its progression early and performs rather modestly at 20 years old, another cluster gathers swimmers with a fast improvement who often perform as good as best swimmers when older. These young athletes are usually thought as the main target for a detection program since they often remain away from top-level structures at young ages.

2.4.1 Further work

While providing first exploratory results by analyzing group structures, the present work lacks predictive results and adequate modelling of the functional data. The irregular and sparse nature of the studied time series often lead to unrealistic reconstructions of functional signals, preventing from efficient forecasting at the scale of one individual. Moreover, the FDA tools presented until now remain of frequentist nature and thus account for uncertainty neither in modelling nor in prediction. Since a probabilistic view appears highly desirable in such a decision-making context, the subsequent chapters aim at providing a new framework for an enhanced analysis, adapted to the problematic we just introduced.

2.5 Appendix

Let us provide in this section a few complementary graphs supporting some of the assertions of the previous analysis on swimmers' progression curves. The selection of the most relevant eigenfunctions and derivatives are displayed along with there associated modes of variations. Besides, the results of the univariate curve clustering for both performance curves and their derivatives is presented as well.

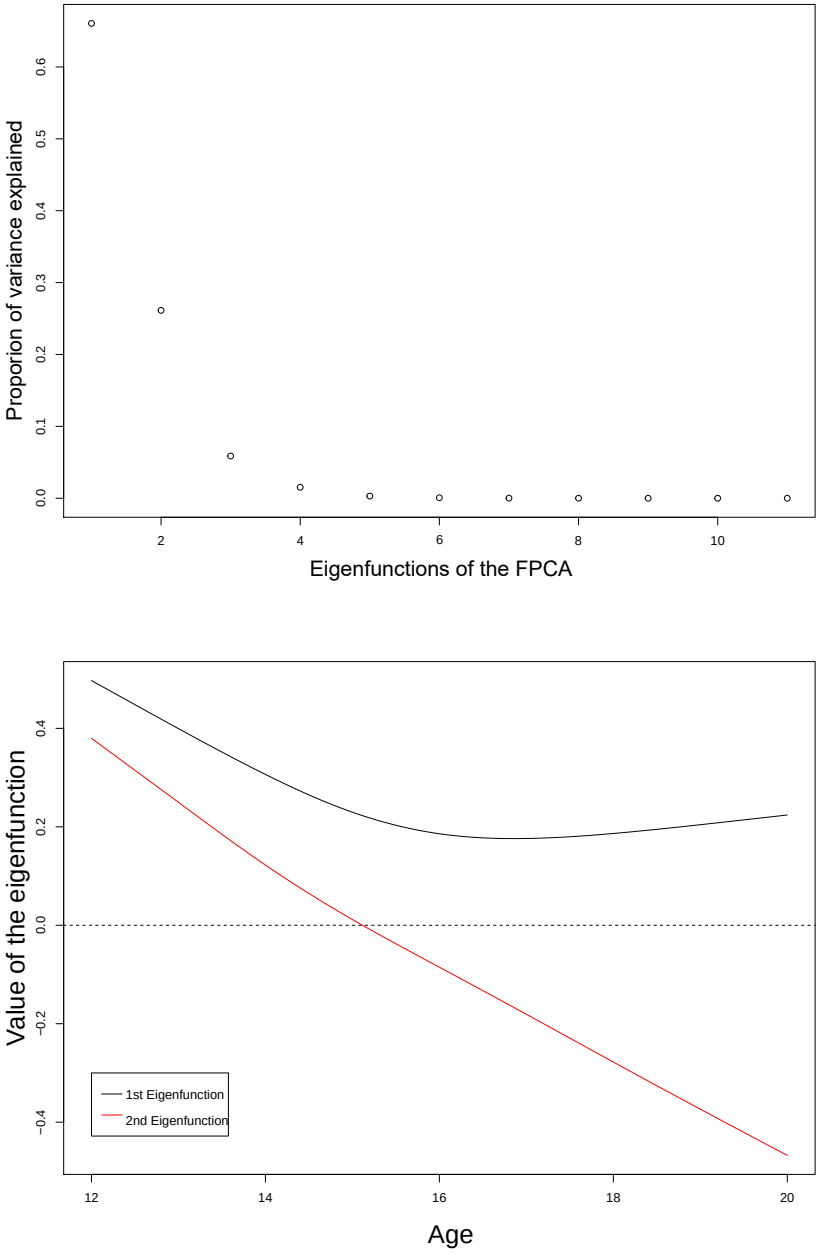

Figure 2.6 – Results of the FPCA on the progression curves. **Top:** Proportion of variance explained by each eigenfunction. With only 2 eigenfunctions, around 90% of the total variance can be expressed. **Bottom:** Values of the two first eigenfunctions. Eigenfunctions are orthogonal each others and display the main modes of variation of the curves. The first eigenfunction mainly informs on differences at young ages, while the second focuses on the opposition between speeds at young and older ages.

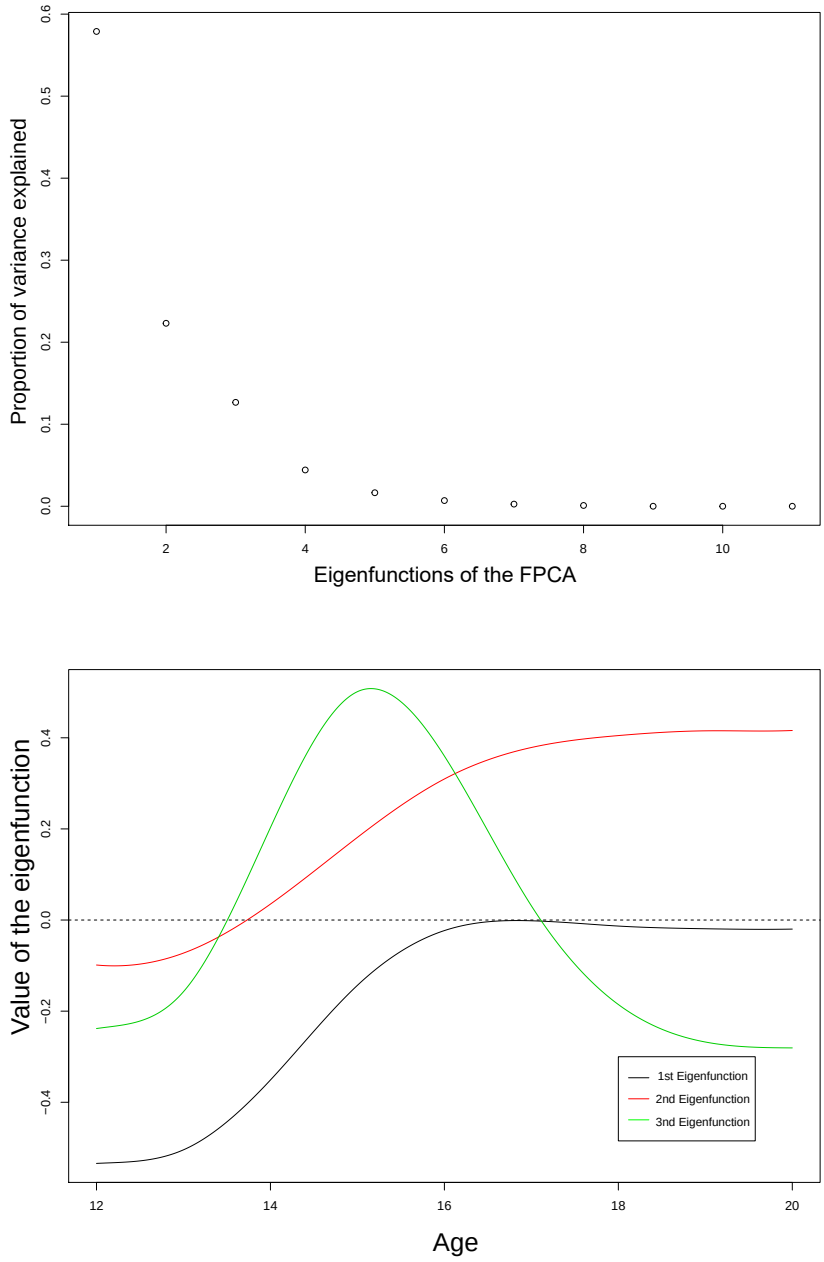

Figure 2.7 – Results of the FPCA on the derivatives of the progression curves. **Top:** Proportion of variance explained by each eigenfunction. With only 3 eigenfunctions, around 90% of the total variance can be expressed. **Bottom:** Values of the three first eigenfunctions. Eigenfunctions are orthogonal each other and display the main modes of variation of the curves. The first eigenfunction mainly informs on derivative differences at young ages, while the second focuses on the behaviour between 14 and 18 years old. The third eigenfunction expresses the differences of swimmers improvement between the middle and the bounds of the time interval.

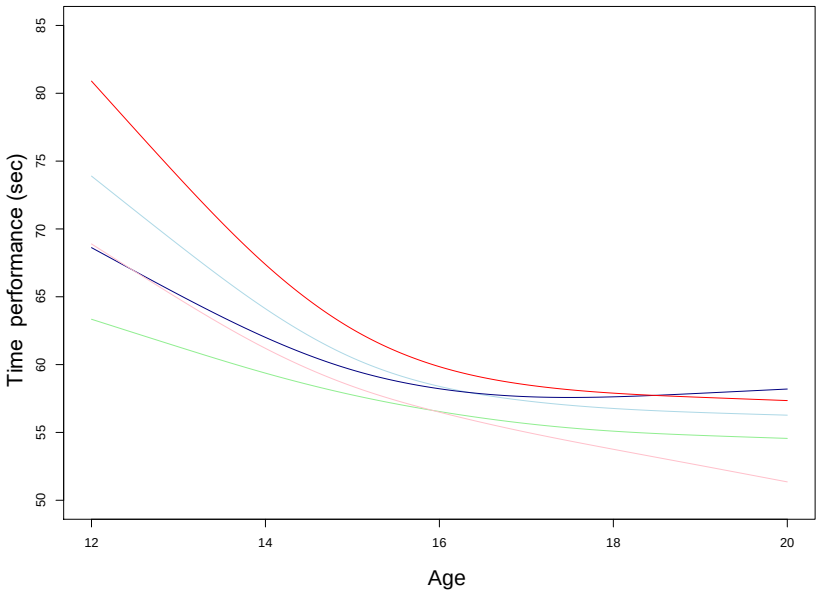

Figure 2.8 – Clusters centres of the progressions curves computed with the univariate *funHDDC* algorithm.

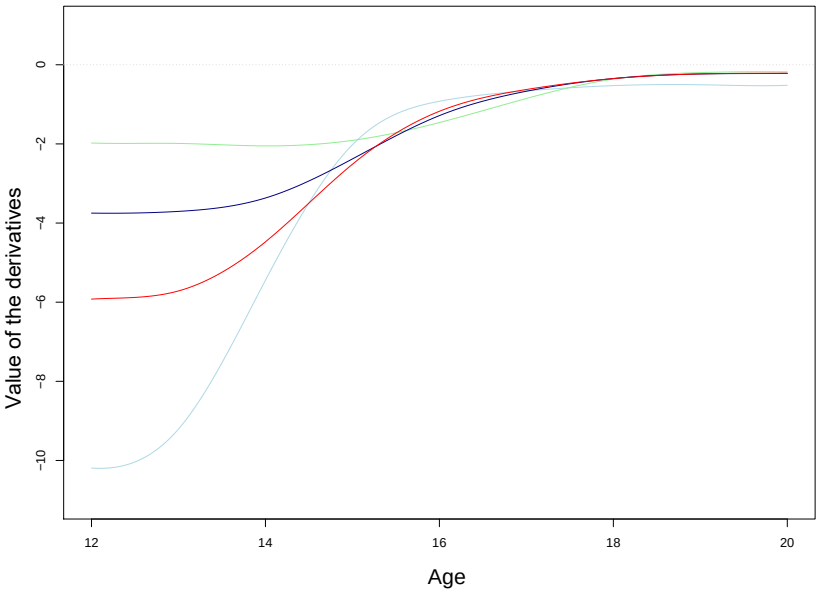

Figure 2.9 – Clusters centres of the derivatives of the progressions curves computed with the univariate *funHDDC* algorithm. Clusters mostly differ on the value of the derivatives at young age and converge all to 0 at 20 years old.

3

Multi-task Gaussian processes with common mean process

This chapter is based on the article Leroy et al. (2020b), which is currently under review.

3.1 Introduction

Gaussian processes (GPs) are a powerful tool, widely used in machine learning (Bishop, 2006; Rasmussen and Williams, 2006). The classic context of regression aims at inferring the underlying mapping function associating input to output data. In a probabilistic framework, a typical strategy is to assume that this function is drawn from a prior GP. Doing so, we may enforce some properties for the function solely by characterising the mean and covariance function of the process, the latter often being associated with a specific kernel. This covariance function plays a central role and GPs are an example of kernel methods. We refer to Álvarez et al. (2012) for a comprehensive review. The mean function is generally set to 0 for all entries assuming that the covariance structure already integrates the desired relationship between observed data and prediction targets. In this section, we consider a multi-task learning framework with a series of Gaussian processes sharing a common mean function. We demonstrate that modelling this function can be key to obtain relevant predictions.

Related work

A major drawback of GPs lies in the $\mathcal{O}(N^3)$ computational cost of the training step, where N denotes the number of observations in the training sample. Many approaches to mitigate this problem with sparse approximations have been proposed in the last two decades. One of the most popular methods can be found in Snelson and Ghahramani (2006), introducing elegant ideas to select pseudo-inputs, and a subsequent review came in Quiñonero-Candela et al. (2007). Titsias (2009) proposed to use variational inference for sparse GPs, and Hensman et al. (2013) extended the idea for larger data sets, whereas Banerjee et al. (2013) used linear projections onto low-dimensional subspaces. Besides, some state-of-the-art approximations have been theoretically studied in Bauer et al. (2016). Another approach to deal with numerical issues has recently been proposed in Wilson et al. (2020) to sample from GP efficiently in MCMC algorithms. Bijl et al. (2015) proposed an online version of some of the sparse approximations mentioned above, while Clingerman and Eaton (2017) and Moreno-Muñoz et al. (2019) developed continual learning methods for multi-task GP.

The multi-task framework consists in using data from several tasks (or batches of individuals) to improve the learning or predictive capacities compared to an isolated model. It has been introduced by Caruana (1997) and then adapted in many fields of machine learning. GP versions of such models where introduced by Schwaighofer et al. (2004), and they proposed an EM algorithm for learning. Similar techniques can be found in Shi et al. (2005). Meanwhile, Yu et al. (2005) offered an extensive study of the relationships between the linear model and GPs to develop a multi-task GP formulation. However, since the introduction in Bonilla et al. (2008) of the idea of two matrices modelling covariance between inputs and tasks respectively, the term *multi-task Gaussian process* has mostly referred to the choice made regarding the covariance structure. Some further developments were discussed by Hayashi et al. (2012), Rakitsch et al. (2013) and Zhu and Sun (2014). Let us also mention the work of Swersky et al. (2013) on Bayesian hyper-parameter optimisation in such models. Real applications were tackled by similar models in Williams et al. (2009) and Alaa and van der Schaar (2017).

As we focus on multi-task time series forecasting, there is an immediate connection to the study of multiple curves, or functional data analysis (FDA). As iniatially proposed in Rice and Silverman (1991), it is possible to model and learn mean and covariance structures simultaneously in this context. We also refer to the monographs (Ramsay and Silverman, 2005; Ferraty and Vieu, 2006). In particular, these books introduced several usual ways to model a set of functional objects in frequentist frameworks, for example by using a decomposition in a basis of functions (such as B-splines, wavelets, Fourier). Subsequently, some Bayesian alternatives were developed in Thompson and Rosen (2008), and Crainiceanu and Goldsmith (2010).

Our contributions

Our aim is to define a multi-task GP framework with common mean process, allowing reliable probabilistic forecasts even in multiple-step-ahead problems, or for sparsely observed individuals. For this purpose, (i) We introduce a GP model where the specific covariance structure of each individual is defined through a kernel and its associated set of hyper-parameters, whereas a mean function μ_0 overcomes the weaknesses of classic GPs in making predictions far from observed data. To account for its uncertainty, we propose to define the common mean process μ_0 as a GP as well. (ii) We derive an algorithm called Magma (available as a R package at `https://github.com/ArthurLeroy/MAGMA`) to compute μ_0's hyper-posterior distribution together with the estimation of hyper-parameters in an EM fashion, and discuss its computational complexity. (iii) We enrich Magma with explicit formulas to make predictions for a new, partially observed, individual. The hyper-posterior distribution of μ_0 provides a prior belief on what we expect to observe before seeing any of the new individual's data, as an already-informed process integrating both trend and uncertainty coming from other individuals. (iv) We illustrate the performance of our method on synthetic and two real-life datasets, and obtain state-of-the-art results compared to alternative approaches.

Outline

The remainder of this chapter is organised as follows. We introduce our multi-task Gaussian process model in Section 3.2, along with notation. Section 3.3 is devoted to the inference procedure, with an Expectation-Maximisation (EM) algorithm to estimate the Gaussian process hyper-parameters. We leverage this strategy in Section 3.4 and derive a prediction algorithm. In Section 3.5, we analyse and discuss the computational complexity of both the inference and prediction procedures. Our methodology is illustrated in Section 3.6, with a series of experiments on both synthetic and real-life datasets, and a comparison to competing state-of-the-art algorithms. On those tasks, we provide empirical evidence that our algorithm outperforms other approaches. Section 3.7 draws perspectives for future work, and we defer all proofs to original results to Section 3.8.

3.2 Modelling

3.2.1 Notation

While GPs can handle many types of data, their continuous nature makes them particularly well suited to study temporal phenomena. Throughout, the term *individual* is used as a synonym of task or batch, and adopt notation and vocabulary of time series to remain consistent with the real datasets application we provide in Section 3.6.5, which addresses young

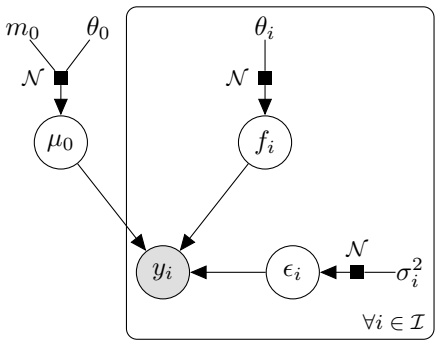

Figure 3.1 – Graphical model of dependencies between variables in the Multi-task Gaussian Process model.

swimmers performances' forecast. These time series are considered as pointwise observations of functions we try to reconstruct thanks to the following generative model.

We are provided with functional data coming from $M \in \mathcal{I}$ different individuals, where $\mathcal{I} \subset \mathbb{N}$. For each individual i, we observe a set of inputs and outputs $\{\left(t_i^1, y_i(t_i^1)\right), \ldots, \left(t_i^{N_i}, y_i(t_i^{N_i})\right)\}$, where N_i is the number of data points for the i-th individual. Since many objects are defined for all individuals, we shorten our notation as follows: for any object x existing for all i, we denote $\{x_i\}_i = \{x_1, \ldots, x_M\}$. Moreover, as we work in a temporal context, the inputs $\left\{t_i^k\right\}_{i,k}$ are referred to as *timestamps*. In the specific case where all individuals are observed at the same timestamps, we call *common* the grid of observations. On the contrary, a grid of observations is *uncommon* if the timestamps are different in number and/or location among the individuals. Some convenient notation:

- $\mathbf{t}_i = \{t_i^1, \ldots, t_i^{N_i}\}$, the set of timestamps for the i-th individual,
- $\mathbf{y}_i = y_i(\mathbf{t}_i)$, the vector of outputs for the i-th individual,
- $\mathbf{t} = \bigcup_{i=1}^{M} \mathbf{t}_i$, the pooled set of timestamps among individuals,
- $N = \#(\mathbf{t})$, the total number of observed timestamps.

3.2.2 Model and hypotheses

Suppose that a functional data is coming from the sum of a mean process, common to all individuals, and an individual-specific centred process. To clarify relationships in the generative model, we illustrate our graphical model in Figure 3.1.

Let $\mathcal{T}$ be the input space, our model is

$$y_i(t) = \mu_0(t) + f_i(t) + \epsilon_i(t), \quad t \in \mathcal{T}, \quad i = 1, \ldots, M,$$

where $\mu_0(\cdot) \sim \mathcal{GP}(m_0(\cdot), k_{\theta_0}(\cdot,\cdot))$ is the mean common process and $f_i(\cdot) \sim \mathcal{GP}(0, c_{\theta_i}(\cdot,\cdot))$ the individual specific process. Moreover, the error term is supposed to be $\epsilon_i(\cdot) \sim \mathcal{GP}(0, \sigma_i^2 I)$. The following notation is used for parameters:

- $k_{\theta_0}(\cdot,\cdot)$, a covariance kernel of hyper-parameters θ_0,

- $\forall i,\ c_{\theta_i}(\cdot,\cdot)$, a covariance kernel of hyper-parameters θ_i,
- $\sigma_i^2 \in \mathbb{R}^+$, the noise term for individual i,
- $\forall i,\ \psi_{\theta_i,\sigma_i^2}(\cdot,\cdot) = c_{\theta_i}(\cdot,\cdot) + \sigma_i^2 I$,
- $\Theta = \{\theta_0, \{\theta_i\}_i, \{\sigma_i^2\}_i\}$, the set of all hyper-parameters of the model.

We also assume that

- $\{f_i\}_i$ are independent,
- $\{\epsilon_i\}_i$ are independent,
- $\forall i,\ \mu_0,\ f_i$ and ϵ_i are independent.

It follows that $\{y_i \mid \mu_0\}_{i=1,\dots,M}$ are independent from one another, and for all $i \in \mathcal{I}$:

$$y_i(\cdot) \mid \mu_0(\cdot) \sim \mathcal{GP}(\mu_0(\cdot), \psi_{\theta_i,\sigma_i^2}(\cdot,\cdot)).$$

Although this model is based on infinite-dimensional GPs, the inference will be conducted on a finite grid of observations. According to the aforementioned notation, we observe $\{(\mathbf{t}_i, \mathbf{y}_i)\}_i$, and the corresponding likelihoods are Gaussian:

$$\mathbf{y}_i \mid \mu_0(\mathbf{t}_i) \sim \mathcal{N}(\mathbf{y}_i; \mu_0(\mathbf{t}_i), \mathbf{\Psi}^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}),$$

where $\mathbf{\Psi}^{\mathbf{t}_i}_{\theta_i,\sigma_i^2} = \psi_{\theta_i,\sigma_i^2}(\mathbf{t}_i,\mathbf{t}_i) = \left[\psi_{\theta_i,\sigma_i^2}(k,l)\right]_{k,\ell\in\mathbf{t}_i}$ is a $N_i \times N_i$ covariance matrix. Since $\mathbf{t}_i$ might be different among individuals, we also need to evaluate μ_0 on the pooled grid $\mathbf{t}$:

$$\mu_0(\mathbf{t}) \sim \mathcal{N}\left(\mu_0(\mathbf{t}); m_0(\mathbf{t}), \mathbf{K}^{\mathbf{t}}_{\theta_0}\right),$$

where $\mathbf{K}^{\mathbf{t}}_{\theta_0} = k_{\theta_0}(\mathbf{t},\mathbf{t}) = [k_{\theta_0}(k,\ell)]_{k,l\in\mathbf{t}}$ is a $N \times N$ covariance matrix.

An alternate hypothesis consists in considering hyper-parameters $\{\theta_i\}_i$ and $\{\sigma_i^2\}_i$ equal for all individuals. We call this hypothesis *Common HP* in the Section 3.6. This particular case models a context where individuals represent different trajectories of the same process, whereas different hyper-parameters indicate different covariance structures and thus a more flexible model. For the sake of generality, the remainder of the chapter is written with θ_i and σ_i^2 notation, when there are no differences in the procedure. Moreover, the model above and the subsequent algorithm may use any covariance function parametrised by a finite set (usually small) of hyper-parameters. For example, a common kernel in the GP literature is known as the *Exponentiated Quadratic* kernel (also called sometimes Squared Exponential or Radial Basis Function kernel). It depends only on two hyper-parameters $\theta = \{v, \ell\}$ and is defined as:

$$k_{\text{EQ}}(x,x') = v^2 \exp\left(-\frac{(x-x')^2}{2\ell^2}\right). \tag{3.1}$$

The *Exponentiated Quadratic* kernel is simple and enjoys useful smoothness properties. This is the kernel used in our implementation (see Section 3.6 for details). Note that there is a rich literature on kernel choice, their construction and properties, which is beyond the scope of the present work: we refer to Rasmussen and Williams (2006) or Duvenaud (2014) for comprehensive studies.

3.3 Inference

3.3.1 Learning

Several approaches to learn hyper-parameters for Gaussian processes have been proposed in the literature, we refer to Rasmussen and Williams (2006) for a comprehensive study. One classical approach, called *empirical Bayes* (Casella, 1985), is based on the maximisation of an explicit likelihood to estimate hyper-parameters. This procedure avoids to sample from intractable distributions, usually resulting in additional computational cost and complicating practical use in moderate to large sample sizes. However, since the likelihood of the model depends on μ_0, we cannot maximise it directly. Therefore, we propose an EM algorithm (see the pseudocode in Algorithm 2) to learn the hyper-parameters Θ. The procedure alternatively computes the hyper-posterior distribution $p(\mu_0 \mid (\mathbf{y}_i)_i, \hat{\Theta})$ with current hyper-parameters, and then optimises Θ according to this hyper-posterior distribution. This EM algorithm converges to local maxima (Dempster et al., 1977), typically in a handful of iterations.

E STEP

For the sake of simplicity, we assume in that section that for all i, j, $\mathbf{t}_i = \mathbf{t}_j = \mathbf{t}$, i.e. the individuals are observed on a common grid of timestamps. The E-step then consists in computing the hyper-posterior distribution of $\mu_0(\mathbf{t})$.

Proposition 3.1. *Assume the hyper-parameters $\hat{\Theta}$ known from initialisation or estimated from a previous M step. The hyper-posterior distribution of μ_0 remains Gaussian:*

$$p\left(\mu_0(\mathbf{t}) \mid \{\mathbf{y}_i\}_i, \hat{\Theta}\right) = \mathcal{N}\left(\mu_0(\mathbf{t}); \hat{m}_0(\mathbf{t}), \hat{\mathbf{K}}^{\mathbf{t}}\right), \tag{3.2}$$

with

- $\hat{\mathbf{K}}^{\mathbf{t}} = \left(\mathbf{K}^{\mathbf{t}}_{\hat{\theta}_0}{}^{-1} + \sum_{i=1}^{M} \mathbf{\Psi}^{\mathbf{t}}_{\hat{\theta}_i, \hat{\sigma}_i^2}{}^{-1}\right)^{-1}$,
- $\hat{m}_0(\mathbf{t}) = \hat{\mathbf{K}}^{\mathbf{t}}\left(\mathbf{K}^{\mathbf{t}}_{\hat{\theta}_0}{}^{-1} m_0(\mathbf{t}) + \sum_{i=1}^{M} \mathbf{\Psi}^{\mathbf{t}}_{\hat{\theta}_i, \hat{\sigma}_i^2}{}^{-1} \mathbf{y}_i\right)$.

Proof. We omit specifying timestamps in what follows since each process is evaluated on $\mathbf{t}$.

$$\begin{aligned}
p\left(\mu_0 \mid \{\mathbf{y}_i\}_i, \hat{\Theta}\right) &\propto p\left(\{\mathbf{y}_i\}_i \mid \mu_0, \hat{\Theta}\right) p\left(\mu_0 \mid \hat{\Theta}\right) \\
&\propto \left\{\prod_{i=1}^{M} p\left(\mathbf{y}_i \mid \mu_0, \hat{\theta}_i, \hat{\sigma}_i^2\right)\right\} p\left(\mu_0 \mid \hat{\theta}_0\right) \\
&\propto \left\{\prod_{i=1}^{M} \mathcal{N}\left(\mathbf{y}_i; \mu_0, \mathbf{\Psi}_{\hat{\theta}_i, \hat{\sigma}_i^2})\right)\right\} \mathcal{N}\left(\mu_0; m_0, \mathbf{K}_{\hat{\theta}_0}\right).
\end{aligned}$$

The term $\mathcal{L}_1 = -(1/2)\log p(\mu_0 \mid \{\mathbf{y}_i\}_i, \hat{\Theta})$ may then be written as

$$\begin{aligned}
\mathcal{L}_1 &= -\frac{1}{2}\log p(\mu_0 \mid \{\mathbf{y}_i\}_i, \hat{\Theta}) \\
&= \sum_{i=1}^{M} (y_i - \mu_0)^\intercal \, \mathbf{\Psi}^{-1}_{\hat{\theta}_i, \hat{\sigma}_i^2} (y_i - \mu_0) + (\mu_0 - m_0)^\intercal \, \mathbf{K}^{-1}_{\hat{\theta}_0} (\mu_0 - m_0) + C_1 \\
&= \sum_{i=1}^{M} \mu_0^\intercal \mathbf{\Psi}^{-1}_{\hat{\theta}_i, \hat{\sigma}_i^2} \mu_0 - 2\mu_0^\intercal \mathbf{\Psi}^{-1}_{\hat{\theta}_i, \hat{\sigma}_i^2} \mathbf{y}_i + \mu_0^\intercal \mathbf{K}^{-1}_{\hat{\theta}_0} \mu_0 - 2\mu_0^\intercal \mathbf{K}^{-1}_{\hat{\theta}_0} m_0 + C_2 \\
&= \mu_0^\intercal \left(\mathbf{K}^{-1}_{\hat{\theta}_0} + \sum_{i=1}^{M} \mathbf{\Psi}^{-1}_{\hat{\theta}_i, \hat{\sigma}_i^2} \right) \mu_0 - 2\mu_0^\intercal \left(\mathbf{K}^{-1}_{\hat{\theta}_0} m_0 + \sum_{i=1}^{M} \mathbf{\Psi}^{-1}_{\hat{\theta}_i, \hat{\sigma}_i^2} \mathbf{y}_i \right) + C_2.
\end{aligned}$$

Identifying terms in the quadratic form with the Gaussian likelihood, we get the desired result.

□

Let us stress here that the above result assumes common timestamps among individuals, which is a simplified setting. We provide a generalisation of this proposition in Section 3.4: Proposition 3.4 holds with uncommon grids of timestamps $\mathbf{t}_i$.

The maximisation step depends on the assumptions on the generative model, resulting in two versions for the EM algorithm (the E step is common to both, the branching point is here).

M STEP: DIFFERENT HYPER-PARAMETERS

Assuming each individual has its own set of hyper-parameters $\{\theta_i, \sigma_i^2\}$, the M step is given by the following.

Proposition 3.2. *Assume* $p(\mu_0 \mid \{\mathbf{y}_i\}_i) = \mathcal{N}\left(\mu_0(\mathbf{t}); \hat{m}_0(\mathbf{t}), \hat{\mathbf{K}}^{\mathbf{t}}\right)$ *given by a previous E step. For a set of hyper-parameters* $\Theta = \{\theta_0, \{\theta_i\}_i, \{\sigma_i^2\}_i\}$*, optimal values are given by*

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmax}} \, \mathbb{E}_{\mu_0 \mid \{\mathbf{y}_i\}_i} \left[p(\{\mathbf{y}_i\}_i, \mu_0(\mathbf{t}) \mid \Theta) \right],$$

inducing $M + 1$ *independent maximisation problems:*

$$\begin{aligned}
\hat{\theta}_0 &= \underset{\theta_0}{\operatorname{argmax}} \; \mathcal{L}^{\mathbf{t}} \left(\hat{m}_0(\mathbf{t}); m_0(\mathbf{t}), \mathbf{K}^{\mathbf{t}}_{\theta_0} \right), \\
(\hat{\theta}_i, \hat{\sigma}_i^2) &= \underset{\theta_i, \sigma_i^2}{\operatorname{argmax}} \; \mathcal{L}^{\mathbf{t}_i}(\mathbf{y}_i; \hat{m}_0(\mathbf{t}), \mathbf{\Psi}^{\mathbf{t}_i}_{\theta_i, \sigma_i^2}), \; \forall i,
\end{aligned}$$

where

$$\mathcal{L}^{\mathbf{t}}(\mathbf{x}; \mathbf{m}, \mathbf{S}) = \log \mathcal{N}(\mathbf{x}; \mathbf{m}, \mathbf{S}) - \frac{1}{2} Tr\left(\hat{\mathbf{K}}^{\mathbf{t}} \mathbf{S}^{-1} \right).$$

Proof. One simply has to distribute the conditional expectation in order to get the right likelihood to maximise, and then notice that the function can be written as a sum of M+1 independent (with respect to the hyper-parameters) terms. Moreover, by rearranging, one can observe that each independent term is the sum of a Gaussian likelihood and a correction trace term. See Section 3.8.2 for details. □

M STEP: COMMON HYPER-PARAMETERS

Alternatively, assuming all individuals share the same set of hyper-parameters $\{\theta, \sigma^2\}$, the M step is given by the following.

Proposition 3.3. *Assume* $p(\mu_0 \mid \{\mathbf{y}_i\}_i) = \mathcal{N}\left(\mu_0(\mathbf{t}); \hat{m}_0(\mathbf{t}), \hat{\mathbf{K}}^{\mathbf{t}}\right)$ *given by a previous E step. For a set of hyper-parameters* $\Theta = \{\theta_0, \theta, \sigma^2\}$*, optimal values are given by*

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmax}}\, \mathbb{E}_{\mu_0 \mid \{\mathbf{y}_i\}_i} \left[p(\{\mathbf{y}_i\}_i, \mu_0(\mathbf{t}) \mid \Theta) \right],$$

inducing two independent maximisation problems:

$$\hat{\theta}_0 = \underset{\theta_0}{\operatorname{argmax}}\ \mathcal{L}^{\mathbf{t}}\left(\hat{m}_0(\mathbf{t}); m_0(\mathbf{t}), \mathbf{K}^{\mathbf{t}}_{\theta_0}\right),$$
$$(\hat{\theta}, \hat{\sigma}^2) = \underset{\theta, \sigma^2}{\operatorname{argmax}}\ \mathcal{L}_M(\theta, \sigma^2),$$

where

$$\mathcal{L}_M(\theta, \sigma^2) = \sum_{i=1}^{M} \mathcal{L}^{\mathbf{t}_i}(\mathbf{y}_i; \hat{m}_0(\mathbf{t}), \mathbf{\Psi}^{\mathbf{t}_i}_{\theta, \sigma^2}).$$

Proof. We use the same strategy as for Proposition 3.2, see Section 3.8.2 for details. □

In both cases, explicit gradients associated with the likelihoods to maximise are available, facilitating the optimisation with gradient-based methods.

3.3.2 Initialisation

To implement the EM algorithm described above, several constants must be (appropriately) initialised:

- $m_0(\cdot)$, the mean parameter from the hyper-prior distribution of the mean process $\mu_0(\cdot)$. A somewhat classical choice in GP is to set its value to a constant, typically 0 in the absence of external knowledge. Notice that, in our multi-task framework, the influence of $m_0(\cdot)$ in hyper-posterior computation decreases quickly as M grows.
- Initial values for kernel parameters θ_0 and $\{\theta_i\}_i$. Those strongly depend on the chosen kernel and its properties. We advise initiating θ_0 and $\{\theta_i\}_i$ with close values, as a too large difference might induce a nearly singular covariance matrix and result in numerical instability. In such pathological regime, the influence of a specific individual tends to overtake others in the calculus of μ_0's hyper-posterior distribution.
- Initial values for the variance of the error terms $\left\{\sigma_i^2\right\}_i$. This choice mostly depends on the context and properties of the dataset. We suggest avoiding initial values with more than an order of magnitude different from the variability of data. In particular, a too high value might result in a model mostly capturing noise.

As a final note, let us stress that the EM algorithm depends on the initialisation and is only guaranteed to converge to local maxima of the likelihood function (McLachlan and Krishnan, 2007). Several strategies have been considered in the literature to tackle this issue such as simulated annealing and the use of multiple initialisations (Biernacki et al., 2003). In this work, we choose the latter option.

3.3.3 Pseudocode

We wrap up this section with the pseudocode of the EM component of our complete algorithm, which we call MAGMA (standing for Multi tAsk Gaussian processes with common MeAn). The corresponding code is available at https://github.com/ArthurLeroy/MAGMA.

Algorithm 2 MAGMA: EM component

Initialise m_0 and $\Theta = \left\{\theta_0, \{\theta_i\}_i, \left\{\sigma_i^2\right\}_i\right\}$.
while not converged **do**
  E step: Compute the hyper-posterior distribution
    $p(\mu_0 \mid \{\mathbf{y}_i\}_i, \hat{\Theta}) = \mathcal{N}(\hat{m}_0, \hat{\mathbf{K}})$.

  M step: Estimate hyper-parameter by maximising
    $\hat{\Theta} = \underset{\Theta}{\operatorname{argmax}}\, \mathbb{E}_{\mu_0|\{\mathbf{y}_i\}_i}\left[p(\mu_0, \{\mathbf{y}_i\}_i \mid \Theta)\right]$.
end while
return $\hat{\Theta}$, $\hat{m}_0$, $\hat{\mathbf{K}}$.

3.3.4 Discussion of EM algorithms and alternatives

Let us stress that even though we focus on prediction purpose in this chapter, the output of the EM algorithm already provides results on related FDA problems. The generative model in Yang et al. (2016) describes a Bayesian framework that resembles ours to smooth multiple curves simultaneously. However, modelling variance structure with an Inverse-Wishard process forces the use of an MCMC algorithm for inference or the introduction of a more tractable approximation in Yang et al. (2017). One can think of the learning through MAGMA and applying a single task GP regression on each individual as an *empirical Bayes* counterpart to their approach. Meanwhile, μ_0's hyper-posterior distribution also provides the probabilistic estimation of a mean curve from a set of functional data. The closest method to our approach can be found in Shi et al. (2007) and the following book Shi and Choi (2011), though by several aspects, authors dealt with more general features like multidimensional or non-functional inputs. The authors also work in the context of a multi-task GP model, and one can retrieve the idea of defining a mean function μ_0 to overcome the weaknesses of classic GPs in making predictions far from observed data. Since their model uses B-splines to estimate this mean function, thanks to information from multiple individuals, this method only works if all individuals share the same grid of observation, and does not account for uncertainty over μ_0.

3.4 Prediction

Once the hyper-parameters of the model have been learned, we can focus on our main goal: prediction at new timestamps. Since $\hat{\Theta}$ is known and for the sake of concision, we omit conditioning on $\hat{\Theta}$ in the sequel. Note there are two cases for prediction (referred to as *Type I* and *Type II* in Shi and Cheng, 2014, Section 3.2.1), depending on whether we observe some data or not for any new individual we wish to predict on. We denote by the index $*$ a new individual for whom we want to make a prediction at timestamps $\mathbf{t}^p$. If there are no available data for this individual, we have no $*$-specific information, and the prediction is merely given by $p(\mu_0(\mathbf{t}^p) \mid \{\mathbf{y}_i\}_i)$. This quantity may be considered as the 'generic' (or

Type II) prediction according to the trained model, and only informs us through the mean process. Computing $p(\mu_0(\mathbf{t}^p) \mid \{\mathbf{y}_i\}_i)$ is also one of the steps leading to the prediction for a partially observed new individual (*Type I*). The latter being the most compelling case, we consider *Type II* prediction as a particular case of the full *Type I* procedure, described below.

If we observe $\{\mathbf{t}_*, y_*(\mathbf{t}_*)\}$ for the new individual, the multi-task GP prediction is obtained by computing the posterior distribution $p(y_*(\mathbf{t}^p) \mid y_*(\mathbf{t}_*), \{\mathbf{y}_i\}_i)$. Note that the conditioning is taken over $y_*(\mathbf{t}_*)$, as for any GP regression, but also on $\{\mathbf{y}_i\}_i$, which is specific to our multi-task setting. Computing this distribution requires the following steps.

1. Choose a grid of prediction $\mathbf{t}^p$ and define the pooled vector of timestamps $\mathbf{t}_*^p$,
2. Compute the hyper-posterior distribution of μ_0 at $\mathbf{t}_*^p$: $p(\mu_0(\mathbf{t}_*^p) \mid \{\mathbf{y}_i\}_i)$,
3. Compute the multi-task prior distribution $p(y_*(\mathbf{t}_*^p) \mid \{\mathbf{y}_i\}_i)$,
4. Compute hyper-parameters θ_* of the new individual's covariance matrix (optional),
5. Compute the multi-task posterior distribution: $p(y_*(\mathbf{t}^p) \mid y_*(\mathbf{t}_*), \{\mathbf{y}_i\}_i)$.

3.4.1 Posterior inference on the mean process

As mentioned above, we observed a new individual at timestamps $\mathbf{t}_*$. The GP regression consists of arbitrarily choosing a vector $\mathbf{t}^p$ of timestamps on which we wish to make a prediction. Since a GP is an infinite-dimensional object, we can pick a finite-dimensional vector at any new location. Then, we define new notation for the pooled vector of timestamps $\mathbf{t}_*^p = \begin{bmatrix} \mathbf{t}^p \\ \mathbf{t}_* \end{bmatrix}$, which will serve as a working grid to define the prior and posterior distributions involved in the prediction process. One can note that, although not mandatory in theory, it is often a good idea to include the observed timestamps of training individuals, $\mathbf{t}$, within $\mathbf{t}_*^p$ since they match locations which contain information for the mean process to 'help' the prediction. In particular, if $\mathbf{t}_*^p = \mathbf{t}$, the computation of μ_0's hyper-posterior distribution is not necessary since $p(\mu_0(\mathbf{t}) \mid \{\mathbf{y}_i\}_i)$ has previously been obtained with the EM algorithm. However, in general, it is necessary to compute the hyper-posterior $p(\mu_0(\mathbf{t}_*^p) \mid \{\mathbf{y}_i\}_i)$ at the new timestamps. The idea remains similar to the E step aforementioned, and we obtain the following result.

Proposition 3.4. *Let $\mathbf{t}_*^p$ be a vector of timestamps of size $\tilde{N}$. The hyper-posterior distribution of μ_0 remains Gaussian:*

$$p\left(\mu_0(\mathbf{t}_*^p) \mid \{\mathbf{y}_i\}_i\right) = \mathcal{N}\left(\mu_0(\mathbf{t}_*^p); \hat{m}_0(\mathbf{t}_*^p), \hat{\mathbf{K}}_*^p\right),$$

with:

- $\hat{\mathbf{K}}_*^p = \left(\tilde{\mathbf{K}}^{-1} + \sum_{i=1}^{M} \tilde{\mathbf{\Psi}}_i^{-1}\right)^{-1}$,
- $\hat{m}_0(\mathbf{t}_*^p) = \hat{\mathbf{K}}_*^p \left(\tilde{\mathbf{K}}^{-1} m_0\left(\mathbf{t}_*^p\right) + \sum_{i=1}^{M} \tilde{\mathbf{\Psi}}_i^{-1} \tilde{\mathbf{y}}_i\right)$,

where we used the shortening notation:

- $\tilde{\mathbf{K}} = k_{\hat{\theta}_0}\left(\mathbf{t}_*^p, \mathbf{t}_*^p\right)$ *($\tilde{N} \times \tilde{N}$ matrix)*,

- $\tilde{\mathbf{y}}_i = \left(\mathbb{1}_{[t \in \mathbf{t}_i]} \times y_i(t)\right)_{t \in \mathbf{t}_*^p}$ *($\tilde{N}$-size vector),*
- $\tilde{\mathbf{\Psi}}_i = \left[\mathbb{1}_{[t,t' \in \mathbf{t}_i]} \times \psi_{\hat{\theta}_i, \hat{\sigma}_i^2}(t,t')\right]_{t,t' \in \mathbf{t}_*^p}$ *($\tilde{N} \times \tilde{N}$ matrix).*

Proof. The sketch of the proof is similar to Proposition 3.1 in the E step. The only technicality consists in dealing carefully with the dimensions of vectors and matrices involved, and whenever relevant, to define augmented versions of $\mathbf{y}_i$ and $\mathbf{\Psi}_{\hat{\theta}_i, \hat{\sigma}_i^2}$ with 0 elements at unobserved timestamps' position for the i-th individual. Note that if we pick a vector $\mathbf{t}_*^p$ including only some of the timestamps from $\mathbf{t}_i$, information coming from y_i at the remaining timestamps is ignored. We defer details to Section 3.8.1. □

3.4.2 Computing the multi-task prior distribution

According to our generative model, given the mean process, any new individual $*$ is modelled as:

$$y_*(\cdot) \mid \mu_0(\cdot) \sim \mathcal{GP}\left(\mu_0(\cdot), \mathbf{\Psi}_{\theta_*, \sigma_*^2}(\cdot,\cdot)\right).$$

Therefore, for any finite-dimensional vector of timestamps, and in particular for $\mathbf{t}_*^p$, $p(y_*(\mathbf{t}_*^p) \mid \mu_0(\mathbf{t}_*^p))$ is a multivariate Gaussian vector. Moreover, from this distribution and μ_0's hyper-posterior, we can figure out the multi-task prior distribution over $y_*(\mathbf{t}_*^p)$.

Proposition 3.5. *For a set of timestamps $\mathbf{t}_*^p$, the multi-task prior distribution of y_* is given by*

$$p(y_*(\mathbf{t}_*^p) \mid \{\mathbf{y}_i\}_i) = \mathcal{N}\left(y_*(\mathbf{t}_*^p); \hat{m}_0(\mathbf{t}_*^p), \hat{\mathbf{K}}_*^p + \mathbf{\Psi}_{\theta_*, \sigma_*^2}^{\mathbf{t}_*^p}\right). \tag{3.3}$$

Proof. To compute this prior, we need to integrate $p(y_* \mid \mu_0, \{\mathbf{y}_i\}_i)$ over the mean process μ_0, whereas the multi-task aspect remains through the conditioning over $\{\mathbf{y}_i\}_i$. We omit the writing of timestamps, by using the simplified notation μ_0 and y_* instead of $\mu_0(\mathbf{t}_*^p)$ and $y_*(\mathbf{t}_*^p)$, respectively. We first use the assumption that $\{y_i \mid \mu_0\}_{i \in \{1,\dots,M\}} \perp\!\!\!\perp y_* \mid \mu_0$, *i.e.*, the individuals are independent conditionally to μ_0. Then, one can notice that the two distributions involved within the integral are Gaussian, which leads to the explicit Gaussian target distribution after integration.

$$\begin{aligned}
p(y_* \mid \{\mathbf{y}_i\}_i) &= \int p\left(y_*, \mu_0 \mid \{\mathbf{y}_i\}_i\right) \mathrm{d}\mu_0 \\
&= \int p\left(y_* \mid \mu_0, \{\mathbf{y}_i\}_i\right) p(\mu_0 \mid \{\mathbf{y}_i\}_i) \, \mathrm{d}\mu_0 \\
&= \int \underbrace{p\left(y_* \mid \mu_0\right))}_{\mathcal{N}\left(y_*; \mu_0, \mathbf{\Psi}_{\theta_*, \sigma_*^2}^{\mathbf{t}_*^p}\right)} \underbrace{p(\mu_0 \mid \{\mathbf{y}_i\}_i)}_{\mathcal{N}\left(\mu_0; \hat{m}_0, \hat{\mathbf{K}}_*^p\right)} \mathrm{d}\mu_0.
\end{aligned}$$

This convolution of two Gaussians remains Gaussian (Bishop, 2006, Chapter 2.3.3). For any random variable $X \in \Omega$, and A_X depending on X, let $\mathbb{E}_{A_X}[X] = \int_\Omega x \, p(A_X) \, \mathrm{d}x$. The mean parameter is then given by

$$
\begin{aligned}
\mathbb{E}_{y_* | \{\mathbf{y}_i\}_i} [\, y_* \,] &= \int y_* \; p\left(y_* \mid \{\mathbf{y}_i\}_i\right) \mathrm{d}y_* \\
&= \int y_* \int p\left(y_* \mid \mu_0\right) p(\mu_0 \mid \{\mathbf{y}_i\}_i) \, \mathrm{d}\mu_0 \, \mathrm{d}y_* \\
&= \int \left(\int y_* p\left(y_* \mid \mu_0\right) \mathrm{d}y_* \right) p(\mu_0 \mid \{\mathbf{y}_i\}_i) \, \mathrm{d}\mu_0 \\
&= \int \mathbb{E}_{y_* | \mu_0} [\, y_* \,] \, p(\mu_0 \mid \{\mathbf{y}_i\}_i) \, \mathrm{d}\mu_0 \\
&= \mathbb{E}_{\mu_0 | \{\mathbf{y}_i\}_i} \left[\mathbb{E}_{y_* | \mu_0} [\, y_* \,] \right] \\
&= \mathbb{E}_{\mu_0 | \{\mathbf{y}_i\}_i} [\, \mu_0 \,] \\
&= \hat{m}_0 .
\end{aligned}
$$

Following the same idea, the second-order moment is given by

$$
\begin{aligned}
\mathbb{E}_{y_* | \{\mathbf{y}_i\}_i} \left[y_*^2 \right] &= \mathbb{E}_{\mu_0 | \{\mathbf{y}_i\}_i} \left[\mathbb{E}_{y_* | \mu_0} \left[y_*^2 \right] \right] \\
&= \mathbb{E}_{\mu_0 | \{\mathbf{y}_i\}_i} \left[\mathbb{V}_{y_* | \mu_0} [\, y_* \,] + \mathbb{E}_{y_* | \mu_0} [\, y_* \,]^2 \right] \\
&= \boldsymbol{\Psi}_{\theta_*, \sigma_*^2} + \mathbb{E}_{\mu_0 | \{\mathbf{y}_i\}_i} \left[\mu_0^2 \right] \\
&= \boldsymbol{\Psi}_{\theta_*, \sigma_*^2} + \mathbb{V}_{\mu_0 | \{\mathbf{y}_i\}_i} [\, \mu_0 \,] + \mathbb{E}_{\mu_0 | \{\mathbf{y}_i\}_i} [\, \mu_0 \,]^2 \\
&= \boldsymbol{\Psi}_{\theta_*, \sigma_*^2} + \hat{\mathbf{K}} + \hat{m}_0^2 ,
\end{aligned}
$$

hence

$$
\begin{aligned}
\mathbb{V}_{y_* | \{\mathbf{y}_i\}_i} [\, y_* \,] &= \mathbb{E}_{y_* | \{\mathbf{y}_i\}_i} \left[y_*^2 \right] - \mathbb{E}_{y_* | \{\mathbf{y}_i\}_i} [\, y_* \,]^2 \\
&= \boldsymbol{\Psi}_{\theta_*, \sigma_*^2} + \hat{\mathbf{K}} + \hat{m}_0^2 - \hat{m}_0^2 \\
&= \boldsymbol{\Psi}_{\theta_*, \sigma_*^2} + \hat{\mathbf{K}} .
\end{aligned}
$$

□

Note that the process $y_*(\cdot) \mid \{\mathbf{y}_i\}_i$ is not a GP, although its finite-dimensional evaluation (3.3) remains Gaussian. The covariance structure cannot be expressed as a kernel that could be directly evaluated on any vector: the process is known as a *degenerated GP*. In practice however, this does not bear much consequence as an arbitrary vector of timestamps τ can still be chosen, then we compute the hyper-posterior $p(\mu_0(\tau) \mid \{\mathbf{y}_i\}_i)$, which yields the Gaussian distribution $p(y_*(\tau) \mid \{\mathbf{y}_i\}_i)$ as above. For the sake of simplicity, we now rename the covariance matrix of the prior distribution:

$$
\hat{\mathbf{K}}_*^p + \boldsymbol{\Psi}_{\theta_*, \sigma_*^2}^{\mathbf{t}_*^p} = \boldsymbol{\Gamma}_*^p = \begin{pmatrix} \boldsymbol{\Gamma}_{pp} & \boldsymbol{\Gamma}_{p*} \\ \boldsymbol{\Gamma}_{*p} & \boldsymbol{\Gamma}_{**} \end{pmatrix},
$$

where the indices in the blocks of the matrix correspond to the associated timestamps $\mathbf{t}^p$ and $\mathbf{t}_*$.

3.4.3 Learning the new hyper-parameters

When we collect data points for a new individual, as in the single-task GP setting, we need to learn the hyper-parameters of its covariance function before making predictions. A salient fact in our multi-task approach is that we include this step in the prediction process, for the two following reasons. First, the model is already trained for individuals $i = 1, \ldots, M$, and this training is general and independent from future individual $*$ or the choice of prediction timestamps. Since learning these new hyper-parameters requires knowledge of $\mu(\mathbf{t}_*^p)$ and thus of the prediction timestamps, we cannot compute them beforehand. Secondly, learning these hyper-parameters with the empirical Bayes approach only requires maximisation of a Gaussian likelihood which is negligible in computing time compared to the previous EM algorithm. As for single-task GP, we have the following estimates for hyper-parameters:

$$\begin{aligned}\hat{\Theta}_* &= \underset{\Theta_*}{\operatorname{argmax}}\, p(y_*(\mathbf{t}_*) \mid \{\mathbf{y}_i\}_i, \Theta_*) \\ &= \underset{\Theta_*}{\operatorname{argmax}}\, \mathcal{N}\left(y_*(\mathbf{t}_*); \hat{m}_0(\mathbf{t}_*), \mathbf{\Gamma}_{**}^{\Theta_*}\right).\end{aligned}$$

Note that this step is optional depending on model: in the common hyper-parameters model (i.e. $(\theta, \sigma^2) = (\theta_i, \sigma_i^2)$), any new individual will share the same hyper-parameters and we already have $\hat{\Theta}_* = (\hat{\theta}_*, \hat{\sigma}_*^2) = (\hat{\theta}, \hat{\sigma}^2)$ from the EM algorithm.

3.4.4 Prediction

We can write the prior distribution, separating observed and prediction timestamps, as:

$$\begin{aligned}p(y_*(\mathbf{t}_*^p) \mid \{\mathbf{y}_i\}_i) &= p(y_*(\mathbf{t}^p), y_*(\mathbf{t}_*) \mid \{\mathbf{y}_i\}_i) \\ &= \mathcal{N}\left(y_*(\mathbf{t}_*^p); \hat{m}_0(\mathbf{t}_*^p), \mathbf{\Gamma}_*^p\right) \\ &= \mathcal{N}\left(\begin{bmatrix} y_*(\mathbf{t}^p) \\ y_*(\mathbf{t}_*) \end{bmatrix}; \begin{bmatrix} \hat{m}_0(\mathbf{t}^p) \\ \hat{m}_0(\mathbf{t}_*) \end{bmatrix}, \begin{pmatrix} \mathbf{\Gamma}_{pp} & \mathbf{\Gamma}_{p*} \\ \mathbf{\Gamma}_{*p} & \mathbf{\Gamma}_{**} \end{pmatrix}\right).\end{aligned}$$

The conditional distribution remains Gaussian (Bishop, 2006), and the predictive distribution is given by:

$$p(y_*(\mathbf{t}^p) \mid y_*(\mathbf{t}_*), \{\mathbf{y}_i\}_i) = \mathcal{N}\left(y_*(\mathbf{t}^p); \hat{\mu}_0^p, \hat{\mathbf{\Gamma}}^p\right),$$

where:

- $\hat{\mu}_0^p = \hat{m}_0(\mathbf{t}^p) + \mathbf{\Gamma}_{p*}\mathbf{\Gamma}_{**}^{-1}\left(y_*(\mathbf{t}_*) - \hat{m}_0(\mathbf{t}_*)\right)$,
- $\hat{\mathbf{\Gamma}}^p = \mathbf{\Gamma}_{pp} - \mathbf{\Gamma}_{p*}\mathbf{\Gamma}_{**}^{-1}\mathbf{\Gamma}_{*p}$.

3.5 Complexity analysis for training and prediction

Computational complexity is of paramount importance in GPs as it quickly scales with large datasets. The classical cost to train a GP is $\mathcal{O}(N^3)$, and $\mathcal{O}(N^2)$ for prediction (Rasmussen and Williams, 2006) where N is the number of data points (see aforementioned references in Section 3.1 for sparse approximations). Since MAGMA uses information from M individuals, each of them providing N_i observations, these quantities determine the overall complexities

of the algorithm. If we recall that N is the number of distinct timestamps (*i.e.* $N \leq \sum_{i=1}^{M} N_i$), the training complexity is $\mathcal{O}\left(M \times N_i^3 + N^3\right)$ (*i.e.* the complexity of each EM iteration). As usual with GPs, the cubic costs come from the inversion of the corresponding matrices, and here, the constant is proportional to the number of iterations of the EM algorithm. The dominating term in this expression depends on the values of M, relatively to N. For a large number of individuals with many common timestamps ($MN_i \gtrsim N$), the first term dominates. For diverse timestamps among individuals ($MN_i \lesssim N$), the second term becomes the primary burden, as in any GP problem. During the prediction step, the re-computation of μ_0's hyper-posterior implies the inversion of a $\tilde{N} \times \tilde{N}$ (dimension of $\mathbf{t}_*^p$) which has a $\mathcal{O}(\tilde{N}^3)$ complexity while the final prediction is $\mathcal{O}(N_*^3)$. In practice, the most computing-expensive steps can be performed in advance to allow for quick on-the-fly prediction when collecting new data. If we observe the training dataset once and pre-compute the hyper-posterior of μ_0 on a fine grid on which to predict later, the immediate computational cost for each new individual is identical to the one of the single-task GP regression.

3.6 Experimental results

We evaluate our Magma algorithm on synthetic data, and two real datasets. The classical GP regression on single tasks separately is used as the baseline alternative for predictions. While it is not expected to perform well on the dataset used, the comparison highlights the interest of multi-task approaches. To our knowledge, the only alternative to Magma is the GPFDA algorithm from Shi et al. (2007), Shi and Choi (2011), described in Section 3.3.4, and the associated R package *GPFDA*, which is applied on the examples. Throughout the section, the standard *Exponentiated Quadratic* kernel (see Equation (3.1)) is used both for simulating the data and for the covariance structures in the three algorithms. Hence, each kernel is associated with $\theta = \{v, \ell\}$, $v, \ell \in \mathbb{R}^+$, a set of, respectively, variance and length-scale hyper-parameters. Each simulated dataset has been drawn from the sampling scheme below:

1. Define a random working grid $\mathbf{t} \subset [0, 10]$ of $N = 200$ timestamps, and a number M of individuals.
2. Define the prior mean for μ_0 : $m_0(t) = at + b$, $\forall t \in \mathbf{t}$, where $a \in [-2, 2]$ and $b \in [0, 10]$.
3. Draw uniformly hyper-parameters for μ_0's kernel : $\theta_0 = \{v_0, \ell_0\}$, where $v_0 \in \left[1, e^5\right]$ and $\ell_0 \in \left[1, e^2\right]$.
4. Draw $\mu_0(\mathbf{t}) \sim \mathcal{N}\left(m_0(\mathbf{t}), \mathbf{K}_{\theta_0}^{\mathbf{t}}\right)$.
5. For all $i = 1, \ldots, M$, $\theta_i = \{v_i, \ell_i\}$, where $v_i \in \left[1, e^5\right]$, $\ell_i \in \left[1, e^2\right]$, and $\sigma_i^2 \in [0, 1]$.
6. For all $i = 1, \ldots, M$, draw a subset uniformly at random $\mathbf{t}_i \subset \mathbf{t}$ of $N_i = 30$ timestamps, and draw $\mathbf{y}_i \sim \mathcal{N}\left(\mu_0(\mathbf{t}_i), \mathbf{\Psi}_{\theta_i, \sigma_i^2}^{\mathbf{t}_i}\right)$.

This procedure provides a synthetic data set $\{\mathbf{t}_i, \mathbf{y}_i\}_i$, and its associated mean process $\mu_0(\mathbf{t})$. Those quantities are used to train the model, make predictions with each algorithm, and then compute errors in μ_0 estimation and forecasts. We recall that the Magma algorithm enables two different settings depending on the model's assumption over hyper-parameters (HP), and we refer to them as *Common HP* and *Different HP* in the following.

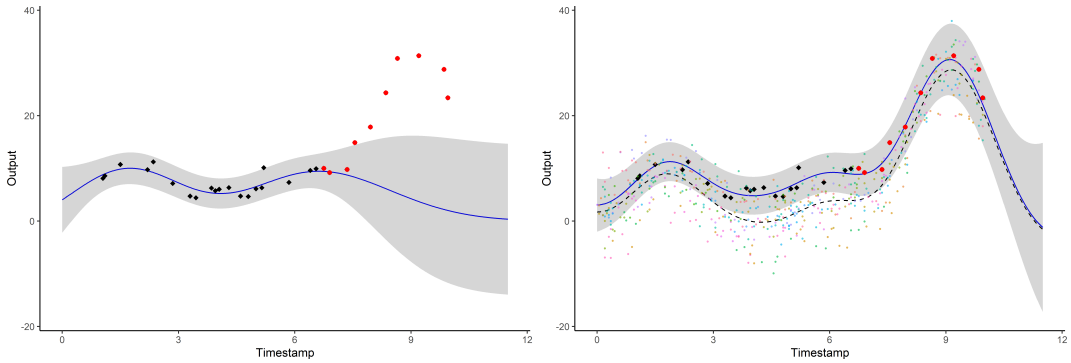

Figure 3.2 – Prediction curves (blue) for a new individual with associated 95% credible intervals (grey) for GP regression (left) and MAGMA (right). The dashed line represents the mean function of the mean process's hyper-posterior $p(\mu_0 \mid \{\mathbf{y}_i\}_i)$. Observed data points are in black, and testing data points are in red. The colourful backward points are the observations from the training dataset, each colour corresponding to a different individual.

In order to test these two contexts, differentiated datasets have been generated, by drawing *Common HP data* or *Different HP data* for each individual at step 5. We previously presented the idea of the model used in GPFDA, and, although the algorithm has many features (in particular about the type and number of input variables), it is not yet usable when timestamps are different among individuals. Therefore, two frameworks are considered, *Common grid* and *Uncommon grid*, to take this specification into account. Thus, the comparison between the different methods can only be performed on data generated under the settings *Common HP* and *Common grid*, and the effect of the different settings on MAGMA is analysed separately. Moreover, without additional knowledge, the initialisation for the prior mean function, $m_0(\cdot)$, is set to be equal to 0 for each algorithm. Except in some experiments, where the influence of the number of individuals is analysed, the generic value is $M = 20$. In the case of prediction on unobserved timestamps for a new individual, the first 20 data points are used as observations, and the remaining 10 are taken as test values.

3.6.1 Illustration on a simple example

To illustrate the multi-task approach of MAGMA, Figure 3.2 displays a comparison between single GP regression and MAGMA on a simple example, from a dataset simulated according to the scheme above. Given the observed data (in black), values on a thin grid of unobserved timestamps are predicted and compared, in particular, with the true test values (in red). As expected, GP regression provides a good fitting close to the data points and then dives rapidly to the prior 0 with increasing uncertainty. Conversely, although the initialisation for the prior mean was also 0 in MAGMA, the hyper-posterior distribution of μ_0 (dashed line) is estimated thanks to all individuals in the training dataset. This process acts as an informed prior helping GP prediction for the new individual, even far from its own observations. More precisely, 3 phases can be distinguished according to the level of information coming from the data: in the first one, close to the observed data ($t \in [1, 7]$), the two processes behave similarly, except a slight increase in the variance for MAGMA, which is logical since the prediction also takes uncertainty over μ_0 into account (see Equation (3.3)); in the second one, on intervals of unobserved timestamps containing data points from the training dataset ($t \in [0, 1] \cup [7, 10]$), the prediction is guided by the information coming from other individuals

	Prediction		Estimation μ_0	
	MSE	CI_{95}	MSE	CI_{95}
MAGMA	**18.7 (31.4)**	**93.8 (13.5)**	**1.3 (2)**	**94.3 (11.3)**
GPFDA	31.8 (49.4)	90.4 (18.1)	2.4 (3.6)	$\star$
GP	87.5 (151.9)	74.0 (32.7)		

Table 3.1 – Average MSE (sd) and average CI_{95} coverage (sd) on 100 runs for GP, GPFDA and MAGMA. ($\star$: 99.6 (2.8), the measure of incertitude from the GPFDA package is not a genuine credible interval)

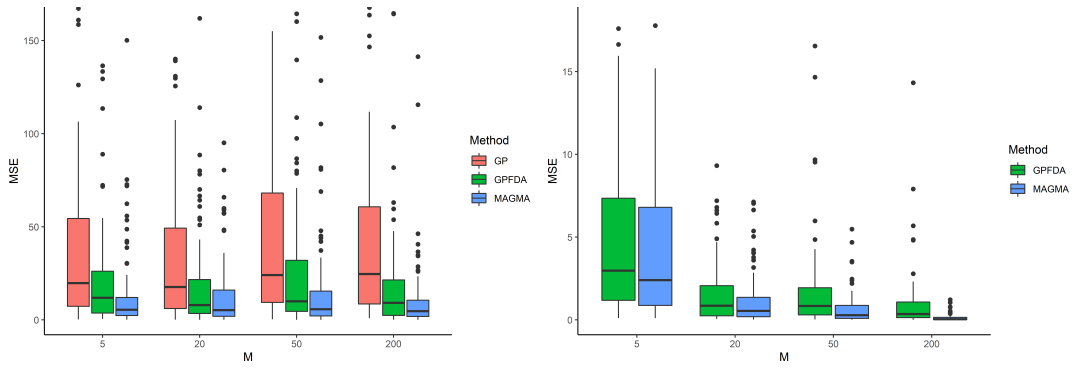

Figure 3.3 – MSE with respect to the number M of training individuals (100 runs in each case). *Left*: prediction error on 10 testing points. *Right*: estimation error of the true mean process μ_0.

through μ_0. In this context, the mean trajectory remains coherent and the uncertainty increases only slightly. In the third case, where no observations are available neither from new individual nor from training dataset ($t \in [\,10, 12\,]$), the prediction behaves as expected, with a slow drifting to the prior mean 0, with highly increasing variance. Overall, the multi-task framework provides reliable probabilistic predictions on a wider range of timestamps, potentially outside of the usual scope for GPs.

3.6.2 Performance comparison on simulated datasets

We confront the performance of MAGMA to alternatives in several situations and for different datasets. In the first place, the classical GP regression (GP), GPFDA and MAGMA are compared through their performance in prediction and estimation of the true mean process μ_0. In the prediction context, the performances are evaluated according to the following indicators:

- the mean squared error (MSE) which compares the predicted values to the true test values of the 10 last timestamps:

$$\frac{1}{10}\sum_{k=21}^{30}\left(y_*^{pred}(t_*^k) - y_*^{true}(t_*^k)\right)^2,$$

- the ratio of CI_{95} coverage, i.e. the percentage of unobserved data points effectively lying within the 95% credible interval defined from the predictive posterior distribution

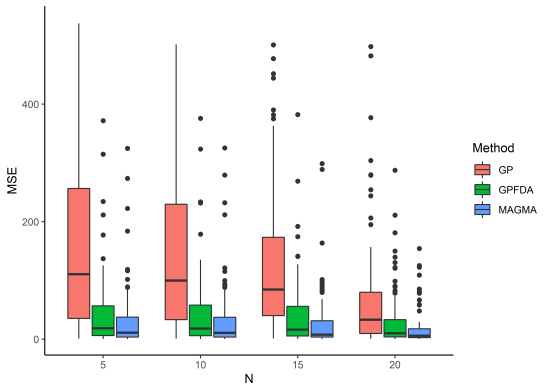

Figure 3.4 – MSE prediction error on the 10 last testing points with respect to the increasing number N of observed timestamps, among the first 20 points (100 runs in each case).

$p(y_*(\mathbf{t}^p) \mid y_*(\mathbf{t}_*), \{\mathbf{y}_i\}_i)$:

$$100 \times \frac{1}{10} \sum_{k=21}^{30} \mathbb{1}_{\{y_*^{true}(t_*^k) \in\, CI_{95}\}}.$$

The ratio of CI_{95} coverage gives an approximation of the predictive variance reliability and should be as close to the value 95% as possible. Other values would indicate a tendency to underestimate or overestimate the uncertainty. Let us recall that GPFDA uses B-splines to estimate the mean process and does not account for uncertainty, contrarily to a probabilistic framework as Magma. However, a measure of uncertainty based on an empirical variance estimated from training curves is proposed (see Shi and Cheng, 2014, Section 3.2.1). In practice, this measure constantly overestimates the true variance, and the CI_{95} coverage is generally equal or close to 100%.

In the estimation context, the performances are evaluated thanks to another MSE, which compares the estimations to the true values of μ_0 at all timestamps:

$$\frac{1}{M} \sum_{i=1}^{M} \frac{1}{N_i} \sum_{k=1}^{N_i} \left(\mu_0^{pred}(t_i^k) - \mu_0^{true}(t_i^k) \right)^2.$$

Table 3.1 presents the results obtained over 100 datasets, where the model is trained on $M = 20$ individuals, each of them observed on $N = 30$ common timestamps. As expected, both multi-task methods lead to better results than GP. However, Magma outperforms GPFDA, both in estimation of μ_0 and in prediction performance. In terms of error as well as in uncertainty quantification, Magma provides more accurate results, in particular with a CI_{95} coverage close to the 95% expected value. Each method presents a quite high standard deviation for MSE in prediction, which is due to some datasets with particularly difficult values to predict, although most of the cases lead to small errors. This behaviour is reasonably expected since the forecast of 10-ahead-timestamps might sometimes be tricky. It can also be noticed on Figure 3.3 that Magma consistently provides lower errors as well as less pathological behaviour, as it may sometimes occur with the B-splines modelling used in GPFDA.

To highlight the effect of the number of individuals M on the performance, Figure 3.3 provides the same 100 runs trial as previously, for different values of M. The boxplots exhibit, for each method, the behaviour of the prediction and estimation MSE as information is added in the training dataset. Let us mention the absence of discernible changes as soon as $M > 200$. As expected, we notice on the right panel that adding information from new individuals improves the estimation of μ_0, leading to shallow errors for high values of M, in particular for MAGMA. Meanwhile, the left panel exhibits reasonably unchanged prediction performance with respect to the values of M, excepted some random fluctuations. This property is expected for GP regression, since no external information is used from the training dataset in this context. For both multi-tasks algorithms though, the estimation of μ_0 improves the prediction by one order of magnitude below the typical errors, even with only a few training individuals. Furthermore, since a new individual behaves independently through f_*, it is natural for a 10-points-ahead forecast to present intrinsic variations, despite an adequate estimation of the shared mean process.

To illustrate the advantage of multi-task methods, even for $M = 20$, we display on Figure 3.4 the evolution of MSE according to the number of timestamps N that are assumed to be observed for the new individual on which we make predictions. These predictions remain computed on the last 10 timestamps, although in this experiment, we only observe the first 5, 10, 15, or 20 timestamps, in order to change the volume of information and the distance from training observations to targets. We observe on Figure 3.3 that, as expected in a GP framework, the closer observations are to targets, the better the results. However, for multi-tasks approaches and in particular for MAGMA, the prediction remains consistently adequate even with few observations. Once more, sharing information across individuals significantly helps the prediction, even for small values of M or few observed data.

3.6.3 MAGMA's specific settings

As we previously discussed, different settings are available for MAGMA according to the nature of data and the model hypotheses. First, the *Common grid* setting corresponds to cases where all individuals share the same timestamps, whereas *Uncommon grid* is used otherwise. Moreover, MAGMA enables to consider identical hyper-parameters for all individuals or specific ones, as previously discussed in Section 3.2.2. To evaluate the effect of the different settings, performances in prediction and μ_0's estimation are evaluated in the following cases in Table 3.2:

- *Common HP*, when data are simulated with a common set of hyper-parameters for all individuals, and Proposition 3.3 is used for inference in MAGMA,
- *Different HP*, when data are simulated with its own set of hyper-parameters for each individual, and Proposition 3.2 is used for inference in MAGMA,
- *Common HP on different HP data*, when data are simulated with its own set of hyper-parameters for each individual, and Proposition 3.3 is used for inference in MAGMA.

Note that the first line of the table (*Common grid / Common HP*) of Table 3.2 is identical to the corresponding results in Table 3.1, providing reference values, significantly better than for other methods. The results obtained in Table 3.2 indicates that the MAGMA performance are not significantly altered by the settings used, or the nature of the simulated data. In order to confirm the robustness of the method, the setting *Common HP* was applied to data generated by drawing different values of hyper-parameters for each individual (*Different HP*

		Prediction		Estimation of μ_0	
		MSE	CI_{95}	MSE	CI_{95}
Common HP	Common grid	18.7 (31.4)	93.8 (13.5)	1.3 (2)	94.3 (11.3)
	Uncommon grid	19.2 (43)	94.6 (13.1)	2.9 (2.6)	93.6 (9.2)
Different HP	Common grid	19.9 (54.7)	91.6 (17.8)	0.5 (0.4)	70.8 (24.3)
	Uncommon grid	14.5 (22.4)	89.1 (17.9)	2.5 (4.5)	81.1 (15.9)
Common HP on different HP data	Common grid	21.7 (36)	91 (19.8)	1.5 (1.2)	91.1 (13)
	Uncommon grid	18.1 (33)	92.5 (15.9)	3.2 (4.5)	93.4 (9.8)

Table 3.2 – Average MSE (sd) and average CI_{95} coverage (sd) on 100 runs for the different settings of MAGMA.

data). In this case, performance in prediction and estimation of μ_0 are slightly deteriorated, although MAGMA still provides quite reliable forecasts. This experience also highlights a particularity of the *Different HP* setting: looking at the estimation of μ_0 performance, we observe a significant decrease in the CI_{95} coverage, due to numerical instability in some pathological cases. Numerical issues, in particular during matrix inversions, are classical problems in the GP literature and, because of the potentially large number of different hyper-parameters to train, the probability for at least one of them to lead to a nearly singular matrix increases. In this case, one individual might overwhelm others in the calculus of μ_0's hyper-posterior (see Proposition 3.4), and thus lead to an underestimated posterior variance. This problem does not occur in the *Common HP* settings, since sharing the same hyper-parameters prevents the associated covariance matrices from running over each other. Thus, except if one specifically wants to smooth multiple curves presenting really different behaviours, keeping *Common HP* as a default setting appear as a reasonable choice. Let us notice that the estimation of μ_0 is slightly better for common than for uncommon grid, since the estimation problem on the union of different timestamps is generally more difficult. However, this feature only depends on the nature of data.

3.6.4 Running times comparisons

The counterpart of the more accurate and general results provided by MAGMA is a natural increase in running time. Table 3.3 exhibits the raw and relative training times for GPFDA and MAGMA (prediction times are negligible and comparable in both cases), with varying values of M on a *Common grid* of $N = 30$ timestamps. The algorithms were run under the *3.6.1 R version*, on a laptop with a dual-core processor cadenced at 2.90GhZ and an 8Go RAM. The reported computing times are in seconds, and for small to moderate datasets ($N \simeq 10^3$, $M \simeq 10^4$) the procedures ran in few minutes to few hours. The difference between the two algorithms is due to GPFDA modelling μ_0 as a deterministic function through B-splines smoothing, whereas MAGMA accounts for uncertainty. The ratio of computing times between the two methods tends to decrease as M increases, and stabilises around 2 for higher numbers of training individuals. This behaviour comes from the E step in MAGMA, which is incompressible and quite insensitive to the value of M. Roughly speaking, one needs to pay twice the computing price of GPFDA for MAGMA to provide (significantly) more accurate predictions and uncertainty over μ_0. Table 3.4 provides running times of MAGMA according to its different settings, with $M = 20$. Because the complexity is linear in M in each case, the ratio in running times would remain roughly similar no matter the value of M. Prediction time appears negligible compared to training time, and generally takes less than one second to run. Besides, the *Different HP* setting increases the running time,

since in this context M maximisations (instead of one for *Common HP*) are required at each EM iteration. In this case, the prediction also takes slightly longer because of the necessity to optimise hyper-parameters for the new individual. Although the nature of the grid of timestamps does not matter in itself, a key limitation lies in the dimension N of the pooled set of timestamps, which tends to get bigger when individuals have different timestamps from one another.

	5	10	50	100
MAGMA	5.2 (2.7)	7.6 (3.2)	24.2 (11.1)	42.8 (10)
GPFDA	1 (0.3)	2.1 (0.6)	10.7 (2.4)	23.1 (5.3)
Ratio	**5.2**	**3.6**	**2.3**	**1.9**

Table 3.3 – Average (sd) training time (in seconds) for MAGMA and GPFDA for different numbers M of individuals in the training dataset. The relative running time between MAGMA and GPFDA is provided on the line *Ratio*.

		Train	Predict
Common HP	Common grid	12.6 (3.5)	0.1 (0)
	Uncommon grid	16.5 (11.4)	0.2 (0.1)
Different HP	Common grid	42.6 (20.5)	0.6 (0.1)
	Uncommon grid	40.2 (17)	0.6 (0.1)

Table 3.4 – Average (sd) training and prediction time (in seconds) for different settings of MAGMA.

3.6.5 Application of MAGMA on swimmers' progression curves

DATA AND PROBLEMATIC

We consider the problem of performance prediction in competition for french swimmers. The French Swimming Federation (FFN) provided us with an anonymised dataset, compiling the age and results of its members between 2000 and 2016. For each competitor, the race times are registered for competitions of 100m freestyle (50m swimming-pool). The database contains results from 1731 women and 7876 men, each of them compiling an average of 22.2 data points (min = 15, max = 61) and 12 data points (min = 5, max = 57) respectively. In the following, age of the i-th swimmer is considered as the input variable (timestamp t) and the performance (in seconds) on a 100m freestyle as the output ($y_i(t)$). For reasons of confidentiality and property, the raw dataset cannot be published. The analysis focuses on the youth period, from 10 to 20 years, where the progression is the most noticeable. In order to get relevant time series, we retained only individuals having a sufficient number of data points on the considered time period. For a young swimmer, observed during its first years of competition, we aim at modelling its progression curve and make predictions on its future performance in the subsequent years. Since we consider a decision-making problem involving irregular time series, the GP probabilistic framework is a natural choice to work on. Thereby, assuming that each swimmer in the database is a realisation y_i defined as previously, we expect MAGMA to provide multi-task predictions for a new young swimmer, that will benefit from information of other swimmers already observed at older ages. To study such modelling, and validate its efficiency in practice, we split the individuals into a training and testing datasets with respective sizes:

- $M^F_{train} = 1039$, for the female training set,
- $M^F_{test} = 692$, for the female testing set,
- $M^M_{train} = 4726$, for the male training set,
- $M^M_{test} = 3150$, for the male testing set.

Inference on the hyper-parameters is performed thanks to the training dataset in both cases. Considering the different timestamps and the relative monotony of the progression curves, the settings *Uncommon grid*/*Common HP* has been used for MAGMA. The overall training lasted around 2 hours with the same hardware configuration as for simulations. To compute MSE and the CI_{95} coverage, the data points of each individual in the testing set has been split into *observed* and *testing* timestamps. Since each individual has a different number of data points, the first 80% of timestamps are taken as *observed*, while the remaining 20% are considered as *testing* timestamps. MAGMA's predictions are compared with the true values of y_i at testing timestamps. As previously, both GP and MAGMA have been initialised with a constant 0 mean function. Initial values for hyper-parameters are also similar for all i, $\theta_0^{ini} = \theta_i^{ini} = (e^1, e^1)$ and $\sigma_i^{ini} = 0.4$. Those values are the default in MAGMA and remain adequate in the context of these datasets.

RESULTS AND INTERPRETATION The overall performance and comparison are summarised in Table 3.5.

		MSE	CI_{95}
Women	MAGMA	**3.8 (10.3)**	**95.3 (15.9)**
	GP	25.3 (97.6)	72.7 (37.1)
Men	MAGMA	**3.7 (5.3)**	**93.9 (15.3)**
	GP	22.1 (94.3)	78.2 (30.4)

Table 3.5 – Average MSE (sd) and average CI_{95} coverage (sd) for prediction on french swimmer testing datasets.

We observe that MAGMA still provides excellent results in this context, and naturally outperform predictions provided by a single GP regression. The progression curves presenting relatively monotonic variations, and thus avoiding pathological behaviours that could occur with synthetic data, the MSE in prediction remains very low. The CI_{95} coverage sticks close to the 95% expected value for MAGMA, indicating an adequate quantification of uncertainty. To illustrate these results, an example is displayed on Figure 3.5 for both men and women. For a randomly chosen testing individual, we plot its predicted progression curve (in blue), where its first 15 data points are used as observations (in black), while the remaining true data points (in red) are displayed for comparison purpose. As previously observed in the simulation study, the simple GP quickly drifts to the prior 0 mean, as soon as data lack. However, for both men and women, the MAGMA predictions remain close to the true data, which also lie within the 95% credible interval. Even for long term forecast, where the mean prediction curve tends to overlap the mean process (dashed line), the true data remain in our range of uncertainty, as the credible interval widens far from observations. For clarity, we displayed only a few individuals from the training dataset (colourful points) in the background. The mean process (dashed line) seems to represent the main trend of

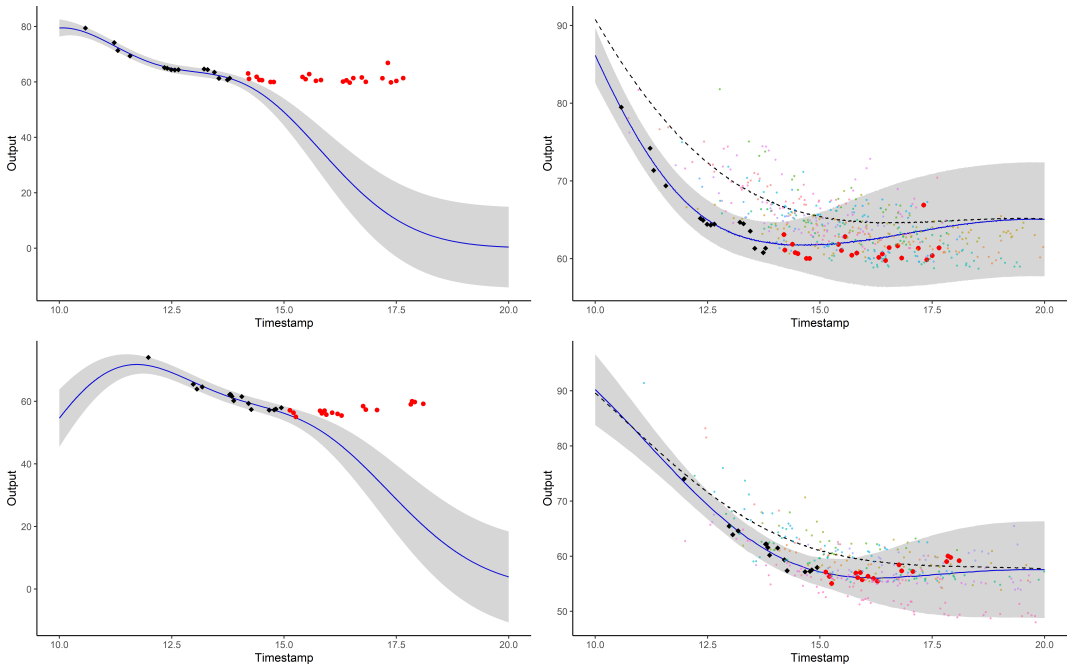

Figure 3.5 – Prediction curves (blue) for a testing individual with associated 95% credible intervals (grey) for GP regression (left) and MAGMA (right), for both women (top) and men (bottom). The dashed lines represent the mean functions of the hyper-posterior mean process $\mu_0 \mid \{\mathbf{y}_i\}_i$. Observed data points are in black, and testing data points are in red. The colourful backward points are observations from the training dataset, each colour corresponding to a different individual.

progression among swimmers correctly, even though we cannot numerically compare μ_0 to any real-life analogous quantity. In a more sport-related perspective, we can note that both genders present similar patterns of progression. However, while performances are roughly similar in mean trend before the age of 14, they start to differentiate afterwards and then converge to average times with approximatively a 5 seconds gap. Interestingly, the difference between world records in 100 freestyle for men and women is currently 4.8 seconds (46.91 versus 51.71). These results, obtained under reasonable hypotheses on several hundreds of swimmers, seem to indicate that MAGMA would give quite reliable predictions for a new young swimmer. Furthermore, the uncertainty provided through the predictive posterior distribution offers an adequate degree of caution in a decision-making process.

3.7 Discussion

We have introduced a unified non-parametric multi-task framework integrating a mean Gaussian process prior in the context of GP regression. While we believe that this process is an interesting object in itself, it also allows individuals to borrow information from each other and provide more accurate predictions, even far from data points. Furthermore, our method accounts for uncertainty in the mean process and remains applicable no matter the observational grid of data. Both on simulated and real-life datasets, we exhibited the adequacy of such an approach, and studied some of its properties and possible settings. MAGMA outper-

forms the alternatives in estimation of the mean process as well as in prediction, and gives a reliable quantification of uncertainty. We also displayed evidence of its predictive efficiency for real-life problems and provided some insights on practical interpretation about the mean process.

Interestingly, despite the extensive literature on these aspects of GPs, our model does not yet include sparse approximations or on-line extensions. While these aspects are beyond the scope of the present chapter, we aim to integrate such existing approaches in our model to widen its applicability. The combination of the covariance structures used in classical multi-task GP (Bonilla et al., 2008; Hensman et al., 2013) with the common mean process we introduced would also open a promising path for future work. Another possible avenue is an adaptation to the classification context, which is presented in Rasmussen and Williams (2006, Chapter 3). Besides, this work leaves the door open to improvement as we only tackled the problem of unidimensional regression: enabling either multidimensional or mixed type of inputs as in Shi and Choi (2011) would be of interest. To conclude, the hypothesis of a unique underlying mean process might be considered as too restrictive for some datasets, and enabling cluster-specific mean processes would be a relevant extension.

3.8 Proofs

The proof below gives details for the calculus of μ_0's hyper-posterior distribution, involved in the E step of the EM algorithm and during the prediction process. Although the main idea is similar to the proof given for common timestamps, there are some cautions to take when working in the general case. Note that the proof of Proposition 3.1 is a particular case of the proof below, where $\boldsymbol{\tau} = \mathbf{t}$ exactly (where $\boldsymbol{\tau}$ is the set of timestamps the hyper-posterior is to be computed on). Moreover, in order to keep an analytical expression for μ_0's hyper-posterior distribution, we discard the superfluous information contained in $\{\mathbf{y}_i\}_i$ at timestamps on which the hyper-posterior is not to be computed. Hence, the proof below states that the remaining data points are observed on subsets $\{\boldsymbol{\tau}_i\}_i$ of $\boldsymbol{\tau}$.

3.8.1 Proof of Proposition 3.1 and Proposition 3.4

Let $\boldsymbol{\tau}$ be a finite vector of timestamps, and $\{\boldsymbol{\tau}_i\}_i$ such as $\forall i = 1, \dots, M, \quad \boldsymbol{\tau}_i \subset \boldsymbol{\tau}$. We define convenient notation:

- $\boldsymbol{\mu}_0^{\boldsymbol{\tau}} = \mu_0(\boldsymbol{\tau})$,
- $\mathbf{m}_0^{\boldsymbol{\tau}} = m_0(\boldsymbol{\tau})$,
- $\boldsymbol{\mu}_0^{\boldsymbol{\tau}_i} = \mu_0(\boldsymbol{\tau}_i), \ \forall i = 1, \dots, M$,
- $\mathbf{y}_i^{\boldsymbol{\tau}_i} = y_i(\boldsymbol{\tau}_i), \ \forall i = 1, \dots, M$,
- $\boldsymbol{\Psi}_i = \psi_{\theta_i, \sigma_i^2}(\boldsymbol{\tau}_i, \boldsymbol{\tau}_i), \forall i = 1, \dots, M$,
- $\mathbf{K} = k_{\theta_0}(\boldsymbol{\tau}, \boldsymbol{\tau})$.

Moreover, for a covariance matrix C, and $u, v \in \boldsymbol{\tau}$, we note $[C]^{-1}_{uv}$ the element of the inverse matrix at row associated with timestamp u, and column associated with timestamp

v. We also ignore the conditionings over $\hat{\Theta}$, $\boldsymbol{\tau}_i$ and $\boldsymbol{\tau}$ to maintain simple expressions. By construction of the models, we have:

$$\begin{aligned} p(\boldsymbol{\mu}_0^{\boldsymbol{\tau}} \mid \{\mathbf{y}_i^{\boldsymbol{\tau}_i}\}_i) &\propto p(\{\mathbf{y}_i^{\boldsymbol{\tau}_i}\}_i \mid \boldsymbol{\mu}_0^{\boldsymbol{\tau}}) p(\boldsymbol{\mu}_0^{\boldsymbol{\tau}}) \\ &\propto \left\{ \prod_{i=1}^{M} p(\mathbf{y}_i^{\boldsymbol{\tau}_i} \mid \boldsymbol{\mu}_0^{\boldsymbol{\tau}_i}) \right\} p(\boldsymbol{\mu}_0^{\boldsymbol{\tau}}) \\ &\propto \left\{ \prod_{i=1}^{M} \mathcal{N}\left(\mathbf{y}_i^{\boldsymbol{\tau}_i}; \boldsymbol{\mu}_0^{\boldsymbol{\tau}_i}, \boldsymbol{\Psi}_i\right) \right\} \mathcal{N}\left(\boldsymbol{\mu}_0^{\boldsymbol{\tau}}; \mathbf{m}_0^{\boldsymbol{\tau}}, \mathbf{K}\right). \end{aligned}$$

The term $\mathcal{L}_1 = -(1/2) \log p(\boldsymbol{\mu}_0^{\boldsymbol{\tau}} \mid \{\mathbf{y}_i^{\boldsymbol{\tau}_i}\}_i)$ associated with the hyper-posterior remains quadratic and we may find the corresponding Gaussian parameters by identification:

$$\begin{aligned} \mathcal{L}_1 &= \sum_{i=1}^{M} \left\{ (\mathbf{y}_i^{\boldsymbol{\tau}_i} - \boldsymbol{\mu}_0^{\boldsymbol{\tau}_i})^\intercal \boldsymbol{\Psi}_i^{-1} (\mathbf{y}_i^{\boldsymbol{\tau}_i} - \boldsymbol{\mu}_0^{\boldsymbol{\tau}_i}) + C_i \right\} + (\boldsymbol{\mu}_0^{\boldsymbol{\tau}} - \mathbf{m}_0^{\boldsymbol{\tau}})^\intercal \mathbf{K}^{-1} (\boldsymbol{\mu}_0^{\boldsymbol{\tau}} - \mathbf{m}_0^{\boldsymbol{\tau}}) + C_0 \\ &= \boldsymbol{\mu}_0^{\boldsymbol{\tau}\intercal} \mathbf{K}^{-1} \boldsymbol{\mu}_0^{\boldsymbol{\tau}} + \sum_{i=1}^{M} \boldsymbol{\mu}_0^{\boldsymbol{\tau}_i\intercal} \boldsymbol{\Psi}_i^{-1} \boldsymbol{\mu}_0^{\boldsymbol{\tau}_i} - 2 \left(\boldsymbol{\mu}_0^{\boldsymbol{\tau}\intercal} \mathbf{K}^{-1} \mathbf{m}_0^{\boldsymbol{\tau}} + \sum_{i=1}^{M} \boldsymbol{\mu}_0^{\boldsymbol{\tau}_i\intercal} \boldsymbol{\Psi}_i^{-1} \mathbf{y}_i^{\boldsymbol{\tau}_i} \right) + C \\ &= \sum_{u \in \boldsymbol{\tau}} \sum_{v \in \boldsymbol{\tau}} \mu_0(u) \left[\mathbf{K}\right]_{uv}^{-1} \mu_0(v) + \sum_{i=1}^{M} \sum_{u \in \boldsymbol{\tau}_i} \sum_{v \in \boldsymbol{\tau}_i} \mu_0(u) \left[\boldsymbol{\Psi}_i\right]_{uv}^{-1} \mu_0(v) \\ &\quad - 2 \sum_{u \in \boldsymbol{\tau}} \sum_{v \in \boldsymbol{\tau}} \mu_0(u) \left[\mathbf{K}\right]_{uv}^{-1} m_0(v) - 2 \sum_{i=1}^{M} \sum_{u \in \boldsymbol{\tau}_i} \sum_{v \in \boldsymbol{\tau}_i} \mu_0(u) \left[\boldsymbol{\Psi}_i\right]_{uv}^{-1} y_i(v) + C, \end{aligned}$$

where we entirely decomposed the vector-matrix products. We factorise the expression according to the common timestamps between τ_i and τ. Since for all $i, \boldsymbol{\tau}_i \subset \boldsymbol{\tau}$, let us introduce a dummy indicator function $\mathbb{1}_{\boldsymbol{\tau}_i} = \mathbb{1}_{\{u,v \in \boldsymbol{\tau}_i\}}$ to write:

$$\begin{aligned} \sum_{i=1}^{M} \sum_{u \in \boldsymbol{\tau}_i} \sum_{v \in \boldsymbol{\tau}_i} A(u,v) &= \sum_{i=1}^{M} \sum_{u \in \boldsymbol{\tau}} \sum_{v \in \boldsymbol{\tau}} \mathbb{1}_{\boldsymbol{\tau}_i} A(u,v) \\ &= \sum_{u \in \boldsymbol{\tau}} \sum_{v \in \boldsymbol{\tau}} \sum_{i=1}^{M} \mathbb{1}_{\boldsymbol{\tau}_i} A(u,v), \end{aligned}$$

subsequently, we can gather the sums such as:

$$
\begin{aligned}
\mathcal{L}_1 &= \sum_{u\in\boldsymbol{\tau}}\sum_{v\in\boldsymbol{\tau}}\Bigg(\mu_0(u)\,[\,\mathbf{K}\,]^{-1}_{uv}\,\mu_0(v) + \sum_{i=1}^{M}\mathbb{1}_{\boldsymbol{\tau}_i}\mu_0(u)\,[\,\boldsymbol{\Psi}_i\,]^{-1}_{uv}\,\mu_0(v) \\
&\qquad\qquad - 2\mu_0(u)\,[\,\mathbf{K}\,]^{-1}_{uv}\,m_0(v) - 2\sum_{i=1}^{M}\mathbb{1}_{\boldsymbol{\tau}_i}\mu_0(u)\,[\,\boldsymbol{\Psi}_i\,]^{-1}_{uv}\,y_i(v)\Bigg) + C \\
&= \sum_{u\in\boldsymbol{\tau}}\sum_{v\in\boldsymbol{\tau}}\Bigg(\mu_0(u)\Big([\,\mathbf{K}\,]^{-1}_{uv} + \sum_{i=1}^{M}\mathbb{1}_{\boldsymbol{\tau}_i}\,[\,\boldsymbol{\Psi}_i\,]^{-1}_{uv}\Big)\mu_0(v) \\
&\qquad\qquad - 2\mu_0(u)\Big([\,\mathbf{K}\,]^{-1}_{uv}\,m_0(v) + \sum_{i=1}^{M}\mathbb{1}_{\boldsymbol{\tau}_i}\,[\,\boldsymbol{\Psi}_i\,]^{-1}_{uv}\,y_i(v)\Big)\Bigg) + C \\
&= \boldsymbol{\mu}_0^{\boldsymbol{\tau}\intercal}\Big(\mathbf{K}^{-1} + \sum_{i=1}^{M}\tilde{\boldsymbol{\Psi}}_i^{-1}\Big)\boldsymbol{\mu}_0^{\boldsymbol{\tau}} - 2\boldsymbol{\mu}_0^{\boldsymbol{\tau}\intercal}\Big(\mathbf{K}^{-1}\mathbf{m}_0^{\boldsymbol{\tau}} + \sum_{i=1}^{M}\tilde{\boldsymbol{\Psi}}_i^{-1}\tilde{\mathbf{y}}_i^{\boldsymbol{\tau}}\Big) + C,
\end{aligned}
$$

where the $\mathbf{y}_i$ and $\boldsymbol{\Psi}_i$ are completed by zeros:

- $\tilde{\mathbf{y}}_i^{\boldsymbol{\tau}} = \mathbb{1}_{\boldsymbol{\tau}_i} y_i(\boldsymbol{\tau})$,
- $\left[\,\tilde{\boldsymbol{\Psi}}_i\,\right]^{-1}_{uv} = \mathbb{1}_{\boldsymbol{\tau}_i}\,[\,\boldsymbol{\Psi}_i\,]^{-1}_{uv}\,,\ \forall u,v\in\boldsymbol{\tau}$.

By identification of the quadratic form, we reach:

$$
p(\boldsymbol{\mu}_0^{\boldsymbol{\tau}} \mid \{\mathbf{y}_i^{\boldsymbol{\tau}_i}\}_i) = \mathcal{N}\left(\boldsymbol{\mu}_0^{\boldsymbol{\tau}}; \hat{m}_0(\boldsymbol{\tau}), \hat{\mathbf{K}}\right),
$$

with,

- $\hat{\mathbf{K}} = \left(\mathbf{K}^{-1} + \sum_{i=1}^{M}\tilde{\boldsymbol{\Psi}}_i^{-1}\right)^{-1}$,
- $\hat{m}_0(\boldsymbol{\tau}) = \hat{\mathbf{K}}\left(\mathbf{K}^{-1}\mathbf{m}_0^{\boldsymbol{\tau}} + \sum_{i=1}^{M}\tilde{\boldsymbol{\Psi}}_i^{-1}\tilde{\mathbf{y}}_i^{\boldsymbol{\tau}}\right)$.

3.8.2 Proof of Proposition 3.2 and Proposition 3.3

Since the central part of the proofs is similar for both propositions, we detail the calculus denoting $\Theta = \{\theta_0, \{\theta_i\}_i, \{\sigma_i^2\}_i\}$ for generality and dissociate the two cases only when necessary. Before considering the maximisation, we notice that the joint density can be developed as:

$$
\begin{aligned}
\mathcal{L}_\in &= p(\{\mathbf{y}_i\}_i, \mu_0(\mathbf{t}) \mid \Theta) \\
&= p(\{\mathbf{y}_i\}_i \mid \mu_0(\mathbf{t}), \Theta)\ p(\mu_0(\mathbf{t}) \mid \Theta) \\
&= \prod_{i=1}^{M}\left\{p(\mathbf{y}_i \mid \mu_0(\mathbf{t}), \theta_i, \sigma_i^2)\right\}\ p(\mu_0(\mathbf{t}) \mid \theta_0) \\
&= \prod_{i=1}^{M}\left\{\mathcal{N}(\mathbf{y}_i; \mu_0(\mathbf{t}), \boldsymbol{\Psi}_{\theta_i,\sigma_i^2})\right\}\mathcal{N}(\mu_0(\mathbf{t}); m_0(\mathbf{t}), \mathbf{K}^{\mathbf{t}}_{\theta_0}).
\end{aligned}
$$

The expectation is taken over $p(\mu_0(\mathbf{t}) \mid \{\mathbf{y}_i\}_i)$ though we write it $\mathbb{E}$ for simplicity. We have:

$$\begin{aligned} f(\Theta) &= \mathbb{E}\left[\log \mathcal{L}_{\in}\right] \\ &= -\frac{1}{2}\mathbb{E}\Bigg[(\mu_0(\mathbf{t}) - m_0(\mathbf{t}))^\intercal {\mathbf{K}^{\mathbf{t}}_{\theta_0}}^{-1}(\mu_0(\mathbf{t}) - m_0(\mathbf{t})) - \log\left|{\mathbf{K}^{\mathbf{t}}_{\theta_0}}^{-1}\right| \\ &\qquad + \sum_{i=1}^{M}(\mathbf{y}_i - \mu_0(\mathbf{t}_i))^\intercal {\boldsymbol{\Psi}^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}}^{-1}(\mathbf{y}_i - \mu_0(\mathbf{t}_i)) - \log\left|{\boldsymbol{\Psi}^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}}^{-1}\right|\Bigg] + C_1. \end{aligned}$$

Lemma 3.1. *Let $X \in \mathbb{R}^N$ be a random Gaussian vector $X \sim \mathcal{N}(m, \mathbf{K})$, $b \in \mathbb{R}^N$, and $\mathbf{S}$, a $N \times N$ covariance matrix. Then:*

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

Lemma 3.1.

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

□

As we note that X and b play symmetrical roles in the calculus of the conditional expectation, we can apply the lemma regardless to the position of μ_0 in the $M+1$ equalities involved. Applying Lemma 3.1 to our previous expression of $f(\Theta)$, we obtain:

$$\begin{aligned} f(\Theta) = -\frac{1}{2}\Bigg[&(\hat{m}_0(\mathbf{t}) - m_0(\mathbf{t}))^\intercal {\mathbf{K}^{\mathbf{t}}_{\theta_0}}^{-1}(\hat{m}_0(\mathbf{t}) - m_0(\mathbf{t})) \\ &+ \sum_{i=1}^{M}(\mathbf{y}_i - \hat{m}_0(\mathbf{t}_i))^\intercal {\boldsymbol{\Psi}^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}}^{-1}(\mathbf{y}_i - \hat{m}_0(\mathbf{t}_i)) \\ &+ Tr\left(\hat{\mathbf{K}}^{\mathbf{t}}{\mathbf{K}^{\mathbf{t}}_{\theta_0}}^{-1}\right) + \sum_{i=1}^{M} Tr\left(\hat{\mathbf{K}}^{\mathbf{t}_i}{\boldsymbol{\Psi}^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}}^{-1}\right) \\ &- \log\left|{\mathbf{K}^{\mathbf{t}}_{\theta_0}}^{-1}\right| - \sum_{i=1}^{M}\log\left|{\boldsymbol{\Psi}^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}}^{-1}\right| + C_1\Bigg]. \end{aligned}$$

We recall that, at the M step, $\hat{m}_0(\mathbf{t})$ is a known constant, computed at the previous E step. Thus, we identify here the characteristic expression of several Gaussian log-likelihoods and associated correction trace terms. Moreover, each set of hyper-parameters is merely involved in independent terms of the whole function to maximise. Hence, the global maximisation problem can be separated into several maximisations of sub-functions according to the hyper-parameters getting optimised. Regardless to additional assumptions, the hyper-parameters θ_0, controlling the covariance matrix of the mean process, appears in a function

which is exactly a Gaussian log-likelihood $\log \mathcal{N}\left(\hat{m}_0(\mathbf{t}), m_0(\mathbf{t}), \mathbf{K}^{\mathbf{t}}_{\theta_0}\right)$, added to a corresponding trace term $-\frac{1}{2} Tr\left(\hat{\mathbf{K}}^{\mathbf{t}} {\mathbf{K}^{\mathbf{t}}_{\theta_0}}^{-1}\right)$. This function can be maximised independently from the other parameters, giving the first part of the results in Proposition 3.2 and Proposition 3.3.

Although the idea is analogous for the remaining hyper-parameters, we have to discriminate here regarding the assumption on the model. If each individual is supposed to have its own set $\{\theta_i, \sigma_i\}$, which thus can be optimised independently from the observations and hyper-parameters of other individuals, we identify a sum of M Gaussian log-likelihoods $\log \mathcal{N}\left(\mathbf{y}_i, \hat{m}_0(\mathbf{t}_i), \mathbf{\Psi}^{\mathbf{t}_i}_{\theta_i, \sigma_i^2}\right)$ and the corresponding trace terms $-\frac{1}{2} Tr(\hat{\mathbf{K}}^{\mathbf{t}} {\mathbf{\Psi}^{\mathbf{t}_i}_{\theta_i, \sigma_i^2}}^{-1})$. This results on M independent maximisation problems on corresponding functions, proving Proposition 3.2. Conversely, if we assume that all individuals in the model shares their hyper-parameters and $\left\{\theta, \sigma^2\right\} = \left\{\theta_i, \sigma_i^2\right\}, \forall i$, we can no longer divide the problem into M sub-maximisations, and the whole sum on all individual should be optimised thanks to observations from all individuals. This case corresponds to the second part of Proposition 3.3.

Availability of data

The synthetic data and table of results are available at https://github.com/ArthurLeroy/MAGMA/tree/master/Simulations

Code availability

The R code associated with the present work is available at https://github.com/ArthurLeroy/MAGMA

4

Curve clustering and cluster-specific multi-task GP regression

This chapter is based on the article Leroy et al. (2020a), which is currently under review.

4.1 Introduction

The present chapter proposes an extension to the model presented in Chapter 3 by introducing a clustering component into the procedure. This approach relies on the definition of a GPs mixture model that we combine with our previously introduced multi-task aspect. Such modelling offers both new results in terms of possible group structures in the data and enhanced predictive abilities, by sharing information across the individuals through multiple cluster-specific mean processes instead of a single. The chapter is organised as follows. We introduce the multi-task Gaussian processes mixture model in Section 4.2, along with notation. Section 4.3 is devoted to the inference procedure, with a Variational Expectation-Maximisation (VEM) algorithm to estimate hyper-parameters and approximation of hyper-posterior distributions along with mixture parameters. We leverage this strategy in Section 4.4 and derive both a mixture and cluster-specific GP prediction formulas, for which we provide an analysis along with computational costs in Section 4.5. The performances or our algorithm for clustering and prediction purposes are illustrated in Section 4.6 with a series of experiments on both synthetic and real-life datasets and a comparison to competing state-of-the-art algorithms. Then, Section 4.7 depicts an overall point-of-view on this work. Finally, we defer all proofs to original results to Section 4.8.

4.2 Modelling

4.2.1 Notation

In order to remain consistent both with the vocabulary introduced in Chapter 3 and with the illustrative example in Section 4.6, we refer to the input variables as *timestamps* and use the term *individual* as a synonym of batch or task. However, although the temporal formulation helps to wrap the mind around the concepts, the present framework still applies to the wide range of data one can usually think of in GP models. As we suppose the dataset to be composed of point-wise observations from multiple functions, the set of all indices is denoted by $\mathcal{I} \subset \mathbb{N}$, which in particular contains $\{1, \dots, M\}$, the indices of the observed individuals (i.e. the training set). The input values being defined over a continuum, let us name $\mathcal{T}$ this input space (we can assume $\mathcal{T} \subset \mathbb{R}$ here for simplicity). Moreover, since the following model is defined for clustering purposes, the set of indices $\mathcal{K} = \{1, \dots, K\}$ refers to the K different groups of individuals. For the sake of concision, let us also shorten the notation as follows: for any object x, $\{x_i\}_i = \{x_1, \dots, x_M\}$ and $\{x_k\}_k = \{x_1, \dots, x_K\}$.

We assume to collect data from M different sources, such as a set of N_i input-output values $\left\{\left(t_i^1, y_i(t_i^1)\right), \dots, \left(t_i^{N_i}, y_i(t_i^{N_i})\right)\right\}$ constitutes the observations for the i-th individual. Below follows additional convenient notation:

- $\mathbf{t}_i = \{t_i^1, \dots, t_i^{N_i}\}$, the set of timestamps for the i-th individual,
- $\mathbf{y}_i = y_i(\mathbf{t}_i)$, the vector of outputs for the i-th individual,

- $\mathbf{t} = \bigcup_{i=1}^{M} \mathbf{t}_i$, the pooled set of all timestamps among individuals,
- $N = card(\mathbf{t})$, the total number of observed timestamps.

Let us stress that the input values may vary both in number and location among individuals, and we refer as a *common grid* of timestamps to the case where $\mathbf{t}_i = \mathbf{t},\ \forall i \in \mathcal{I}$. Otherwise, we call it an *uncommon grid*. Besides, in order to define a GP mixture model, a latent binary random vector $Z_i = (Z_{i1}, \ldots, Z_{iK})^\intercal$ needs to be associated with each individual, indicating in which cluster it belongs. Namely, if the i-th individual comes from the k-th cluster, then $Z_{ik} = 1$ and 0 otherwise. Moreover, we assume these latent variables to come from the same multinomial distribution: $Z_i \sim \mathcal{M}(1, \boldsymbol{\pi}),\ \forall i \in \mathcal{I}$, with a vector of mixing proportions $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)^\intercal$ and $\sum_{k=1}^{K} \pi_k = 1$.

4.2.2 Model and assumptions

Assuming that the i-th individual belongs to the k-th group, we can define its functional expression as the sum of a cluster-specific mean process and an individual-specific centred process:

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

where:

- $\mu_k(\cdot) \sim \mathcal{GP}(m_k(\cdot), c_{\theta_k}(\cdot,\cdot))$ is the common mean process of the k-th cluster,
- $f_i(\cdot) \sim \mathcal{GP}(0, \xi_{\theta_i}(\cdot,\cdot))$ is the specific process of the i-th individual,
- $\varepsilon_i(\cdot) \sim \mathcal{GP}(0, \sigma_i^2 I)$ is the error term.

This general model depends upon several mean and covariance parameters, fixed as modelling choices, and hyper-parameters to be estimated:

- $\forall k \in \mathcal{K},\ m_k(\cdot)$ is the prior mean function of the k-th cluster,
- $\forall k \in \mathcal{K},\ c_{\gamma_k}(\cdot,\cdot)$ is the covariance kernel of hyper-parameters γ_k,
- $\forall i \in \mathcal{I},\ \xi_{\theta_i}(\cdot,\cdot)$ is the covariance kernel of hyper-parameters θ_i,
- $\forall i \in \mathcal{I},\ \sigma_i^2 \in \mathbb{R}$ is the noise variance associated with the i-th individual,
- $\forall i \in \mathcal{I}$, we define the shorthand $\psi_{\theta_i,\sigma_i^2}(\cdot,\cdot) = \xi_{\theta_i}(\cdot,\cdot) + \sigma_i^2 I$,
- $\Theta = \left\{ \{\gamma_k\}_k, \{\theta_i\}_i, \{\sigma_i^2\}_i, \boldsymbol{\pi} \right\}$, the set of all hyper-parameters of the model.

Let us note that we assume here the error term to be individual-specific, although we could also assume it to be cluster-specific and thus indexed by k. Such a choice would result in a valid model since the upcoming developments remain tractable if we substitute ε_k to ε_i everywhere, and associate $\sigma_k^2 I$ with $c_{\gamma_k}(\cdot,\cdot)$ instead of $\xi_{\theta_i}(\cdot,\cdot)$. However, we work throughout with ε_i to remain as general as possible, and a discussion about additionally available assumptions on the covariance structures follows in Section 4.2.3. Regardless of this remark, we only seek an estimation for Θ among the above quantities, whereas the other objects are pre-specified in the model. For instance, the prior mean $m_k(\cdot)$ is usually set to zero but could also integrate experts knowledge if available. Furthermore, we assume that:

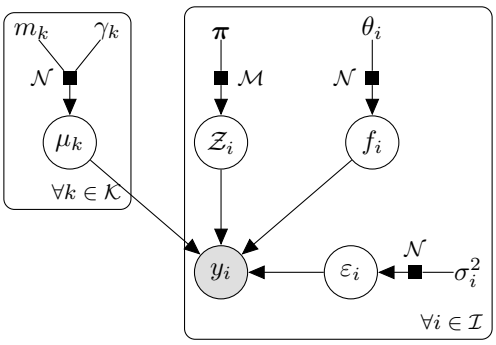

Figure 4.1 – Graphical model of dependencies between variables in the multi-task Gaussian Processes mixture model.

- $\{\mu_k\}_k$ are independent,
- $\{f_i\}_i$ are independent,
- $\{\mathbf{Z}_i\}_i$ are independent,
- $\{\varepsilon_i\}_i$ are independent,
- $\forall i \in \mathcal{I}, \forall k \in \mathcal{K},\ \mu_k,\ f_i,\ \mathbf{Z}_i$ and ε_i are independent.

We display a graphical model on Figure 4.1 to enlighten the relationships between the different components. From these hypotheses, we can naturally integrate out f_i and derive the conditional prior distribution of $y_i(\cdot)$, providing a hierarchical formulation for the model:

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

As a consequence, the output processes $\{y_i(\cdot) \mid \{\mathbf{Z}_i\}_i, \{\mu_k(\cdot)\}_k\}_i$ are also independent (conditionally to the latent variables) from one another. Although this model is expressed in terms of infinite-dimensional GPs, we proceed to the inference using finite-dimensional sets of observations $\{\mathbf{t}_i, \mathbf{y}_i\}_i$. Therefore, we can write the joint conditional likelihood of the model (conditioning on the inputs is omitted throughout the paper for clarity):

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

where $\forall i \in \mathcal{I},\ \mathbf{\Psi}_{\theta_i,\sigma_i^2}^{\mathbf{t}_i} = \psi_{\theta_i,\sigma_i^2}(\mathbf{t}_i, \mathbf{t}_i) = \left[\psi_{\theta_i,\sigma_i^2}(k,l)\right]_{k,\ell \in \mathbf{t}_i}$ is a $N_i \times N_i$ covariance matrix. The mean processes being common to all individuals in a cluster, we need to evaluate their

prior distributions on the pooled grid of timestamps $\mathbf{t}$:

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

where $\mathbf{C}^{\mathbf{t}}_{\gamma_k} = c_{\gamma_k}(\mathbf{t},\mathbf{t}) = [c_{\gamma_k}(k,\ell)]_{k,l\in\mathbf{t}}$ is a $N \times N$ covariance matrix. Finally, the joint distribution of the clustering latent variables also factorises over the individuals:

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

From all these expressions, the complete-data likelihood of the model can be derived:

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

This expression would usually serve to estimate the hyper-parameters Θ, although it depends here on latent variables that cannot be evaluated directly. Even if the prior distributions over $\{\mathbf{Z}_i\}_i$ and $\{\mu_k(\mathbf{t})\}_k$ are independent, the expressions of their respective posteriors would inevitably depend on each other. Nevertheless, it remains possible to derive variational approximations for these distributions that still factorise nicely over the terms $\mathbf{Z}_i, \forall i \in \mathcal{I}$, and $\mu_k(\mathbf{t}), \forall k \in \mathcal{K}$. Consequently, the following inference procedure involves a variational EM algorithm that we shall detail after a quick discussion on the optional hypotheses for the model.

4.2.3 Assumptions on the covariance structure

Throughout this chapter, we detail a common ground procedure that remains consistent regardless of the covariance structure of the considered GPs. Let us remark that we chose a parametric distinction of the covariance kernels through the definition of hyper-parameters, different from one individual to another. However, there are no theoretical restrictions on the underlying form of the considered kernels, and we indicate a differentiation on the sole hyper-parameters merely for convenience in writing. As already presented and used in Chapter 3, a common kernel in the GP literature is known as the *exponentiated quadratic* kernel (also called sometimes squared exponential or radial basis function kernel). This kernel only depends upon two hyper-parameters $\theta = \{v, \ell\}$ such as:

	$\theta_0 = \theta_i, \forall i \in \mathcal{I}$		$\theta_i \neq \theta_j, \forall i \neq j$	
	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

Table 4.1 – Summary of the 4 available assumptions on the hyper-parameters, with their respective shortening notation and the associated number of sets of hyper-parameters (HPs) to optimise.

$$k_{\text{EQ}}(x, x') = v^2 \exp\left(-\frac{(x - x')^2}{2\ell^2}\right). \tag{4.1}$$

The *exponentiated quadratic* kernel is used for simplicity as covariance structure for both cluster-specific and individual-specific GPs in the simulation section (see Section 4.6 for details).However, the hypotheses on the hyper-parameters are interesting to consider and offer some control over the interaction between the individuals. Let us mention the existence of a rich literature on kernel choices, properties and combinations: see Rasmussen and Williams (2006, Chapter 4) or Duvenaud (2014) for comprehensive studies. More details can also be found in Section 1.1.2.b of the present thesis.

In the initial version (see Chapter 3) and in the present chapter, the multi-task aspect is mainly supported by the mean process, although the model also allows information sharing among individual through the covariance structure. These two aspects being constructed independently, we could think of the model as potentially *double multi-task*, both in mean and covariance. More precisely, if we assume $\{\{\theta_i\}_i, \{\sigma_i^2\}_i\} = \{\theta_0, \sigma_0^2\}$, $\forall i \in \mathcal{I}$, then all f_i are assumed to be different realisations of the same GP, and thus all individuals contributes to the estimation of the common hyper-parameters. Hence, such an assumption that may appear restrictive at first glance actually offers a valuable way to share common patterns between tasks. Furthermore, the same kind of hypothesis can be proposed at the cluster level with $\{\gamma_k\}_k = \gamma_0$, $\forall k \in \mathcal{K}$. In this case, we would assume that all the clusters' mean processes $\{\mu_k\}_k$ share the same covariance structure. This property would indicate that the patterns, or the variations of the curves, are expected to be roughly identical from one cluster to another and that the differentiation would be mainly due to the mean values. Conversely, different covariance structures across kernels offer additional flexibility for the groups to differ both in position and in trend, smoothness, or any property that could be coded in a kernel. Speaking rather loosely, we may think of these different settings as a trade-off between flexibility and information sharing, or as a choice between an individual or collective modelling of the covariance. Overall, our algorithm provides 4 different settings, offering a rather wide range of assumptions for an adequate adaptation to different applicative situations. Note that the computational considerations are also of paramount importance when it comes to optimising a likelihood over a potentially high number of parameters. Hence, we display on Table 4.1 a summary of the 4 different settings, providing a shortening notation along with the associated number of hyper-parameters (or sets of hyper-parameters in the case of θ_i and γ_k) that are required to be learnt in practice.

4.3 Inference

Although a fully Bayesian point-of-view could be taken on the learning procedure by defining prior distributions of the hyper-parameters and directly use an MCMC algorithm (Rasmussen and Williams, 2006; Yang et al., 2016) for approximate inference on the posteriors, this approach remains computationally challenging in practice. Conversely, variational methods have proved to be highly efficient to conduct inference in tricky GP problems (Titsias, 2009; Hensman et al., 2013) and may apply in our context as well. By introducing an adequate independence assumption, we are able to derive a variational formulation leading to analytical approximations for the true hyper-posterior distributions of the latent variables. Then, these hyper-posterior updates allow the computation of a lower bound of the true log-likelihood, thereby specifying the E step of the VEM algorithm (Attias, 2000) that conducts the overall inference. Alternatively, we can maximise this lower bound with respect to the hyper-parameters in the M step for optimisation purpose, to provide estimates. By iterating on these two steps until convergence (pseudo-code in Algorithm 3), the procedure is proved to reach local optima of the lower bound (Boyd and Vandenberghe, 2004), usually in a few iterations. For the sake of clarity, the shorthand $\mathbf{Z} = \{\mathbf{Z}_i\}_i$ and $\boldsymbol{\mu} = \{\mu_k(\mathbf{t})\}_k$ is used in this section when referring to the corresponding set of latent variables.

4.3.1 Variational EM algorithm

We seek an appropriate and analytical approximation $q(\mathbf{Z}, \boldsymbol{\mu})$ for the exact hyper-posterior distribution $p(\mathbf{Z}, \boldsymbol{\mu} \mid \{\mathbf{y}_i\}_i, \Theta)$. Let us first notice that for any distribution $q(\mathbf{Z}, \boldsymbol{\mu})$, the following decomposition holds for the observed-data log-likelihood:

$$\log p(\{\mathbf{y}_i\}_i \mid \Theta) = \mathrm{KL}\left(q \| p\right) + \mathcal{L}(q; \Theta), \tag{4.2}$$

with:

$$\mathrm{KL}\left(q \| p\right) = \int\int q(\mathbf{Z}, \boldsymbol{\mu}) \log \frac{q(\mathbf{Z}, \boldsymbol{\mu})}{p(\mathbf{Z}, \boldsymbol{\mu} \mid \{\mathbf{y}_i\}_i, \Theta)} \,\mathrm{d}\mathbf{Z}\,\mathrm{d}\boldsymbol{\mu},$$
$$\mathcal{L}(q; \Theta) = -\int\int q(\mathbf{Z}, \boldsymbol{\mu}) \log \frac{q(\mathbf{Z}, \boldsymbol{\mu})}{p(\mathbf{Z}, \boldsymbol{\mu}, \{\mathbf{y}_i\}_i \mid \Theta)} \,\mathrm{d}\mathbf{Z}\,\mathrm{d}\boldsymbol{\mu}.$$

Therefore, we expressed the intractable log-likelihood of the model by introducing the Kullback-Leibler (KL) divergence between the approximation $q(\mathbf{Z}, \boldsymbol{\mu})$ and the corresponding true distribution $p(\mathbf{Z}, \boldsymbol{\mu} \mid \{\mathbf{y}_i\}_i, \Theta)$. The right-hand term $\mathcal{L}(q; \Theta)$ in (4.2) defines a so-called *lower bound* for $\log p(\{\mathbf{y}_i\}_i \mid \Theta)$ since a KL divergence is nonnegative by definition. This lower bound depends both upon the approximate distribution $q(\cdot)$ and the hyper-parameters Θ, while remaining tractable under adequate assumptions. By maximising $\mathcal{L}(q; \Theta)$ alternatively with respect to both quantities, optima for the hyper-parameters shall be reached. To achieve such a procedure, the following factorisation is assumed for the approximated distribution:

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

Colloquially, we could say that the independence property that lacks to compute explicit hyper-posterior distributions is *imposed*. Such a condition restricts the family of distributions from which we choose $q(\cdot)$, and we now seek approximations within this family that are as close as possible to the true hyper-posteriors.

E STEP

In the expectation step (E step) of the VEM algorithm, the lower bound of the marginal likelihood $\mathcal{L}(q;\Theta)$ is maximised with respect to the distribution $q(\cdot)$, considering that initial or previously estimated values for $\hat{\Theta}$ are available. Making use of the factorised form previously assumed, we can derive analytical expressions for the optimal distributions over $q_{\mathbf{Z}}(\mathbf{Z})$ and $q_{\boldsymbol{\mu}}(\boldsymbol{\mu})$. As the computing of each distribution involves taking an expectation with respect to the other one, this suggests an iterative procedure where whether the initialisation or a previous estimation serves in the current optimisation process. Therefore, let us introduce two propositions below respectively detailing the exact derivation of the optimal distributions $\hat{q}_{\mathbf{Z}}(\mathbf{Z})$ and $\hat{q}_{\boldsymbol{\mu}}(\boldsymbol{\mu})$ (all proofs are deferred to the corresponding Section 4.8).

Proposition 4.1. *Assume that the hyper-parameters $\hat{\Theta}$ and the variational distribution $\hat{q}_{\boldsymbol{\mu}}(\boldsymbol{\mu}) = \prod_{k=1}^{K} \mathcal{N}\left(\mu_k(\mathbf{t}); \hat{m}_k(\mathbf{t}), \widehat{\mathbf{C}}_k^{\mathbf{t}}\right)$ are known. The optimal variational approximation $\hat{q}_{\mathbf{Z}}(\mathbf{Z})$ of the true hyper-posterior $p\left(\mathbf{Z} \mid \{\mathbf{y}_i\}_i, \hat{\Theta}\right)$ factorises as a product of multinomial distributions:*

$$\hat{q}_{\mathbf{Z}}(\mathbf{Z}) = \prod_{i=1}^{M} \mathcal{M}\left(\mathbf{Z}_i; 1, \boldsymbol{\tau}_i = (\tau_{i1}, \ldots, \tau_{iN})^{\intercal}\right), \tag{4.3}$$

where:

$$\tau_{ik} = \frac{\hat{\pi}_k \, \mathcal{N}\left(\mathbf{y}_i; \hat{m}_k(\mathbf{t}_i), \boldsymbol{\Psi}^{\mathbf{t}_i}_{\hat{\theta}_i, \hat{\sigma}_i^2}\right) \exp\left(-\frac{1}{2} tr\left({\boldsymbol{\Psi}^{\mathbf{t}_i}_{\hat{\theta}_i, \hat{\sigma}_i^2}}^{-1} \widehat{\mathbf{C}}_k^{\mathbf{t}_i}\right)\right)}{\sum_{l=1}^{K} \hat{\pi}_l \, \mathcal{N}\left(\mathbf{y}_i; \hat{m}_l(\mathbf{t}_i), \boldsymbol{\Psi}^{\mathbf{t}_i}_{\hat{\theta}_i, \hat{\sigma}_i^2}\right) \exp\left(-\frac{1}{2} tr\left({\boldsymbol{\Psi}^{\mathbf{t}_i}_{\hat{\theta}_i, \hat{\sigma}_i^2}}^{-1} \widehat{\mathbf{C}}_l^{\mathbf{t}_i}\right)\right)}, \; \forall i \in \mathcal{I}, \forall k \in \mathcal{K}. \tag{4.4}$$

Proposition 4.2. *Assume that the hyper-parameters $\hat{\Theta}$ and the variational distribution $\hat{q}_{\mathbf{Z}}(\mathbf{Z}) = \prod_{i=1}^{M} \mathcal{M}\left(\mathbf{Z}_i; 1, \boldsymbol{\tau}_i\right)$ are known. The optimal variational approximation $\hat{q}_{\boldsymbol{\mu}}(\boldsymbol{\mu})$ of the true hyper-posterior $p\left(\boldsymbol{\mu} \mid \{\mathbf{y}_i\}_i, \hat{\Theta}\right)$ factorises as a product of multivariate Gaussian distributions:*

$$\hat{q}_{\boldsymbol{\mu}}(\boldsymbol{\mu}) = \prod_{k=1}^{K} \mathcal{N}\left(\mu_k(\mathbf{t}); \hat{m}_k(\mathbf{t}), \widehat{\mathbf{C}}_k^{\mathbf{t}}\right), \tag{4.5}$$

with:

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

where the following shorthand notation is used:

- $\tilde{\mathbf{y}}_i = \left(\mathbb{1}_{[t \in \mathbf{t}_i]} \times y_i(t)\right)_{t \in \mathbf{t}}$ *(N-dimensional vector),*
- $\tilde{\boldsymbol{\Psi}}_i = \left[\mathbb{1}_{[t,t' \in \mathbf{t}_i]} \times \psi_{\hat{\theta}_i, \hat{\sigma}_i^2}(t, t')\right]_{t,t' \in \mathbf{t}}$ *($N \times N$ matrix).*

Notice that the *forced* factorisation we assumed between $\mathbf{Z}$ and $\boldsymbol{\mu}$ for approximation purpose additionally offers an induced independence between individuals as indicated by the factorisation in (4.3), and between clusters (see (4.5)).

M STEP

At this point, we have fixed an estimation for $q(\cdot)$ in the lower bound that shall serve to handle the maximisation of $\mathcal{L}(\hat{q}, \Theta)$ with respect to the hyper-parameters. This maximisation step (M step) depends on the initial assumptions on the generative model (Table 4.1), resulting in four different versions for the VEM algorithm (the E step is common to all of them, the branching point is here).

Proposition 4.3. *Assume the variational distributions* $\hat{q}_{\mathbf{Z}}(\mathbf{Z}) = \prod_{i=1}^{M} \mathcal{M}(\mathbf{Z}_i; 1, \boldsymbol{\tau}_i)$ *and* $\hat{q}_{\boldsymbol{\mu}}(\boldsymbol{\mu}) = \prod_{k=1}^{K} \mathcal{N}\left(\mu_k(\mathbf{t}); \hat{m}_k(\mathbf{t}), \widehat{\mathbf{C}}_k^{\mathbf{t}}\right)$ *to be known. For a set of hyper-parameters* $\Theta = \{\{\gamma_k\}_k, \{\theta_i\}_i, \{\sigma_i^2\}_i, \boldsymbol{\pi}\}$, *the optimal values are given by:*

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

where $\mathbb{E}_{\{\mathbf{Z},\boldsymbol{\mu}\}}$ *indicates an expectation taken with respect to* $\hat{q}_{\boldsymbol{\mu}}(\boldsymbol{\mu})$ *and* $\hat{q}_{\mathbf{Z}}(\mathbf{Z})$. *In particular, optimal values for* $\boldsymbol{\pi}$ *can be computed explicitly with:*

$$\hat{\pi}_k = \frac{1}{M} \sum_{i=1}^{M} \boldsymbol{\tau}_{ik}, \ \forall k \in \mathcal{K}.$$

The remaining hyper-parameters are estimated by solving the following maximisation problems, according to the situation. Let us note:

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

Then, for hypothesis $\mathcal{H}_{ki}$*:*

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

For hypothesis $\mathcal{H}_{k0}$*:*

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

For hypothesis $\mathcal{H}_{0i}$*:*

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

For hypothesis $\mathcal{H}_{00}$:

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

Let us stress that, for each sub-case, explicit gradients are available for the functions to maximise, facilitating the optimisation process with gradient-based methods (Hestenes and Stiefel, 1952; Bengio, 2000). The current version of our code implements those gradients and makes use of them within the L-BFGS-B algorithm (Nocedal, 1980; Morales and Nocedal, 2011) devoted to the numerical maximisation. As previously discussed, the hypothesis $\mathcal{H}_{ki}$ necessitates to learn $M+K$ sets of hyper-parameters. However, we notice in Proposition 4.3 that the factorised forms defined as the sum of a Gaussian log-likelihoods and trace terms offer a way to operate the maximisations in parallel on simple functions. Conversely, for the hypothesis $\mathcal{H}_{00}$, only 2 sets of hyper-parameters need to be optimised, namely γ_0, and $\{\theta_0, \sigma_0^2\}$. The small number of functions to maximise is explained by the fact that they are defined as larger sums over all individuals (respectively all clusters). Moreover, this context highlights a multi-task pattern in covariance structures, since each individual (respectively cluster) contributes to the learning of shared hyper-parameters. In practice, $\mathcal{H}_{00}$ is far easier to manage, and we generally reach robust optima in a few iterations. On the contrary, the settings with many hyper-parameters to learn, using mechanically less data for each, may lead more often to computational burden or pathological results. The remaining hypotheses, $\mathcal{H}_{0i}$ and $\mathcal{H}_{k0}$, are somehow middle ground situations between the two extremes and might be used as a compromise according to the problem being dealt with.

4.3.2 Initialisation

Let us discuss here some modelling choices about the initialisation of some quantities involved in the VEM algorithm:

- $\{m_k(\cdot)\}_k$; the mean functions from the hyper-prior distributions of the associated mean processes $\{\mu_k(\cdot)\}_k$. As it may be difficult to pre-specify meaningful values in the absence of external or expert knowledge, these values are often assumed to be 0. However, it remains possible to integrate information in the model by this mean. However, as exhibited in Proposition 4.2, the influence of $\{m_k(\cdot)\}_k$ in hyper-posterior computations decreases rapidly when M grows in a multi-task framework.
- $\{\gamma_k\}_k$, $\{\theta_i\}_i$ and $\{\sigma_i^2\}_i$; the kernel hyper-parameters. We already discussed that the form itself of kernels has to be chosen as well, but once set, we would advise initiating $\{\gamma_k\}_k$ and $\{\theta_i\}_i$ with close and reasonable values whenever possible. As previously noticed in Chapter 3, nearly singular covariance matrices and numerical instability may occur for pathological initialisations, in particular for the hypotheses, like $\mathcal{H}_{ki}$, with many hyper-parameters to learn. This behaviour frequently occurs in the GP framework, and one way to handle this issue is to add a so-called *jitter* term (Bernardo et al., 1998) on the diagonal of the ill-defined covariance matrices.
- $\{\tau_{ik}\}_{ik}$; the estimated individual membership probabilities (or $\boldsymbol{\pi}$; the prior vector of clusters' proportions). Both quantities are valid initialisation depending on whether we start the VEM iterations by an E step or an M step. If we only want to set

the initial proportions of each cluster in the absence of additional information, we may merely specify $\boldsymbol{\pi}$ and start with an E step. Otherwise, if we insert the results from a previous clustering algorithm as an initialisation, the probabilities τ_{ik} for each individual and cluster can be fully specified before proceeding to an M step (or to the $\hat{q}_{\boldsymbol{\mu}}(\boldsymbol{\mu})$'s computing and then the M step).

Let us finally stress that the convergence (to local maxima) of VEM algorithms partly depends on these initialisations. We previously discussed this topic in Section 1.1.3.a and different strategies have been proposed in the literature to manage this issue, among which simulated annealing (Ueda and Nakano, 1998) or repeated short runs (Biernacki et al., 2003).

4.3.3 Pseudocode

The overall algorithm is called MagmaClust (as an extension of the algorithm Magma to cluster-specific mean GPs) and we provide below the pseudo-code summarising the inference procedure. The corresponding R code is available at https://github.com/ArthurLeroy/MAGMAclust.

Algorithm 3 MagmaClust: Variational EM algorithm

Initialise $\{m_k(\mathbf{t})\}_k$, $\Theta = \left\{\{\gamma_k\}_k, \{\theta_i\}_i, \left\{\sigma_i^2\right\}_i\right\}$ and $\{\boldsymbol{\tau}_i^{ini}\}_i$ (or $\boldsymbol{\pi}$).
while not converged **do**
  E step: Optimise $\mathcal{L}(q;\Theta)$ w.r.t. $q(\cdot)$:

$$\hat{q}_{\mathbf{Z}}(\mathbf{Z}) = \prod_{i=1}^{M} \mathcal{M}(\mathbf{Z}_i; 1, \boldsymbol{\tau}_i).$$

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

  M step: Optimise $\mathcal{L}(q;\Theta)$ w.r.t. Θ:

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

end while
return $\hat{\Theta}$, $\{\boldsymbol{\tau}_i\}_i$ $\{\hat{m}_k(\mathbf{t})\}_k$, $\{\widehat{\mathbf{C}}_k^{\mathbf{t}}\}_k$.

4.4 Prediction

At this point, we would consider that the inference on the model is completed, since the training dataset of observed individuals $\{\mathbf{y}_i\}_i$ enabled to estimate the desired hyper-parameters and latent variables' distributions. For the sake of concision, we thus omit the writing of conditionings over $\hat{\Theta}$ in the sequel. Then, let us now assume the partial observation of a new individual, denoted by the index $*$, for whom we collected a few data points $y_*(\mathbf{t}_*)$ at timestamps $\mathbf{t}_*$. Defining a multi-task GPs mixture prediction consists in seeking an analytical distribution $p(y_*(\cdot) \mid y_*(\mathbf{t}_*), \{\mathbf{y}_i\}_i)$, according to the information brought by: its own observations; the training dataset; the cluster structure among individuals. As we aim at studying the output values $y_*(\cdot)$ at arbitrarily chosen timestamps, say $\mathbf{t}^p$ (the index p stands for *prediction*), a new notation for the pooled vector of timestamps $\mathbf{t}_*^p = \begin{bmatrix} \mathbf{t}^p \\ \mathbf{t}_* \end{bmatrix}$ is proposed. This vector serves as a working grid on which the different distributions involved in the prediction procedure are evaluated. In the absence of external restrictions, we would

	$\mathbf{t}_*^p = \mathbf{t}$	$\mathbf{t}_*^p \neq \mathbf{t}$
$\mathcal{H}_{00}$	2-3bis-4-5	1-2-3bis-4-5
$\mathcal{H}_{k0}$	2-3bis-4-5	1-2-3bis-4-5
$\mathcal{H}_{0i}$	2-3-4-5	1-2-3-4-5
$\mathcal{H}_{ki}$	2-3-4-5	1-2-3-4-5

Table 4.2 – Summary of the different steps to perform in the prediction procedure, according to the model assumptions and the target grid of timestamps.

strongly advise to include the observed timestamps of all training individuals, $\mathbf{t}$, within $\mathbf{t}_*^p$, since evaluating the processes at these locations allows for sharing information across tasks. Otherwise, any data points defined on timestamps outside of the working grid would be discarded from the multi-task aspect of the model. In particular, if $\mathbf{t}_*^p = \mathbf{t}$, we may even use directly the variational distribution $q_{\boldsymbol{\mu}}(\boldsymbol{\mu})$ computed in the VEM algorithm, and thus skip one step of the prediction procedure that is described below. Throughout the section, we aim at defining a probabilistic prediction for this new individual, accounting for the information of all training data $\{\mathbf{y}_i\}_i$. To this end, we manipulate several distributions of the type $p(\cdot \mid \{\mathbf{y}_i\}_i)$ and refer to them with the adjective *multi-task*. Additionally to highlighting the information-sharing aspect, this term allows us to distinguish the role of $\{\mathbf{y}_i\}_i$ from the one of the newly observed data $y_*(\mathbf{t}_*)$, which are now the reference data for establishing if a distribution is called a *prior* or a *posterior*. Deriving a predictive distribution in our multi-task GP framework requires to complete the following steps.

1. Compute the hyper-posterior approximation of $\{\mu_k(\cdot)\}_k$ at $\mathbf{t}_*^p$: $\hat{q}_\mu(\{\mu_k(\mathbf{t}_*^p)\}_k)$,
2. Deduce the multi-task prior distribution: $p(y_*(\mathbf{t}_*^p) \mid \mathbf{Z}_*, \{\mathbf{y}_i\}_i)$,
3. Compute the new hyper-parameters $\{\theta_*, \sigma_*^2\}$ and $p(\mathbf{Z}_* \mid y_*(\mathbf{t}_*), \{\mathbf{y}_i\}_i)$ via an EM,

3bis. Assign $\theta_* = \theta_0$, $\sigma_*^2 = \sigma_0^2$ and compute directly $p(\mathbf{Z}_* \mid y_*(\mathbf{t}_*), \{\mathbf{y}_i\}_i)$,

4. Compute the multi-task posterior distribution: $p(y_*(\mathbf{t}^p) \mid y_*(\mathbf{t}_*), \mathbf{Z}_*, \{\mathbf{y}_i\}_i)$,
5. Deduce the multi-task GPs mixture prediction: $p(y_*(\mathbf{t}^p) \mid y_*(\mathbf{t}_*), \{\mathbf{y}_i\}_i)$.

We already discussed the influence of the initial modelling hypotheses on the overall procedure. Hence, let us display in Table 4.2 a quick reminder helping to keep track of which steps need to be performed in each context.

4.4.1 Posterior inference on the mean processes

In order to integrate the information contained in the shared mean processes, we first need to re-compute the variational approximation of $\{\mu_k(\cdot)\}_k$'s hyper-posterior on the new $\tilde{N}$-dimensional working grid $\mathbf{t}_*^p$. By using once more Proposition 4.2, it appears straightforward to derive this quantity that still factorises as a product of Gaussian distributions where we merely substitute the values of timestamps:

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

with:

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

where the following shorthand notation is used:

- $\tilde{\mathbf{y}}_i = \left(\mathbb{1}_{[t \in \mathbf{t}_i]} \times y_i(t) \right)_{t \in \mathbf{t}_*^p}$ ($\tilde{N}$-dimensional vector),
- $\tilde{\mathbf{\Psi}}_i = \left[\mathbb{1}_{[t,t' \in \mathbf{t}_i]} \times \psi_{\hat{\theta}_i, \hat{\sigma}_i^2}(t,t') \right]_{t,t' \in \mathbf{t}_*^p}$ ($\tilde{N} \times \tilde{N}$ matrix).

Let us acknowledge that the subsequent analytical developments party rely on this variational approximate distribution $\hat{q}_{\boldsymbol{\mu}}(\{\mu_k(\mathbf{t}_*^p)\}_k)$, and may thus be considered, in a sense, as approximated as well. However, this quantity provides a valuable closed-form expression that we substitute to the true hyper-posterior in Proposition 4.4 below, while keeping the signs $=$ instead of $\approx$ for clarity.

4.4.2 Computation of the multi-task prior distributions

For a sake of completeness, let us recall the equivalence between two ways of writing conditional distributions that are used in the subsequent results:

$$p(\cdot \mid \mathbf{Z}_*) = \prod_{k=1}^{K} p(\cdot \mid Z_{*k} = 1)^{Z_{*k}}.$$

We may regularly substitute one to the other in the sequel depending on the handier in the context. Once the mean processes' distributions are re-computed on the working grid, their underlying influence shall be directly plugged into a marginalised multi-task prior over $y_*(\mathbf{t}_*^p)$ by integrating out the $\{\mu_k(\mathbf{t}_*^p)\}_k$. As the mean processes vanish, the new individual's outputs $y_*(\mathbf{t}_*^p)$ directly depends upon the training dataset $\{\mathbf{y}_i\}_i$, as highlighted in the proposition below.

Proposition 4.4. *For a set of timestamps $\mathbf{t}_*^p$, the multi-task prior distribution of y_* knowing its clustering latent variable is given by:*

$$p(y_*(\mathbf{t}_*^p) \mid \mathbf{Z}_*, \{\mathbf{y}_i\}_i) = \prod_{k=1}^{K} \mathcal{N}\left(y_*(\mathbf{t}_*^p); \hat{m}_k(\mathbf{t}_*^p), \widehat{\mathbf{C}}_k^{\mathbf{t}_*^p} + \Psi_{\theta_*, \sigma_*^2}^{\mathbf{t}_*^p} \right)^{Z_{*k}}. \tag{4.6}$$

Proof. Let us recall that, conditionally to their mean process, the individuals are independent of one another. Then, for all $k \in \mathcal{K}$, we have:

$$\begin{aligned}
p(y_*(\mathbf{t}_*^p) \mid Z_{*k} = 1, \{\mathbf{y}_i\}_i) &= \int p(y_*(\mathbf{t}_*^p), \mu_k(\mathbf{t}_*^p) \mid Z_{*k} = 1, \{\mathbf{y}_i\}_i) \, \mathrm{d}\mu_k(\mathbf{t}_*^p) \\
&= \int p(y_*(\mathbf{t}_*^p) \mid \mu_k(\mathbf{t}_*^p), Z_{*k} = 1) \underbrace{p(\mu_k(\mathbf{t}_*^p) \mid Z_{*k} = 1, \{\mathbf{y}_i\}_i)}_{\approx q_{\boldsymbol{\mu}}(\mu_k(\mathbf{t}_*^p))} \mathrm{d}\mu_k(\mathbf{t}_*^p) \\
&= \int \mathcal{N}\left(y_*(\mathbf{t}_*^p); \mu_k(\mathbf{t}_*^p), \Psi_{\theta_*, \sigma_*^2}^{\mathbf{t}_*^p} \right) \mathcal{N}\left(\mu_k(\mathbf{t}_*^p); \hat{m}_k(\mathbf{t}_*^p), \widehat{\mathbf{C}}_k^{\mathbf{t}_*^p} \right) \mathrm{d}\mu_k(\mathbf{t}_*^p) \\
&= \mathcal{N}\left(y_*(\mathbf{t}_*^p); \hat{m}_k(\mathbf{t}_*^p), \widehat{\mathbf{C}}_k^{\mathbf{t}_*^p} + \Psi_{\theta_*, \sigma_*^2}^{\mathbf{t}_*^p} \right).
\end{aligned}$$

The final line is obtained by remarking that such a convolution of Gaussian distributions remains Gaussian as well (Bishop, 2006, Chapter 2), and we refer to Section 3.8.1 for the detailed calculus in this exact context. Therefore, we finally get:

$$
\begin{aligned}
p(y_*(\mathbf{t}_*^p) \mid \mathbf{Z}_*, \{\mathbf{y}_i\}_i) &= \prod_{k=1}^{K} p(y_*(\mathbf{t}_*^p) \mid Z_{*k} = 1, \{\mathbf{y}_i\}_i)^{Z_{*k}} \\
&= \prod_{k=1}^{K} \mathcal{N}\left(y_*(\mathbf{t}_*^p); \hat{m}_k(\mathbf{t}_*^p), \widehat{\mathbf{C}}_k^{\mathbf{t}_*^p} + \Psi_{\theta_*,\sigma_*^2}^{\mathbf{t}_*^p}\right)^{Z_{*k}}.
\end{aligned}
$$

□

4.4.3 Optimisation of the new hyper-parameters and computation of the clusters' probabilities

Now that the mean processes have been removed at the previous step, this section strongly resembles the classical learning procedure through an EM algorithm for a Gaussian mixture model. In our case, it allows us both to estimate the hyper-parameters of the new individual $\{\theta_*, \sigma_*\}$ and to compute the hyper-posterior distribution of its latent clustering variable $\mathbf{Z}_*$, which provides the associated clusters' membership probabilities $\boldsymbol{\tau}_*$. As before, E steps and M steps are alternatively processed until convergence, but this time by working with exact formulations instead of variational ones.

E STEP

In the E step, hyper-parameters estimates are assumed to be known. Recalling that the latent clustering variable $\mathbf{Z}_*$ is independent from the training data $\{\mathbf{y}_i\}_i$, the multi-task hyper-posterior distribution maintains an explicit derivation:

$$
\begin{aligned}
p(\mathbf{Z}_* \mid y_*(\mathbf{t}_*), \{\mathbf{y}_i\}_i, \hat{\theta}_*, \hat{\sigma}_*^2, \hat{\boldsymbol{\pi}}) &\propto p(y_*(\mathbf{t}_*) \mid \mathbf{Z}_*, \{\mathbf{y}_i\}_i, \hat{\theta}_*, \hat{\sigma}_*^2) p(\mathbf{Z}_* \mid \hat{\boldsymbol{\pi}}) \\
&\propto \prod_{k=1}^{K} \left\{ \mathcal{N}\left(y_*(\mathbf{t}_*); \hat{m}_k(\mathbf{t}_*), \widehat{\mathbf{C}}_k^{\mathbf{t}_*} + \Psi_{\hat{\theta}_*,\hat{\sigma}_*^2}^{\mathbf{t}_*}\right)^{Z_{*k}} \right\} \prod_{l=1}^{K} \hat{\pi}_l^{Z_{*l}} \\
&\propto \prod_{k=1}^{K} \left(\hat{\pi}_k \mathcal{N}\left(y_*(\mathbf{t}_*); \hat{m}_k(\mathbf{t}_*), \widehat{\mathbf{C}}_k^{\mathbf{t}_*} + \Psi_{\hat{\theta}_*,\hat{\sigma}_*^2}^{\mathbf{t}_*}\right) \right)^{Z_{*k}}.
\end{aligned}
$$

By inspection, we recognise the form of a multinomial distribution and thus retrieve the corresponding normalisation constant to deduce:

$$
p(\mathbf{Z}_* \mid y_*(\mathbf{t}_*), \{\mathbf{y}_i\}_i, \hat{\theta}_*, \hat{\sigma}_*^2, \hat{\boldsymbol{\pi}}) = \mathcal{M}\left(\mathbf{Z}_*; 1, \boldsymbol{\tau}_* = (\tau_{*1}, \ldots, \tau_{*K})^\intercal\right), \tag{4.7}
$$

with:

$$
\tau_{*k} = \frac{\hat{\pi}_k \mathcal{N}\left(y_*(\mathbf{t}_*); \hat{m}_k(\mathbf{t}_*), \widehat{\mathbf{C}}_k^{\mathbf{t}_*} + \Psi_{\hat{\theta}_*,\hat{\sigma}_*^2}^{\mathbf{t}_*}\right)}{\sum_{l=1}^{K} \hat{\pi}_l \mathcal{N}\left(y_*(\mathbf{t}_*); \hat{m}_l(\mathbf{t}_*), \widehat{\mathbf{C}}_l^{\mathbf{t}_*} + \Psi_{\hat{\theta}_*,\hat{\sigma}_*^2}^{\mathbf{t}_*}\right)}, \ \forall k \in \mathcal{K}. \tag{4.8}
$$

M STEP

Assuming to know the value of $\boldsymbol{\tau}_*$, we may derive optimal values for the hyper-parameters of the new individual through the following maximisation:

$$\{\hat{\theta}_*, \hat{\sigma}_*^2\} = \underset{\theta_*, \sigma_*}{\operatorname{argmax}} \mathbb{E}_{\mathbf{Z}_*} \left[\log p(y_*(\mathbf{t}_*), \mathbf{Z}_* \mid \{\mathbf{y}_i\}_i, \theta_*, \sigma_*, \hat{\boldsymbol{\pi}}) \right].$$

Let us note $\mathcal{L}_*(\theta_*, \sigma_*) = \log p(y_*(\mathbf{t}_*), \mathbf{Z}_* \mid \{\mathbf{y}_i\}_i, \theta_*, \sigma_*, \hat{\boldsymbol{\pi}})$. By remarking that $\hat{\boldsymbol{\pi}}$ has already been estimated previously, we may easily derive the expression to maximise with respect to θ_* and σ_* in practice:

$$\begin{aligned}
\mathbb{E}_{\mathbf{Z}_*} \left[\mathcal{L}_*(\theta_*, \sigma_*) \right] &= \mathbb{E}_{\mathbf{Z}_*} \left[\log p(y_*(\mathbf{t}_*), \mathbf{Z}_* \mid \{\mathbf{y}_i\}_i, \theta_*, \sigma_*, \hat{\boldsymbol{\pi}} \right]) \\
&= \mathbb{E}_{\mathbf{Z}_*} \left[\log p(y_*(\mathbf{t}_*) \mid \mathbf{Z}_*, \{\mathbf{y}_i\}_i, \theta_*, \sigma_*) + \log p(\mathbf{Z}_* \mid \hat{\boldsymbol{\pi}}) \right] \\
&= \mathbb{E}_{\mathbf{Z}_*} \left[\log \prod_{k=1}^{K} \mathcal{N}\left(y_*(\mathbf{t}_*); \hat{m}_k(\mathbf{t}_*), \widehat{\mathbf{C}}_k^{\mathbf{t}_*} + \Psi_{\theta_*, \sigma_*^2}^{\mathbf{t}_*} \right)^{Z_{*k}} \right] + C_1 \\
&= \sum_{k=1}^{K} \mathbb{E}_{\mathbf{Z}_*} \left[Z_{*k} \right] \log \mathcal{N}\left(y_*(\mathbf{t}_*); \hat{m}_k(\mathbf{t}_*), \widehat{\mathbf{C}}_k^{\mathbf{t}_*} + \Psi_{\theta_*, \sigma_*^2}^{\mathbf{t}_*} \right) + C_1 \\
&= \sum_{k=1}^{K} \tau_{*k} \log \mathcal{N}\left(y_*(\mathbf{t}_*); \hat{m}_k(\mathbf{t}_*), \widehat{\mathbf{C}}_k^{\mathbf{t}_*} + \Psi_{\theta_*, \sigma_*^2}^{\mathbf{t}_*} \right) + C_1,
\end{aligned}$$

where C_1 is a constant term. Thus, the optimisation in this case merely relies on the maximisation of a weighted sum of Gaussian log-likelihoods, for which gradients are well-known.

3BIS.

In the case where the hyper-parameters are supposed to be common across individuals ($\mathcal{H}_{00}$ or $\mathcal{H}_{k0}$), there is no need to additional optimisation since we already have $\hat{\theta}_* = \hat{\theta}_0$ and $\hat{\sigma}_*^2 = \hat{\sigma}_0$ by definition. However, the probabilities of lying in each cluster $\boldsymbol{\tau}_*$ for the new individual still need to be computed, which shall be handled by directly using the expression (4.8) from the E step.

3TER.

Conversely, let us note that even if hyper-parameters for each individual are supposed to be different ($\mathcal{H}_{0i}$ or $\mathcal{H}_{ki}$), it remains possible to avoid the implementation of an EM algorithm by stating $\boldsymbol{\tau}_* = \hat{\boldsymbol{\pi}}$. Such an assumption intuitively expresses that we would guess the membership probabilities of each cluster from the previously estimated mixing proportions, without taking new individual's observations into account. Although we would not recommend this choice for getting optimal results, it still seems to be worth a mention for applications with a compelling need to avoid EM's extra computations during the prediction process.

4.4.4 Computation of the multi-task posterior distributions

Once the needed hyper-parameters have been estimated and the prior distribution established, the classical formula for GP predictions can be applied to the new individual, for each possible latent cluster. First, let us recall the prior distribution by separating observed from target timestamps, and introducing a shorthand notation for the covariance:

$$p(y_*(\mathbf{t}_*^p) \mid Z_{*k} = 1, \{\mathbf{y}_i\}_i) = \mathcal{N}\left(\begin{bmatrix} y_*(\mathbf{t}^p) \\ y_*(\mathbf{t}_*) \end{bmatrix}; \begin{bmatrix} \hat{m}_k(\mathbf{t}^p) \\ \hat{m}_k(\mathbf{t}_*) \end{bmatrix}, \begin{pmatrix} \mathbf{\Gamma}_k^{\mathbf{t}^p\mathbf{t}^p} & \mathbf{\Gamma}_k^{\mathbf{t}^p\mathbf{t}_*} \\ \mathbf{\Gamma}_k^{\mathbf{t}_*\mathbf{t}^p} & \mathbf{\Gamma}_k^{\mathbf{t}_*\mathbf{t}_*} \end{pmatrix}\right), \ \forall k \in \mathcal{K},$$

where $\mathbf{\Gamma}_k^{\mathbf{t}^p,\mathbf{t}^p} = \widehat{\mathbf{C}}_k^{\mathbf{t}^p} + \Psi_{\theta_*,\sigma_*^2}^{\mathbf{t}^p}$ and likewise for the other blocks of the matrices. Therefore, recalling that conditioning on the sub-vector of observed values $y_*(\mathbf{t}_*)$ maintains a Gaussian distribution (Bishop, 2006; Rasmussen and Williams, 2006), we can derive the multi-task posterior distribution for each latent cluster:

$$p(y_*(\mathbf{t}^p) \mid Z_{*k} = 1, y_*(\mathbf{t}_*), \{\mathbf{y}_i\}_i) = \mathcal{N}\left(y_*(\mathbf{t}^p); \hat{\mu}_{*k}(\mathbf{t}^p), \widehat{\mathbf{\Gamma}}_{*k}^{\mathbf{t}^p}\right), \ \forall k \in \mathcal{K}, \tag{4.9}$$

where:

- $\hat{\mu}_{*k}(\mathbf{t}^p) = \hat{m}_k(\mathbf{t}^p) + \mathbf{\Gamma}_k^{\mathbf{t}^p\mathbf{t}_*}{\mathbf{\Gamma}_k^{\mathbf{t}_*\mathbf{t}_*}}^{-1}\left(y_*(\mathbf{t}_*) - \hat{m}_k(\mathbf{t}_*)\right), \ \forall k \in \mathcal{K},$
- $\widehat{\mathbf{\Gamma}}_{*k}^{\mathbf{t}^p} = \mathbf{\Gamma}_k^{\mathbf{t}^p\mathbf{t}^p} - \mathbf{\Gamma}_k^{\mathbf{t}^p\mathbf{t}_*}{\mathbf{\Gamma}_k^{\mathbf{t}_*\mathbf{t}_*}}^{-1}\mathbf{\Gamma}_k^{\mathbf{t}_*\mathbf{t}^p}, \ \forall k \in \mathcal{K}.$

4.4.5 Computation of the multi-task GPs mixture prediction

To conclude, by summing over all possible combinations for the latent clustering variable $\mathbf{Z}_*$, we can derive the final predictive distribution.

Proposition 4.5. *The multi-task GPs mixture posterior distribution for $y_*(\mathbf{t}^p)$ takes the form below:*

$$p(y_*(\mathbf{t}^p) \mid y_*(\mathbf{t}_*), \{\mathbf{y}_i\}_i) = \sum_{k=1}^{K} \tau_{*k} \ \mathcal{N}\left(y_*(\mathbf{t}^p); \hat{\mu}_{*k}(\mathbf{t}^p), \widehat{\mathbf{\Gamma}}_{*k}^{\mathbf{t}^p}\right).$$

Proof. Taking advantage of (4.9) and the multi-task hyper-posterior distribution of $\mathbf{Z}_*$ as computed in (4.7), it is straightforward to integrate out the latent clustering variable:

$$\begin{aligned}
p(y_*(\mathbf{t}^p) \mid y_*(\mathbf{t}_*), \{\mathbf{y}_i\}_i) &= \sum_{\mathbf{Z}_*} p(y_*(\mathbf{t}^p), \mathbf{Z}_* \mid y_*(\mathbf{t}_*), \{\mathbf{y}_i\}_i) \\
&= \sum_{\mathbf{Z}_*} p(y_*(\mathbf{t}^p) \mid \mathbf{Z}_*, y_*(\mathbf{t}_*), \{\mathbf{y}_i\}_i) p(\mathbf{Z}_* \mid y_*(\mathbf{t}_*), \{\mathbf{y}_i\}_i) \\
&= \sum_{\mathbf{Z}_*} \prod_{k=1}^{K} \left(\tau_{*k} \ p(y_*(\mathbf{t}^p) \mid Z_{*k} = 1, y_*(\mathbf{t}_*), \{\mathbf{y}_i\}_i)\right)^{Z_{*k}} \\
&= \sum_{\mathbf{Z}_*} \prod_{k=1}^{K} \left(\tau_{*k} \ \mathcal{N}\left(y_*(\mathbf{t}^p); \hat{\mu}_{*k}(\mathbf{t}^p), \widehat{\mathbf{\Gamma}}_{*k}^{\mathbf{t}^p}\right)\right)^{Z_{*k}} \\
&= \sum_{k=1}^{K} \tau_{*k} \ \mathcal{N}\left(y_*(\mathbf{t}^p); \hat{\mu}_{*k}(\mathbf{t}^p), \widehat{\mathbf{\Gamma}}_{*k}^{\mathbf{t}^p}\right),
\end{aligned}$$

where we recall for the transition to the last line that $Z_{*k} = 1$ if the $*$-th individual belongs to the k-th cluster and $Z_{*k} = 0$ otherwise. Hence, summing a product with only one non-zero exponent over all possible combination for $\mathbf{Z}_*$ is equivalent to merely sum over the values of k, and the variable Z_{*k} simply vanishes. □

Alternative predictions

Even though Proposition 4.5 provides an elegant probabilistic prediction in terms of GPs mixture, it remains important to notice that this quantity is no longer a Gaussian distribution. In particular, the distribution of an output value at any point-wise evaluation is expected to differ significantly from a classical Gaussian variable, by being multi-modal for instance. This property is especially true for individuals with high uncertainty about the clusters they probably belong to, whereas the distribution would be close to the Gaussian when $\tau_{*k} \approx 1$ for one cluster and almost zero for the others. While we believe that such a GPs mixture distribution highlights the uncertainty resulting from a possible cluster structure in data and offers a rather original view on the matter of GP predictions, some applications may suffer from this non-Gaussian final distribution. Fortunately, it remains pretty straightforward to proceed to a simplification of the clustering inference by assuming that the $*$-individual only belongs to its more probable cluster, which is equivalent to postulate $max\{\tau_{*k}\}_k = 1$ and the others to be zero. In this case, the final Gaussian mixture turns back into a Gaussian distribution, and we retrieve a uni-modal prediction, easily displayed by its mean along with credible intervals.

4.5 Complexity analysis for training and prediction

It is customary to stress that computational complexity is of paramount importance in GP models as a consequence of their usual cubic (resp. quadratic) cost in the number of data points for learning (resp. prediction). In the case of MagmaClust, we use information from M individuals scattered into K clusters, each of them providing N_i observations, and those quantities mainly specify the overall complexity of the algorithm. Moreover, N refers to the number of distinct timestamps (i.e. $N \leq \sum_{i=1}^{M} N_i$) in the training dataset and corresponds to the dimension of the objects involved in the kernel-specific mean processes computations. Typically, the learning complexity would be proportional to one iteration of the VEM algorithm, which requires $\mathcal{O}\left(M \times N_i^3 + K \times N^3\right)$ operations.

As previously discussed, the hypotheses formulated on the hyper-parameters would influence the constant of this complexity but generally not in more than an order of magnitude. For instance, the models under the assumption $\mathcal{H}_{00}$ usually require less optimisation time in practice, although it does not change the number or the dimensions of the covariance matrices to inverse, which mainly control the overall computing time. The dominating terms in this expression depend on the context, regarding the relative values of M, N_i, N and K. In contexts where the number of individuals M dominates, like with small common grids of timestamps for instance, the left-hand term would control the complexity, and clustering's additional cost would be negligible. Conversely, for a relatively low number of individuals or a large size N for the pooled grid of timestamps, the right-hand term becomes the primary burden, and the computing time increases proportionally to the number of clusters compared to the original Magma algorithm.

During the prediction step, the re-computation of $\{\mu_k(\cdot)\}_k$'s variational distributions implies K inversions of covariance matrices with dimensions depending on the size of the prediction grid $\mathbf{t}_*^p$. In practice though, if we fix a fine grid of target timestamps in advance, this operation can be assimilated to the learning step. In this case, the prediction complexity remains at most in the same order as the usual learning for a single-task GP, that is $\mathcal{O}(K \times$

N_*^3) (this corresponds to the estimation of the new individual's hyper-parameters, and would decrease to $\mathcal{O}(K \times N_*^2)$ for $\mathcal{H}_{k0}$ or $\mathcal{H}_{00}$). In many contexts, most of the time-consuming learning steps can be performed in advance, and the immediate prediction cost for each new individual is negligible in comparison (generally comparable to a single-task GP prediction).

4.6 Experiments

The present section is dedicated to the evaluation of MagmaClust on both synthetic and real datasets. The performance of the algorithm is assessed in regards to its clustering and forecast abilities. To this purpose, let us introduce the simulation scheme generating the synthetic data along with the measures used to compare our method to alternatives quantitatively. Throughout, the *exponentiated quadratic* (EQ) kernel (Equation (4.1)) serves as covariance structure for both generating data and modelling. The manipulation of more sophisticated kernels remains a topic beyond the scope of the present chapter, and the EQ proposes a fair common ground for comparison between methods. Thereby, each kernel introduced in the sequel is associated with two hyper-parameters. Namely, $v \in \mathbb{R}^+$ represents a variance term whereas $\ell \in \mathbb{R}^+$ specifies the length-scale. The synthetic datasets are generated following the general procedure below, with minor modifications according to the model assumptions $\mathcal{H}_{00}$, $\mathcal{H}_{k0}$, $\mathcal{H}_{0i}$ or $\mathcal{H}_{ki}$:

1. Define a random working grid $\mathbf{t} \subset [0, 10]$ of $N = 200$ timestamps to study $M = 50$ individuals, scattered into K clusters,
2. Draw the prior mean functions for $\{\mu_k(\cdot)\}_k$: $m_k(t) = at + b,\ \forall t \in \mathbf{t}, \forall k \in \mathcal{K}$, where $a \in [-2, 2]$ and $b \in [20, 30]$,
3. Draw uniformly hyper-parameters for $\{\mu_k(\cdot)\}_k$'s kernels : $\gamma_k = \{v_{\gamma_k}, \ell_{\gamma_k}\},\ \forall k \in \mathcal{K}$, where $v_{\gamma_k} \in \left[1, e^3\right]$ and $\ell_{\gamma_k} \in \left[1, e^1\right]$, (or $\gamma_0 = \{v_{\gamma_0}, \ell_{\gamma_0}\}$),
4. Draw $\mu_k(\mathbf{t}) \sim \mathcal{N}\left(m_k(\mathbf{t}), \mathbf{C}_{\gamma_k}^{\mathbf{t}}\right), \forall k \in \mathcal{K}$,
5. For all $i \in \mathcal{I}$, draw uniformly the hyper-parameters for individual kernels $\theta_i = \{v_{\theta_i}, \ell_{\theta_i}\}$, where $v_{\theta_i} \in \left[1, e^3\right]$, $\ell_{\theta_i} \in \left[1, e^1\right]$, and $\sigma_i^2 \in [0, 0.1]$, (or $\theta_0 = \{v_{\theta_0}, \ell_{\theta_0}\}$ and σ_0^2),
6. Define $\boldsymbol{\pi} = (\frac{1}{K}, \ldots, \frac{1}{K})^\intercal$ and draw $\mathbf{Z}_i \sim \mathcal{M}(1, \boldsymbol{\pi}),\ \forall i \in \mathcal{I}$,
7. For all $i \in \mathcal{I}$ and $Z_{ik} = 1$, draw uniformly a random subset $\mathbf{t}_i \subset \mathbf{t}$ of $N_i = 30$ timestamps, and draw $\mathbf{y}_i \sim \mathcal{N}\left(\mu_k(\mathbf{t}_i), \boldsymbol{\Psi}_{\theta_i, \sigma_i^2}^{\mathbf{t}_i}\right)$.

This procedure offers datasets for both individuals $\{\mathbf{t}_i, \mathbf{y}_i\}_i$ and the underlying mean processes $\{\mathbf{t}, \mu_k(\mathbf{t})\}_k$. In the context of prediction, a new individual is generated according to the same scheme, although its first 20 data points are assumed to be observed while the remaining 10 serve as testing values. While it may be argued that this repartition 20/10 is somehow arbitrary, a more detailed analysis with changing numbers of observed points in Section 3.6.2 revealed a low effect on the global evaluation. Unless otherwise stated, we fix the number of clusters to be $K = 3$ and the model assumption to be $\mathcal{H}_{00}$ for generating the data. The question of the adequate choice of K in clustering applications is a recurrent concern for which we do not provide any specific proposition. Therefore, we assume to know the true number of clusters in the synthetic framework and the real-life application, for

which this number has already been determined in Chapter 2.

Besides, the Adjusted Rand Index (ARI) (Hubert and Arabie, 1985) is used as a measure of adequacy for comparison between the groups obtained through the clustering procedure and the true clusters that generated the data. More specifically, the ARI is defined by counting the proportions of matching pairs between groups, and a value of 1 represents a perfect correspondence. Let us note that ARI still applies when it comes to evaluating clustering partitions with different numbers of clusters. On the matter of prediction, the mean square error (MSE) between predicted means and the true values offers a measure of the average forecast performance. Formally, we define the MSE in prediction on the 10 testing points for the new individual as:

$$\frac{1}{10}\sum_{u=21}^{30}\left(y_*^{pred}(t_*^u) - y_*^{true}(t_*^u)\right)^2.$$

Moreover, an additional measure accounting for the validity of uncertainty quantification is defined in Chapter 3 as the percentage of true data effectively lying within the 95% credible interval (CI_{95}), which is constructed from the predictive distribution. We extend here this measure to the context of GPs mixture, where CI_{95} is no longer available directly (as for any multi-modal distribution). Namely, the weighted CI_{95} coverage ($WCIC_{95}$) is defined to be:

$$100 \times \frac{1}{10}\sum_{u=21}^{30}\sum_{k=1}^{K}\tau_{*k}\ \mathbb{1}_{\{y_*^{true}(t_*^u)\in\ CI_{95}^k\}},$$

where CI_{95}^k represents the CI_{95} computed for the k-th cluster-specific Gaussian predictive distribution (4.9). In the case where $K = 1$, i.e. a simple Gaussian instead of a GPs mixture, the $WCIC_{95}$ reduces to the previously evoked CI_{95} coverage. By averaging the weighted cluster-specific CI_{95}^k coverage, we still obtain an adequate and comparable quantification of the uncertainty relevance for our predictions. By definition, the value of this indicator should be as close as possible to 95%. Finally, the mean functions $\{m_k(\cdot)\}_k$ are set to be 0 in MagmaClust, as usual for GPs, whereas the membership probabilities τ_{ik} are initialised thanks to a preliminary k-means algorithm.

4.6.1 Illustration on synthetic examples

Figure 4.2 provides a comparison on the same dataset between a classical GP regression (top), the multi-task GP algorithm Magma (middle), and the multi-task GPs mixture approach MagmaClust (bottom). On each sub-graph, the plain blue line represents the mean parameter from the predictive distribution, and the grey shaded area covers the CI_{95}. The dashed lines stand for the multi-task prior mean functions $\{\hat{m}_k(\cdot)\}_k$ resulting from the estimation of the mean processes. The points in black are the observations for the new individual $*$, whereas the red points constitute the true target values to forecast. Moreover, the colourful background points depict the data of the training individuals, which we colour according to their true cluster in MagmaClust displays (bottom). As expected, a simple GP regression provides an adequate fit close to the data points before quickly diving to the prior value 0 when lacking information. Conversely, Magma takes advantage of its multi-task component to share knowledge across individuals by estimating a more relevant mean process. However, this unique mean process appears unable to account for the clear group structure, although adequately recovering the dispersion of the data. In the case of

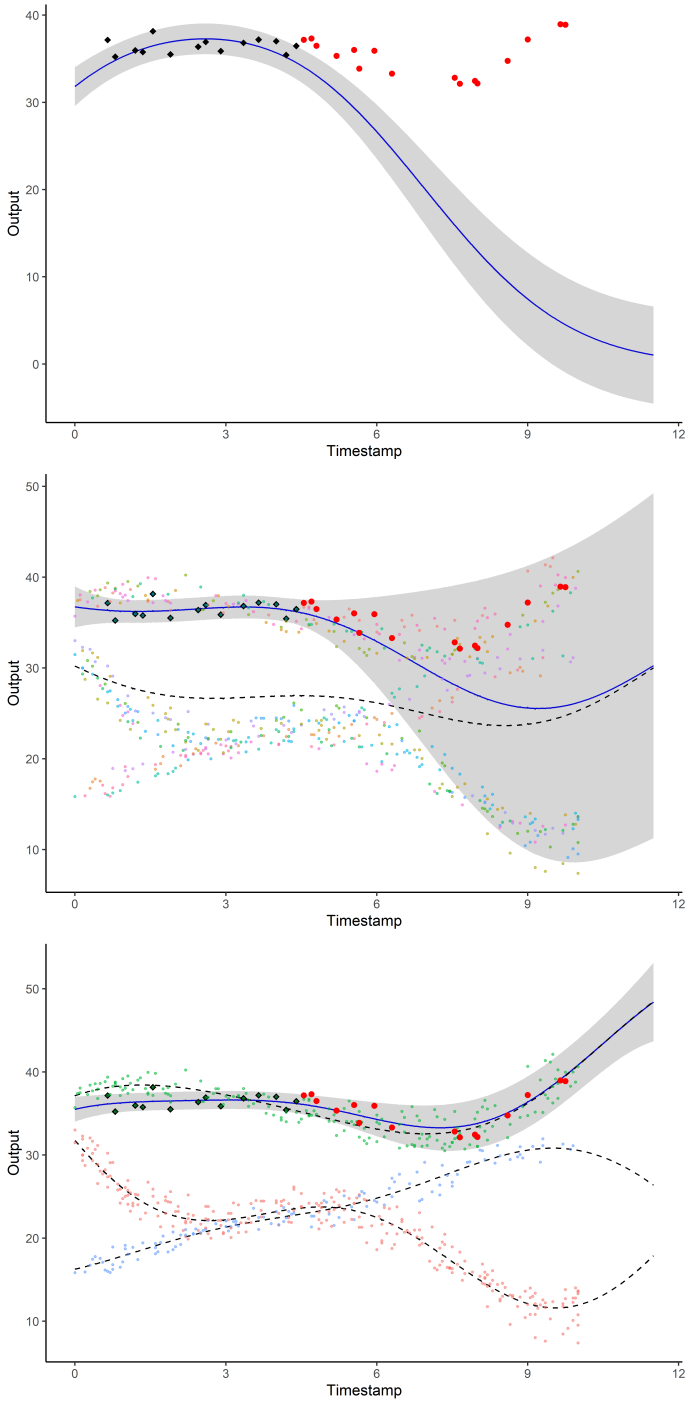

Figure 4.2 – Prediction curves (blue) with associated 95% credible intervals (grey) from GP regression (top), MAGMA (middle) and MAGMACLUST (bottom). The dashed lines represent the mean parameters from the mean processes estimates. Observed data points are in black, testing data points are in red. Backward points are the observations from the training dataset, coloured relatively to individuals (middle) or clusters (bottom).

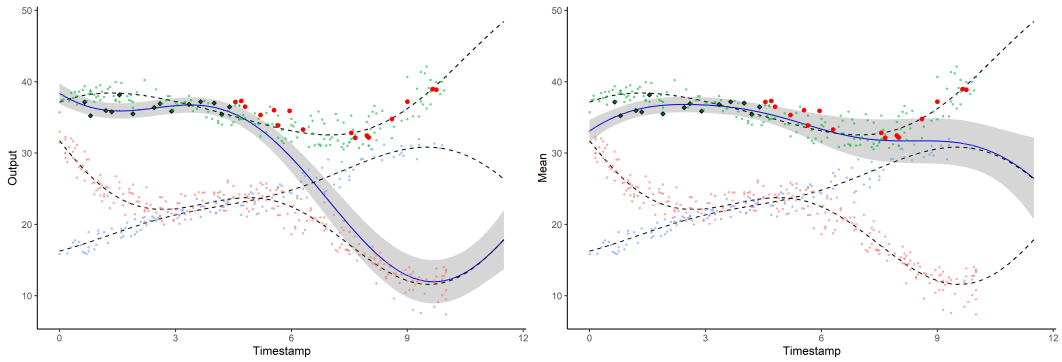

Figure 4.3 – Cluster-specific prediction curves (blue) with associated 95% credible intervals (grey) from MagmaClust, for two unlikely clusters. The dashed lines represent the mean parameters from the mean processes estimates. Observed data points are in black, testing data points are in red. Backward points are the observations from the training dataset, coloured by clusters.

MagmaClust, we display the cluster-specific prediction (4.9) for the most probable group instead of the GPs mixture prediction, since $\max_k(\boldsymbol{\tau}_*) \approx 1$ in this example. It can be noticed that our method offers both a significant improvement in mean prediction and a narrowed uncertainty around this value.

This example highlights the benefit we can get from considering group-structured similarities between individuals in GP predictions. Additionally, we display on Figure 4.3 the specific predictions according to the two remaining clusters (although associated with almost 0 probabilities). Let us remark that the predictions move towards the cluster specific mean processes as soon as the observations become too distant. In this idealistic example, we displayed Gaussian predictive distributions for convenience since, in general, a Gaussian mixture might rarely be unimodal. Therefore, we propose in Figure 4.4 another example with a higher variance and groups that are tougher to separate. While the ARI between predicted and true clusters was equal to 1 (perfect match) in the previous example, it now decreases to 0.78. Moreover, the vector of membership probabilities associated with the Figure 4.4 for the predicted individual happens to be: $\boldsymbol{\tau}_* = (0.95, 0.05, 0)$. The left-hand graph provides an illustration of the predictive mean, acquired from the multi-task GPs mixture distribution described in Proposition 4.5. We may notice that this curve lies very close to one cluster's mean although not completely overlapping it, because of the $\tau_{*k} = 0.05$ probability for another cluster, which slightly pulls the prediction onto its own mean. Besides, the right-hand graph of Figure 4.4 proposes a representation of the multi-task GPs mixture distribution as a heatmap of probabilities for the location of our predictions. This way, we can display, even in this multi-modal context, a thorough visual quantification for both the dispersion of the predicted values and the confidence we may grant to each of them.

Finally, let us propose on Figure 4.5 an illustration of the capacity of MagmaClust to retrieve the shape of the underlying mean processes, by plotting their estimations $\{\hat{m}_k(\cdot)\}_k$ (dotted lines) along with the true curves (plain coloured lines) generated by the simulation scheme. The ability to perform this task generally depends on the structure of the data as well as on the initialisation, although we may observe satisfactory results both on the previous fuzzy example (left) and on a well-separated case (right). Let us remark that the

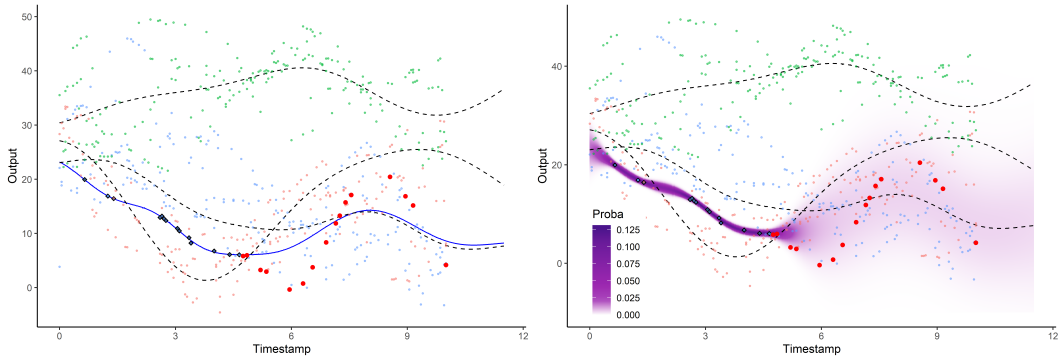

Figure 4.4 – Left: GPs mixture mean prediction curve (blue) from MagmaClust. Right: heatmap of probabilities for the GPs mixture predictive distribution from MagmaClust. The dashed lines represent the mean parameters from the mean processes estimates. Observed data points are in black, testing data points are in red. Backward points are the observations from the training dataset, coloured by clusters.

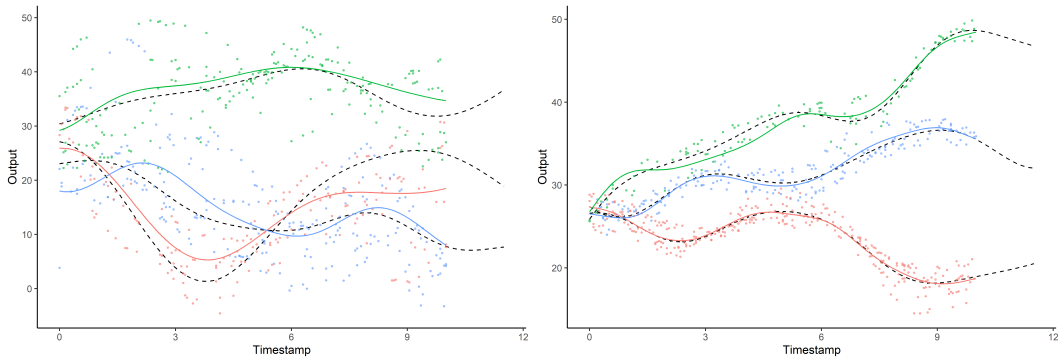

Figure 4.5 – **Left:** fuzzy case. **Right:** well-separated case. Curves of the simulated underlying mean processes, coloured by clusters. The dashed lines represent the mean parameters from the mean processes estimates. Backward points are the observations from the training dataset, coloured by clusters.

mean processes' estimations also come with uncertainty quantification, albeit not displayed on Figure 4.5 for the sake of clarity.

4.6.2 Clustering performance

As previously detailed in Chapter 2, curve clustering methods often struggle to handle irregularly observed data. Therefore, for the sake of fairness and to avoid introducing too many smoothing biases in alternative methods, the datasets used in the following are sampled on regular grids, although MagmaClust remains reasonably insensitive to this matter. The competing algorithms are the B-splines expansion associated with a kmeans algorithm proposed in Abraham et al. (2003) and funHDDC (Bouveyron and Jacques, 2011; Schmutz et al., 2018). A naive multivariate kmeans is used as initialisation for both funHDDC and MagmaClust. We propose on Figure 4.6 an evaluation of each algorithm in terms of ARI on 100 datasets, for each of the 4 different hypotheses of generating models ($\mathcal{H}_{ki}, \mathcal{H}_{k0}, \mathcal{H}_{0i}, \mathcal{H}_{00}$). It can be noticed that MagmaClust outperforms the alternatives in all situations. In par-

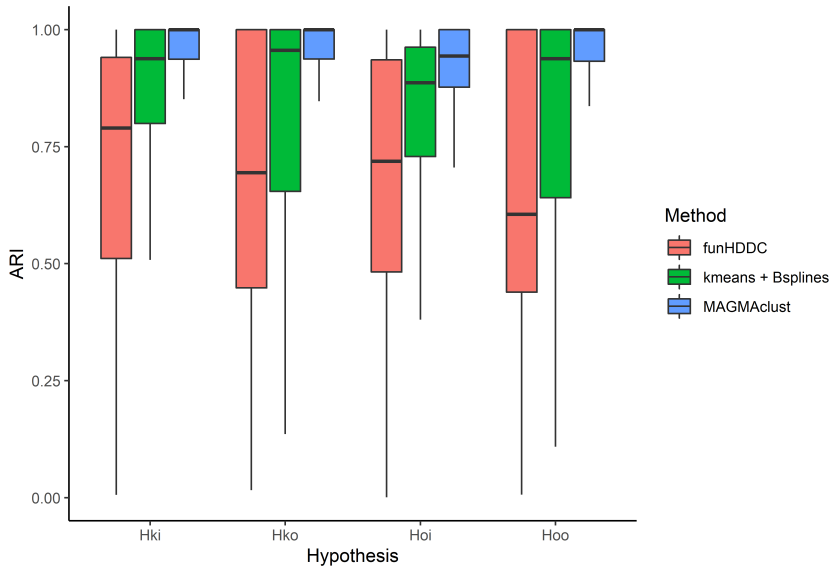

Figure 4.6 – Adjusted Rand Index values between the true clusters and the partitions estimated by kmeans, funHDDC, and MagmaClust. The value of K is set to the true number of clusters for all methods. The ARI is computed on 100 datasets for each generating model's assumption $\mathcal{H}_{ki}, \mathcal{H}_{k0}, \mathcal{H}_{0i}$, and $\mathcal{H}_{00}$.

ticular, our approach provides consistent results and a lower variance. Furthermore, while performances of the other methods are expected to deteriorate because of additional smoothing procedures in the case of irregular grids, MagmaClust would run the same without any change.

On another aspect, Figure 4.7 provides some insights into the robustness of MagmaClust to a wrong setting of K, the number of clusters. For 100 datasets with a true value $K^* = 3$, the ARI has been computed between the true partitions and the ones estimated by MagmaClust initialised with different settings $K = 2, \ldots, 10$. Except for $K = 2$ where the low number of clusters prevents from getting enough matching pairs by definition, we may notice relatively unaffected performances as K increases. Despite a non-negligible variance in results, the partitions remain consistent overall, and the clustering performances of MagmaClust seem pretty robust to misspecification of K.

4.6.3 Prediction performance

Another piece of evidence for this robustness is highlighted by Table 4.3 in the context of forecasting. The predictive aspect of MagmaClust remains the main purpose of the method and its performances of this task partly rely on the adequate clustering of the individuals. It may be noticed on Table 4.3 that both MSE and $WCIC_{95}$ regularly but slowly deteriorate as we move away from the true value of K. However, the performances remain of the same order, and we may still be confident about the predictions obtained through a misspecified running of MagmaClust. In particular, the values of MSE happen to be even better when setting $K = 4, \ldots, 6$ (we recall that the same 100 datasets are used in all cases, which can thus be readily compared). Besides, the right-hand part of the table

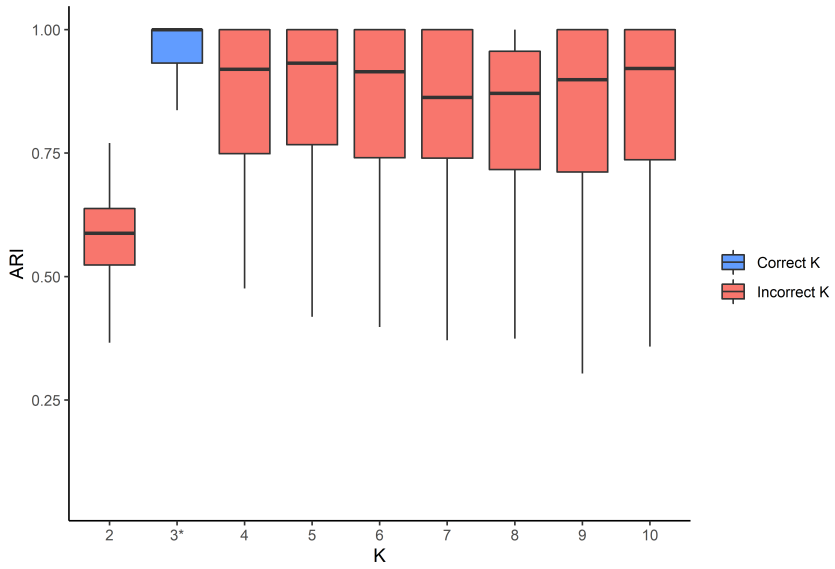

Figure 4.7 – Adjusted Rand Index values between the true clusters and the partitions estimated by MagmaClust with respect to the values of K used as setting. The ARI is computed on the same 100 datasets for each value of K. (3*: the true number of clusters for all datasets)

provides indications on the relative time (in seconds) that is needed to train the model for one dataset and to make predictions. As expected, both training and prediction times increase roughly linearly with the values of K, which seems consistent with the complexities exposed in Section 4.5. This property is a consequence of the extra mean processes and hyper-parameters that need to be estimated as K grows. Nonetheless, the influence of K is lesser on the prediction time, which yet remains negligible, even when computing many group-specific predictions.

K	MSE	$WCIC_{95}$	Training time	Prediction time
2	7.7 (18.4)	92 (20.3)	70.4 (25)	0.4 (0.1)
3*	3.7 (8.1)	**95 (13.2)**	97.7 (33.2)	0.5 (0.1)
4	3.2 (5.3)	94.9 (13.6)	116.5 (47.3)	0.6 (0.2)
5	3.2 (5.6)	94.4 (14.3)	133 (40.8)	0.6 (0.2)
6	**3.1 (5.4)**	94.4 (13.6)	153.3 (42)	0.8 (0.3)
7	4 (9)	93.6 (15.4)	173.7 (45.1)	1 (0.4)
8	4.7 (13)	93.8 (16)	191.3 (44.7)	1 (0.3)
9	4.1 (9.5)	94 (14.6)	211.6 (52)	0.8 (0.4)
10	4.5 (14.8)	94.4 (14.4)	235 (52.7)	1.8 (1.4)

Table 4.3 – Average (sd) values of MSE, $WCIC_{95}$, training and prediction times (in secs) on 100 runs for different numbers of clusters as setting for MagmaClust. (3* : the true number of clusters for all datasets)

To pursue the matter of prediction, let us provide on Table 4.4 the comparative results between GP regression, Magma, and MagmaClust. On the group-structured datasets

generated by the simulation scheme, our approach outperforms these alternatives. In terms of MSE, MagmaClust takes advantage of its multiple mean processes to provide enhanced predictions. Moreover, the quantification of uncertainty appears highly satisfactory since there are effectively 95% of the observations lying within the weighted CI_{95}, as expected. It is important to note that Magma is merely equivalent to MagmaClust with the setting $K = 1$. Therefore, the latter can be seen as a generalisation of the former, although no particular gain should be expected in the absence of group structure in the data. Once again, the increase in training and prediction times displayed in Table 4.4 remains proportional to the value of K (we recall that MagmaClust assumes $K = 3$ here).

	MSE	$WCIC_{95}$	Training time	Prediction time
GP	138 (174)	78.4 (31.1)	0 (0)	0.6 (0.1)
Magma	31.7 (45)	84.4 (27.9)	61.1 (25.7)	0.5 (0.2)
MagmaClust	**3.7 (8.1)**	**95 (13.2)**	132 (55.6)	0.6 (0.2)

Table 4.4 – Average (sd) values of MSE, $WCIC_{95}$, training and prediction times (in secs) on 100 runs for GP, Magma and MagmaClust.

4.6.4 Application of MagmaClust on swimmers' progression curves

In this paragraph, the datasets initially proposed in Section 3.6.5, gathering 100m race's performances for female and male swimmers, are analysed in the new light of MagmaClust. Let us recall that we aim at modelling a curve of progression from competition results for each individual in order to forecast their future performances. Assuming that a process $y_i(\cdot)$, defined as previously, has generated the data of the i-th swimmer, we expect MagmaClust to provide relevant predictions by taking advantage both of its multi-task feature and the group structure highlighted in Chapter 1. For this datasets indeed, it has already been exhibited that the swimmers can be grouped into 5 different clusters according to their pattern of progression. In the absence of a dedicated method of model selection for the current version of MagmaClust's implementation, the number of clusters is then set to $K = 5$ in this analysis. To evaluate the efficiency of our approach in this real-life application, the individuals are split into training and testing sets (in proportions $60\% - 40\%$). The prior mean functions $\{m_k(\cdot)\}_k$ are set to be constant equal to 0. In this context of relatively monotonic variations among progression curves, the hypothesis $\mathcal{H}_{00}$ is specified for the hyper-parameters, which are initialised to be $\theta_0 = \gamma_0 = (e^1, e^1)$ and $\sigma_0 = 0.04$. Those values are the default in MagmaClust and remain adequate for this framework. For both men and women, the hyper-parameters, the mean processes and the cluster's membership probabilities are learnt on the training data set. Then, the data points of each testing individual are split for evaluation purpose between observed (the first 80%) and testing values (the remaining 20%). Therefore, each new process $y_*(\cdot)$ associated with a test individual is assumed to be partially observed, and its testing values are used to compute MSE and $WCIC_{95}$ for the predictions.

As exhibited by Table 4.5, MagmaClust offers excellent performances on both datasets and slightly improves Magma's predictions. Values of MSE and $WCIC_{95}$ appear satisfactory although one may fairly argue that the gain provided by the cluster-specific predictions remains mild in this context. One of the explaining reasons is highlighted in the bottom graph of Figure 4.8. Although clear distinctions between the different patterns of progression

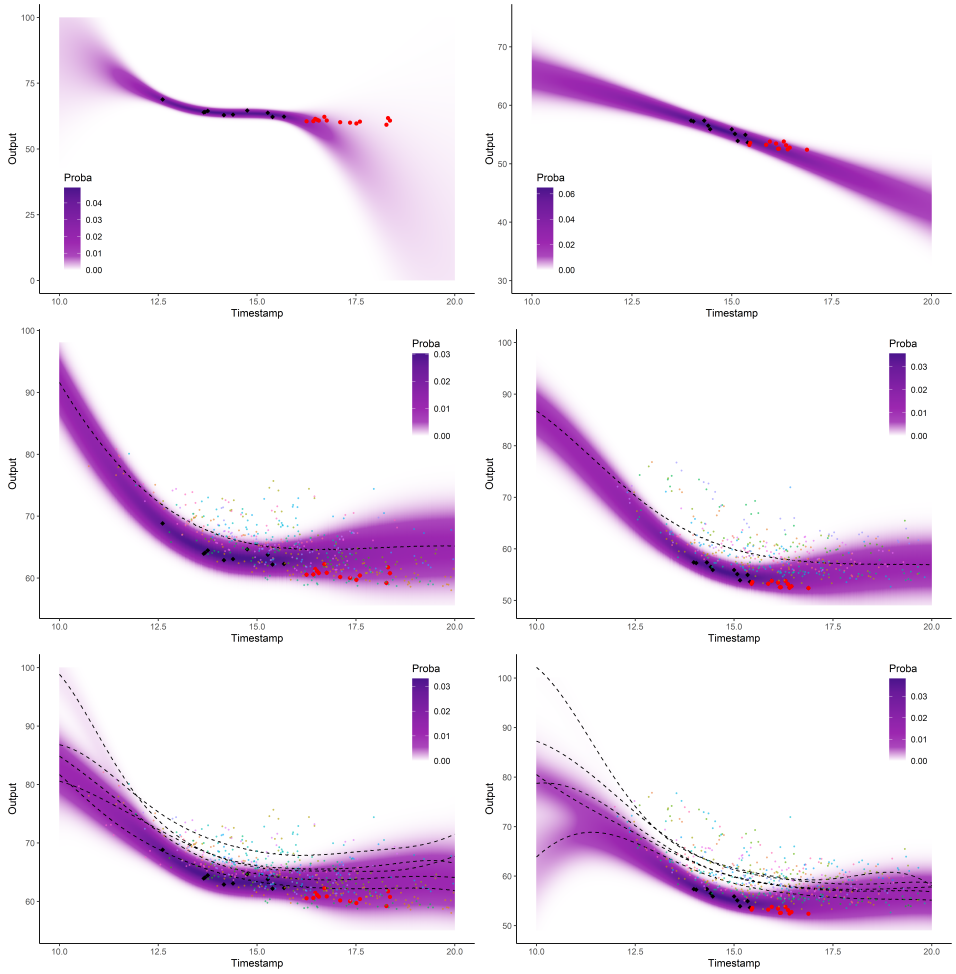

Figure 4.8 – **Left:** women dataset. **Right:** men dataset. Prediction and uncertainty obtained through GP (top), MAGMA (middle), and MAGMACLUST (bottom) for a random illustrative swimmer. The dashed lines represent the mean parameters from the mean processes estimates. Observed data points are in black, testing data points are in red. Backward points are the observations from the training dataset.

occur at young ages, the differences tend to narrow afterwards. Hence, the cluster's mean processes appear pretty close to each other at older ages, especially in regards to the overall signal-on-noise ratio. Nevertheless, MagmaClust provides several additional insights into this problem compared to Magma.

First, the clusters offer interesting results in themselves to assess the profile of young swimmers and to determine the individuals to whom they most resemble. In particular, it is also possible to differentiate future evolutions associated with each cluster, along with their probabilities to occur (we do not display all the cluster-specific predictions here for the sake of concision). On the other hand, our method leads to tighter predictive distributions in terms of uncertainty. Compared to Magma that uses all training data identically, MagmaClust somehow discards the superfluous information, through the weights τ_{*k}, to only retain the most relevant cluster-specific mean processes. Letting aside the GP regression that is generally too limited, Magma exhibits on Figure 4.8 satisfactory results, for which the uncertainty encompasses most the dispersion of training data. However, for both men and women examples, MagmaClust offers narrower predictions than Magma, by ignoring most of the data coming from the two upper clusters.

		MSE	$WCIC_{95}$
Women	GP	22.8 (84.7)	77.5 (30.4)
	Magma	3.7 (5.6)	**93.5 (15.6)**
	MagmaClust	**3.5 (5.3)**	92.2 (15.8)
Men	GP	19.6 (86)	80.7 (29.5)
	Magma	2.5 (3.8)	95.6 (12.7)
	MagmaClust	**2.4 (3.6)**	**94.5 (14.2)**

Table 4.5 – Average (sd) values of MSE and $WCIC_{95}$ for GP, Magma and MagmaClust on the french swimmer testing datasets.

Let us point out that, whereas the predictions at older ages remain roughly similar, the multi-modal aspect of MagmaClust distributions occurs more clearly between 10 and 12 years, where the highest probabilities smoothly follow the clusters' mean. Overall, although we shall expect even more noticeable improvements in applications with well-separated groups of individuals, the swimmers' progression curves example highlights MagmaClust's potential for tackling this kind of clustering and forecast problems.

4.7 Discussion

Throughout this chapter, we introduced a novel framework to handle clustering and regression purposes with a multi-task GPs mixture model. This approach, called MagmaClust, extends the algorithm Magma presented in the previous chapter to deal with group-structured data more efficiently. The method provides new insights on the matter of GP regression by introducing cluster-specific modelling and predictions while remaining efficiently tractable through the use of variational approximations for inference. Moreover, this nonparametric probabilist framework accounts for uncertainty both in the clustering aspect and in the final predictions, which appears to be notable in the learning literature. We demonstrated the practical efficiency of MagmaClust on both synthetic and real

datasets where it over-performs the alternatives, particularly in group-structured context. Even though the main concern of our method remains the predictive abilities, the clustering performances also deserve to be noticed since results comparable to state-of-the-art functional clustering algorithms are reached as well.

While we recall that computational cost is of paramount importance to ensure broad applicability of GP models, the present version of MagmaClust yet lacks a sparse approximation. As MagmaClust however, one of the state-of-the-art sparse method (Titsias, 2009; Bauer et al., 2016) makes use of variational inference, both to select pseudo-inputs and learn hyper-parameters of GP models. Therefore, an interesting extension could come from simultaneously computing $\{\mu_k(\cdot)\}_k$'s hyper-posteriors and pseudo-inputs, allowing for a sparse approximation of the highest dimensional object in our model. Besides, the traditional model selection's problem of finding the number of groups in clustering applications have been purposefully set aside in the present paper. Tackling this issue, which is required in our GPs mixture model, is generally non-trivial and many criteria have been developed in this sense from the earliest AIC (Akaike, 1974) and BIC (Schwarz, 1978) to most recent proposals such as ICL (Biernacki et al., 2000) or the slope heuristic (Birgé and Massart, 2006; Baudry et al., 2012). As we mainly work here with tractable likelihoods, the adaptation of efficient heuristics to develop specific model selection tools seems achievable, although the presence of multiple latent processes needs to be carefully dealt with. Overall, we believe that MagmaClust provides a valuable methodological contribution, initially tailored to handle the swimmers' curves application in this thesis, standing as a significant extension of the GP framework for dealing with a wider range of problems.

4.8 Proofs

4.8.1 Proof of Proposition 4.1

Let us note $\mathbb{E}_{\boldsymbol{\mu}}$ the expectation with respect to the variational distribution $\hat{q}_{\boldsymbol{\mu}}(\boldsymbol{\mu})$. From Bishop (2006, Chapter 10), the optimal solution $\hat{q}_{\mathbf{Z}}(\mathbf{Z})$ to the variational formulation verifies:

$$
\begin{aligned}
\log \hat{q}_{\mathbf{Z}}(\mathbf{Z}) &= \mathbb{E}_{\boldsymbol{\mu}}\left[\log p(\{\mathbf{y}_i\}_i, \mathbf{Z}, \boldsymbol{\mu} \mid \hat{\Theta})\right] + C_1 \\
&= \mathbb{E}_{\boldsymbol{\mu}}\left[\log p(\{\mathbf{y}_i\}_i \mid \mathbf{Z}, \boldsymbol{\mu}, \left\{\hat{\theta}_i\right\}_i, \{\hat{\sigma}_i^2\}_i) + \log p(\mathbf{Z} \mid \hat{\boldsymbol{\pi}}) + \log p(\boldsymbol{\mu} \mid \{\hat{\gamma}_k\}_k)\right] + C_1 \\
&= \mathbb{E}_{\boldsymbol{\mu}}\left[\log p(\{\mathbf{y}_i\}_i \mid \mathbf{Z}, \boldsymbol{\mu}, \left\{\hat{\theta}_i\right\}_i, \{\hat{\sigma}_i^2\}_i)\right] + \log p(\mathbf{Z} \mid \hat{\boldsymbol{\pi}}) + C_2 \\
&= \mathbb{E}_{\boldsymbol{\mu}}\left[\sum_{i=1}^{M}\sum_{k=1}^{K} Z_{ik} \log p(\mathbf{y}_i \mid Z_{ik} = 1, \mu_k(\mathbf{t}_i), \hat{\theta}_i, \hat{\sigma}_i^2)\right] + \sum_{i=1}^{M}\sum_{k=1}^{K} Z_{ik} \log(\hat{\pi}_k) + C_2 \\
&= \sum_{i=1}^{M}\sum_{k=1}^{K} Z_{ik}\left[\log(\hat{\pi}_k) + \mathbb{E}_{\mu_k}\left[\log p(\mathbf{y}_i \mid Z_{ik} = 1, \mu_k(\mathbf{t}_i), \hat{\theta}_i, \hat{\sigma}_i^2)\right]\right] + C_2 \\
&= \sum_{i=1}^{M}\sum_{k=1}^{K} Z_{ik}\left[\log(\hat{\pi}_k) - \frac{1}{2}\log\left|\boldsymbol{\Psi}^{\mathbf{t}_i}_{\hat{\theta}_i, \hat{\sigma}_i^2}\right|\right. \\
&\qquad\qquad \left. - \frac{1}{2}\mathbb{E}_{\mu_k}\left[(\mathbf{y}_i - \mu_k(\mathbf{t}_i))^\intercal {\boldsymbol{\Psi}^{\mathbf{t}_i}_{\hat{\theta}_i, \hat{\sigma}_i^2}}^{-1} (\mathbf{y}_i - \mu_k(\mathbf{t}_i))\right]\right] + C_3.
\end{aligned}
$$

Lemma 4.1. *Let $X \in \mathbb{R}^N$ be a random Gaussian vector $X \sim \mathcal{N}(m, \mathbf{K})$, $b \in \mathbb{R}^N$, and $\mathbf{S}$, a $N \times N$ covariance matrix. Then:*

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

Proof.

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

□

Applying Lemma 4.1 to the expectation in the right hand term of the previous expression, we obtain:

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

where (by inspection of both Gaussian and multinomial distributions):

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

Therefore, the optimal solution may be written as a factorised form of multinomial distributions:

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

4.8.2 Proof of Proposition 4.2

Let us denote by $\mathbb{E}_{\mathbf{Z}}$ the expectation with respect to the variational distribution $\hat{q}_{\mathbf{Z}}(\mathbf{Z})$. From Bishop (2006, Chapter 10), the optimal solution $\hat{q}_{\boldsymbol{\mu}}(\boldsymbol{\mu})$ to the variational formulation verifies:

$$
\begin{aligned}
\log \hat{q}_{\boldsymbol{\mu}}(\boldsymbol{\mu}) &= \mathbb{E}_{\mathbf{Z}}\left[\log p(\{\mathbf{y}_i\}_i, \mathbf{Z}, \boldsymbol{\mu} \mid \hat{\Theta})\right] + C_1 \\
&= \mathbb{E}_{\mathbf{Z}}\left[\log p(\{\mathbf{y}_i\}_i \mid \mathbf{Z}, \boldsymbol{\mu}, \left\{\hat{\theta}_i\right\}_i, \{\hat{\sigma}_i^2\}_i) + \log p(\mathbf{Z} \mid \hat{\boldsymbol{\pi}}) + \log p(\boldsymbol{\mu} \mid \{\hat{\gamma}_k\}_k)\right] + C_1 \\
&= \mathbb{E}_{\mathbf{Z}}\left[\log p(\{\mathbf{y}_i\}_i \mid \mathbf{Z}, \boldsymbol{\mu}, \left\{\hat{\theta}_i\right\}_i, \{\hat{\sigma}_i^2\}_i)\right] + \log p(\boldsymbol{\mu} \mid \{\hat{\gamma}_k\}_k) + C_2 \\
&= \sum_{i=1}^{M} \mathbb{E}_{\mathbf{Z}_i}\left[\log p(\mathbf{y}_i \mid \mathbf{Z}_i, \boldsymbol{\mu}, \hat{\theta}_i, \hat{\sigma}_i^2)\right] + \sum_{k=1}^{K} \log p(\mu_k(\mathbf{t}) \mid \hat{\gamma}_k) + C_2 \\
&= \sum_{i=1}^{M}\sum_{k=1}^{K} \mathbb{E}_{\mathbf{Z}_i}\left[\, Z_{ik} \,\right] \log p(\mathbf{y}_i \mid Z_{ik} = 1, \mu_k(\mathbf{t}_i), \hat{\theta}_i, \hat{\sigma}_i^2) + \sum_{k=1}^{K} \log p(\mu_k(\mathbf{t}) \mid \hat{\gamma}_k) + C_2 \\
&= -\frac{1}{2}\sum_{k=1}^{K}\Bigg[(\mu_k(\mathbf{t}) - m_k(\mathbf{t}))^\intercal {\mathbf{C}^{\mathbf{t}}_{\hat{\gamma}_k}}^{-1}(\mu_k(\mathbf{t}) - m_k(\mathbf{t})) \\
&\qquad + \sum_{i=1}^{M} \tau_{ik}(\mathbf{y}_i - \mu_k(\mathbf{t}_i))^\intercal {\boldsymbol{\Psi}^{\mathbf{t}_i}_{\hat{\theta}_i, \hat{\sigma}_i^2}}^{-1}(\mathbf{y}_i - \mu_k(\mathbf{t}_i))\Bigg] + C_3.
\end{aligned}
$$

If we regroup the scalar coefficient τ_{ik} with the covariance matrix ${\boldsymbol{\Psi}^{\mathbf{t}_i}_{\hat{\theta}_i, \hat{\sigma}_i^2}}^{-1}$, we simply recognise two quadratic terms of Gaussian likelihoods on the variables $\mu_k(\cdot)$, although evaluated onto different sets of timestamps $\mathbf{t}$ and $\mathbf{t}_i$. By taking some writing cautions and expanding the vector-matrix products entirely, it has been proved in Leroy et al. (2020b) that this expression factorises with respect to $\mu_k(\mathbf{t})$ simply by expanding vectors $\mathbf{y}_i$ and matrices $\boldsymbol{\Psi}^{\mathbf{t}_i}_{\hat{\theta}_i, \hat{\sigma}_i^2}$ with zeros, $\forall t \in \mathbf{t},\ t \notin \mathbf{t}_i$. Namely, let us note:

- $\forall i \in \mathcal{I},\ \tilde{\mathbf{y}}_i = \left(\mathbb{1}_{[t \in \mathbf{t}_i]} \times y_i(t)\right)_{t \in \mathbf{t}}$, a N-dimensional vector,
- $\forall i \in \mathcal{I},\ \tilde{\boldsymbol{\Psi}}_i = \left[\, \mathbb{1}_{[t,t' \in \mathbf{t}_i]} \times \psi_{\hat{\theta}_i, \hat{\sigma}_i^2}(t, t') \,\right]_{t,t' \in \mathbf{t}}$, a $N \times N$ matrix.

Therefore:

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

By inspection, we recognise a sum of a Gaussian log-likelihoods, which implies the underlying values of the constants. Finally:

$$
\hat{q}_{\boldsymbol{\mu}}(\boldsymbol{\mu}) = \prod_{k=1}^{K} \mathcal{N}\left(\mu_k(\mathbf{t}); \hat{m}_k(\mathbf{t}), \widehat{\mathbf{C}}^{\mathbf{t}}_k\right), \tag{4.10}
$$

with:

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

4.8.3 Proof of Proposition 4.3

Let us note $\mathbb{E}_{\mathbf{Z},\boldsymbol{\mu}}$ the expectation with respect to the optimised variational distributions $\hat{q}_{\mathbf{Z}}(\mathbf{Z})$ and $\hat{q}_{\boldsymbol{\mu}}(\boldsymbol{\mu})$. From Bishop (2006, Chapter 10), we can figure out the optimal values for the hyper-parameters Θ by maximising the lower bound $\mathcal{L}(\hat{q};\Theta)$ with respect to Θ:

$$\hat{\Theta} = \underset{\Theta}{\operatorname{argmax}}\, \mathcal{L}(\hat{q};\Theta).$$

Moreover, we can develop the formulation of the lower bound by expressing the integrals as an expectation, namely $\mathbb{E}_{\mathbf{Z},\boldsymbol{\mu}}$. Recalling the complete-data likelihood analytical expression and focusing on quantities depending upon Θ, we can write:

$$\begin{aligned}
\mathcal{L}(\hat{q};\Theta) &= -\mathbb{E}_{\{\mathbf{Z},\boldsymbol{\mu}\}}\left[\underbrace{\log \hat{q}_{\mathbf{Z},\boldsymbol{\mu}}(\mathbf{Z},\boldsymbol{\mu})}_{constant\ w.r.t.\ \Theta} - \log p(\{\mathbf{y}_i\}_i, \mathbf{Z}, \boldsymbol{\mu} \mid \Theta)\right] \\
&= \mathbb{E}_{\{\mathbf{Z},\boldsymbol{\mu}\}}\left[\log \prod_{k=1}^{K}\left\{\mathcal{N}\left(\mu_k(\mathbf{t}); m_k(\mathbf{t}), \mathbf{C}^{\mathbf{t}}_{\gamma_k}\right)\prod_{i=1}^{M}\left(\pi_k \mathcal{N}\left(\mathbf{y}_i; \mu_k(\mathbf{t}_i), \boldsymbol{\Psi}^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}\right)\right)^{Z_{ik}}\right\}\right] + C_1 \\
&= \sum_{k=1}^{K}\Bigg[-\frac{1}{2}\left(\log\left|\mathbf{C}^{\mathbf{t}}_{\gamma_k}\right| + \mathbb{E}_{\boldsymbol{\mu}}\left[(\mu_k(\mathbf{t}) - m_k(\mathbf{t}))^{\intercal}{\mathbf{C}^{\mathbf{t}}_{\gamma_k}}^{-1}(\mu_k(\mathbf{t}) - m_k(\mathbf{t}))\right]\right) \\
&\qquad - \frac{1}{2}\sum_{i=1}^{M}\mathbb{E}_{\{\mathbf{Z},\boldsymbol{\mu}\}}\left[Z_{ik}\left(\log\left|\boldsymbol{\Psi}^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}\right| + (\mathbf{y}_i - \mu_k(\mathbf{t}_i))^{\intercal}{\boldsymbol{\Psi}^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}}^{-1}(\mathbf{y}_i - \mu_k(\mathbf{t}_i))\right)\right] \\
&\qquad + \sum_{i=1}^{M}\mathbb{E}_{\mathbf{Z}}\left[Z_{ik}\right]\log\pi_k\Bigg] + C_2 \\
&= \sum_{k=1}^{K}\Bigg[-\frac{1}{2}\left(\log\left|\mathbf{C}^{\mathbf{t}}_{\gamma_k}\right| + (\hat{m}_k(\mathbf{t}) - m_k(\mathbf{t}))^{\intercal}{\mathbf{C}^{\mathbf{t}}_{\gamma_k}}^{-1}(\hat{m}_k(\mathbf{t}) - m_k(\mathbf{t})) + \operatorname{tr}\left(\widehat{\mathbf{C}}^{\mathbf{t}}_k{\mathbf{C}^{\mathbf{t}}_{\gamma_k}}^{-1}\right)\right) \\
&\qquad - \frac{1}{2}\sum_{i=1}^{M}\tau_{ik}\left(\log\left|\boldsymbol{\Psi}^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}\right| + (\mathbf{y}_i - \hat{m}_k(\mathbf{t}_i))^{\intercal}{\boldsymbol{\Psi}^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}}^{-1}(\mathbf{y}_i - \hat{m}_k(\mathbf{t}_i)) + \operatorname{tr}\left(\widehat{\mathbf{C}}^{\mathbf{t}}_k{\boldsymbol{\Psi}^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}}^{-1}\right)\right) \\
&\qquad + \sum_{i=1}^{M}\tau_{ik}\log\pi_k\Bigg] + C_2,
\end{aligned}$$

where we made use of Lemma 4.1 twice, at the first and second lines for the last equality. Let us note that, by reorganising the terms on the second line, there exists another formulation of this lower bound that allows for better managing of the computational resources. For information, we give this expression below since it is the quantity implemented in the current version of the MagmaClust code:

$$\mathcal{L}(\hat{q};\Theta) = -\frac{1}{2}\sum_{k=1}^{K}\left[\log\left|{\mathbf{C}^{\mathbf{t}}_{\gamma_k}}^{-1}\right| + (\hat{m}_k(\mathbf{t}) - m_k(\mathbf{t}))^\intercal {\mathbf{C}^{\mathbf{t}}_{\gamma_k}}^{-1}(\hat{m}_k(\mathbf{t}) - m_k(\mathbf{t})) + \operatorname{tr}\left(\widehat{\mathbf{C}}^{\mathbf{t}}_k {\mathbf{C}^{\mathbf{t}}_{\gamma_k}}^{-1}\right)\right]$$
$$-\frac{1}{2}\sum_{i=1}^{M}\left[\log\left|{\mathbf{\Psi}^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}}\right| + \mathbf{y}_i^\intercal {\mathbf{\Psi}^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}}^{-1}\mathbf{y}_i - 2\mathbf{y}_i^\intercal {\mathbf{\Psi}^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}}^{-1}\sum_{k=1}^{K}\boldsymbol{\tau}_{ik}\hat{m}_k(\mathbf{t}_i)\right.$$
$$\left. + \operatorname{tr}\left({\mathbf{\Psi}^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}}^{-1}\sum_{k=1}^{K}\boldsymbol{\tau}_{ik}\left(\hat{m}_k(\mathbf{t}_i)\hat{m}_k(\mathbf{t}_i)^\intercal + \widehat{\mathbf{C}}^{\mathbf{t}}_k\right)\right)\right]$$
$$+\sum_{k=1}^{K}\sum_{i=1}^{M}\boldsymbol{\tau}_{ik}\left(\log\pi_k\right) + C_2.$$

Regardless of the expression we choose for the following, we can notice that we expressed the lower bound $\mathcal{L}(q;\Theta)$ as a sum where the hyper-parameters $\{\gamma_k\}_k$, $\{\{\theta_i\}_i, \{\sigma_i^2\}_i\}$ and $\boldsymbol{\pi}$ appear in separate terms. Hence, the resulting maximisation procedures are independent of each other. First, let us focus on the simplest term that concerns $\boldsymbol{\pi}$, for which we have an analytical update equation. Since there is a constraint on the sum $\sum_{k=1}^{K}\pi_k = 1$, we first need to introduce a Lagrange multiplier in the expression to maximise:

$$\lambda\left(\sum_{k=1}^{K}\pi_k - 1\right) + \mathcal{L}(q;\Theta). \tag{4.11}$$

Setting to 0 the gradient with respect to π_k in (4.11), we get:

$$\lambda + \frac{1}{\pi_k}\sum_{i=1}^{M}\boldsymbol{\tau}_{ik} = 0,\ \forall k \in \mathcal{K}.$$

Multiplying by π_k and summing over k, we deduce the value of λ:

$$\sum_{k=1}^{K}\pi_k\lambda = -\sum_{k=1}^{K}\sum_{i=1}^{M}\boldsymbol{\tau}_{ik}$$
$$\lambda \times 1 = -\sum_{i=1}^{M}1$$
$$\lambda = -M.$$

Therefore, the optimal values for π_k are expressed as:

$$\hat{\pi}_k = \frac{1}{M}\sum_{i=1}^{M}\boldsymbol{\tau}_{ik},\ \forall k \in \mathcal{K}. \tag{4.12}$$

Concerning the remaining hyper-parameters, in the absence of analytical optima, we have no choice but to numerically maximise the corresponding terms in $\mathcal{L}(\hat{q};\Theta)$, namely:

$$-\frac{1}{2}\sum_{k=1}^{K}\left(\log\left|{\mathbf{C}^{\mathbf{t}}_{\gamma_k}}^{-1}\right| + (\hat{m}_k(\mathbf{t}) - m_k(\mathbf{t}))^\intercal {\mathbf{C}^{\mathbf{t}}_{\gamma_k}}^{-1}(\hat{m}_k(\mathbf{t}) - m_k(\mathbf{t})) + \operatorname{tr}\left(\widehat{\mathbf{C}}^{\mathbf{t}}_k {\mathbf{C}^{\mathbf{t}}_{\gamma_k}}^{-1}\right)\right), \tag{4.13}$$

and

$$-\frac{1}{2}\sum_{i=1}^{M}\sum_{k=1}^{K}\tau_{ik}\left(\log\left|\mathbf{\Psi}^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}\right|+(\mathbf{y}_i-\hat{m}_k(\mathbf{t}_i))^{\intercal}{\mathbf{\Psi}^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}}^{-1}(\mathbf{y}_i-\hat{m}_k(\mathbf{t}_i))+\operatorname{tr}\left(\widehat{\mathbf{C}}^{\mathbf{t}}_k{\mathbf{\Psi}^{\mathbf{t}_i}_{\theta_i,\sigma_i^2}}^{-1}\right)\right). \tag{4.14}$$

It is straightforward to see that some manipulations of linear algebra also allows the derivation of explicit gradients with respect to $\{\gamma_k\}_k$, $\{\theta_i\}_i$ and $\left\{\sigma_i^2\right\}_i$. Hence, we may take advantage of efficient gradient-based methods to handle the optimisation process. Let us stress that the quantity (4.13) is a sum on the sole values of k, whereas (4.14) also implies a sum on the values of i. Hence, each term of these sums involves only one hyper-parameter at a time, which thus may be optimised apart from the others. Conversely, if we assume all individuals (respectively all clusters) to share the same set of hyper-parameters, then the full sum has to be maximised upon at once. Therefore, recalling that we introduced 4 different settings according to whether we consider common or specific hyper-parameters for both clusters and individuals, we shall notice the desired maximisation problems that are induced by (4.13) and (4.14).

Bibliography

C. Abraham, P. A. Cornillon, E. Matzner-Løber, and N. Molinari. Unsupervised Curve Clustering using B-Splines. *Scandinavian Journal of Statistics*, 30(3):581–595, Sept. 2003. ISSN 1467-9469. doi: 10.1111/1467-9469.00350. [Cited on pages 5, 8, and 110.]

M. A. Aiserman, È. M. Braverman, and L. I. Rozonoer. Theoretical foundations of the potential function method in pattern recognition. *Automation and Remote Control*, 25 (6):917–936, 1964. [Cited on page 12.]

H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, Dec. 1974. ISSN 1558-2523. doi: 10.1109/TAC.1974.1100705. [Cited on pages 24 and 116.]

A. M. Alaa and M. van der Schaar. Bayesian Inference of Individualized Treatment Effects using Multi-task Gaussian Processes. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3424–3432. Curran Associates, Inc., 2017. [Cited on pages 30 and 62.]

M. A. Álvarez and N. D. Lawrence. Computationally Efficient Convolved Multiple Output Gaussian Processes. *Journal of Machine Learning Research*, 12(41):1459–1500, 2011. ISSN 1533-7928. [Cited on page 20.]

M. A. Álvarez, L. Rosasco, and N. D. Lawrence. Kernels for Vector-Valued Functions: A Review. *Foundations and Trends® in Machine Learning*, 4(3):195–266, 2012. ISSN 1935-8237, 1935-8245. doi: 10.1561/2200000036. [Cited on page 62.]

O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J. M. Pérez, and I. Perona. An extensive comparative study of cluster validity indices. *Pattern Recognition*, 46(1):243–256, Jan. 2013. ISSN 0031-3203. doi: 10.1016/j.patcog.2012.07.021. [Cited on page 52.]

A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. In *Advances in Neural Information Processing Systems*, pages 41–48, 2007. [Cited on page 27.]

A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine learning*, 73(3):243–272, 2008. [Cited on page 27.]

S. Arlot. Minimal penalties and the slope heuristics: A survey. *Journal de la Société Française de Statistique*, 160(3), Oct. 2019. [Cited on pages 10 and 53.]

H. Attias. Inferring Parameters and Structure of Latent Variable Models by Variational Bayes. *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, 1999. [Cited on page 27.]

H. Attias. A Variational Baysian Framework for Graphical Models. In S. A. Solla, T. K. Leen, and K. Müller, editors, *Advances in Neural Information Processing Systems 12*, pages 209–215. MIT Press, 2000. [Cited on page 95.]

A. Banerjee, D. B. Dunson, and S. T. Tokdar. Efficient Gaussian process regression for large datasets. *Biometrika*, 100(1):75–89, 2013. ISSN 0006-3444. doi: 10.1093/biomet/ass068. [Cited on page 62.]

J.-P. Baudry, C. Maugis, and B. Michel. Slope heuristics: Overview and implementation. *Statistics and Computing*, 22(2):455–470, Mar. 2012. ISSN 0960-3174, 1573-1375. doi: 10.1007/s11222-011-9236-1. [Cited on pages 24 and 116.]

M. Bauer, M. van der Wilk, and C. E. Rasmussen. Understanding Probabilistic Sparse Gaussian Process Approximations. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 1533–1541. Curran Associates, Inc., 2016. [Cited on pages 19, 20, 42, 62, and 116.]

M. J. Beal and Z. Ghahramani. The Variational Bayesian EM Algorithm for Incomplete Data: With Application to Scoring Graphical Model Structures. *Oxford University Press*, page 10, 2003. [Cited on page 27.]

Y. Bengio. Gradient-Based Optimization of Hyperparameters. *Neural Computation*, 12(8): 1889–1900, Aug. 2000. ISSN 0899-7667. doi: 10.1162/089976600300015187. [Cited on page 98.]

J. Bernardo, J. Berger, A. Dawid, and A. Smith. Regression and classification using Gaussian process priors. *Bayesian statistics*, 6:475, 1998. [Cited on page 98.]

G. Berthelot, A. Sedeaud, A. Marck, J. Antero-Jacquemin, G. Saulière, A. Marc, F.-D. Desgorces, and J.-F. Toussaint. Has Athletic Performance Reached its Peak? *Sports Medicine (Auckland, N.Z.)*, 45(9):1263–1271, Sept. 2015. ISSN 1179-2035. doi: 10.1007/s40279-015-0347-2. [Cited on page 2.]

C. Biernacki and G. Govaert. Choosing models in model-based clustering and discriminant analysis. *Journal of Statistical Computation and Simulation*, 64(1):49–71, Aug. 1999. ISSN 0094-9655, 1563-5163. doi: 10.1080/00949659908811966. [Cited on page 24.]

C. Biernacki, G. Celeux, and G. Govaert. Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):719–725, July 2000. ISSN 1939-3539. doi: 10.1109/34.865189. [Cited on pages 24, 43, and 116.]

C. Biernacki, G. Celeux, and G. Govaert. Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate Gaussian mixture models. *Computational Statistics & Data Analysis*, 41(3):561–575, 2003. ISSN 0167-9473. doi: 10.1016/S0167-9473(02)00163-9. [Cited on pages 22, 68, and 99.]

H. Bijl, J.-W. van Wingerden, T. B. Schön, and M. Verhaegen. Online sparse Gaussian process regression using FITC and PITC approximations. *IFAC-PapersOnLine*, 48(28): 703–708, 2015. ISSN 2405-8963. doi: 10.1016/j.ifacol.2015.12.212. [Cited on pages 20 and 62.]

L. Birgé and P. Massart. Minimal Penalties for Gaussian Model Selection. *Probability Theory and Related Fields*, 138(1-2):33–73, 2006. ISSN 0178-8051, 1432-2064. doi: 10.1007/s00440-006-0011-8. [Cited on pages 10, 24, 43, 53, and 116.]

C. M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer, New York, 2006. ISBN 978-0-387-31073-2. [Cited on pages 14, 16, 17, 24, 26, 62, 71, 73, 102, 104, 116, 117, and 119.]

G. Boccia, P. Moisè, A. Franceschi, F. Trova, D. Panero, A. La Torre, A. Rainoldi, F. Schena, and M. Cardinale. Career Performance Trajectories in Track and Field Jumping Events from Youth to Senior Success: The Importance of Learning and Development. *PLoS ONE*, 12(1), Jan. 2017. ISSN 1932-6203. doi: 10.1371/journal.pone.0170744. [Cited on pages x, 2, 3, 32, and 55.]

E. V. Bonilla, K. M. Chai, and C. Williams. Multi-task Gaussian Process Prediction. In J. C. Platt, D. Koller, Y. Singer, and S. T. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 153–160. Curran Associates, Inc., 2008. [Cited on pages 28, 29, 42, 62, and 83.]

B. E. Boser, I. M. Guyon, and V. N. Vapnik. A Training Algorithm for Optimal Margin Classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152. ACM Press, 1992. [Cited on page 13.]

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, Dec. 2011. ISSN 1862-5355. doi: 10.1007/s11634-011-0095-6. [Cited on pages xi, 9, 34, 46, and 110.]

C. Bouveyron, S. Girard, and C. Schmid. High-Dimensional Data Clustering. *Computational Statistics & Data Analysis*, 52(1):502–519, Sept. 2007. ISSN 01679473. doi: 10.1016/j.csda.2007.02.009. [Cited on page 10.]

C. Bouveyron, L. Bozzi, J. Jacques, and F.-X. Jollois. The functional latent block model for the co-clustering of electricity consumption curves. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 67(4):897–915, Aug. 2018. ISSN 1467-9876. doi: 10.1111/rssc.12260. [Cited on pages 4 and 53.]

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, 2004. ISBN 978-0-521-83378-3. doi: 10.1017/CBO9780511804441. [Cited on pages 26 and 95.]

P. Boyle and M. Frean. Dependent Gaussian Processes. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 217–224. MIT Press, 2005. [Cited on page 20.]

P. J. Brockwell and R. A. Davis. *Time Series: Theory and Methods*. Springer Science & Business Media, Nov. 2013. ISBN 978-1-4899-0004-3. [Cited on page 3.]

D. L. Carey, K. M. Crossley, R. Whiteley, A. Mosler, K.-L. Ong, J. Crow, and M. E. Morris. Modelling Training Loads and Injuries: The Dangers of Discretization. *Medicine and Science in Sports and Exercise*, June 2018. ISSN 1530-0315. doi: 10.1249/MSS.0000000000001685. [Cited on pages 2 and 3.]

R. Caruana. Multitask Learning. *Machine Learning*, 28(1):41–75, July 1997. ISSN 1573-0565. doi: 10.1023/A:1007379606734. [Cited on pages 27 and 62.]

G. Casella. An Introduction to Empirical Bayes Data Analysis. *The American Statistician*, 39(2):83–87, 1985. ISSN 0003-1305. doi: 10.2307/2682801. [Cited on page 66.]

G. Celeux, D. Chauveau, and J. Diebolt. On Stochastic Versions of the EM Algorithm. *Rapport de recherche RR-1364, Inria*, 37(1):55–57, Dec. 1992. ISSN 0759-1063, 2070-2779. doi: 10.1177/075910639203700105. [Cited on page 22.]

K. Chen, P. Groot, J. Chen, and E. Marchiori. Generalized Spectral Mixture Kernels for Multi-Task Gaussian Processes. *arXiv:1808.01132 [cs, stat]*, Dec. 2018. [Cited on page 28.]

J.-M. Chiou and P.-L. Li. Functional clustering and identifying substructures of longitudinal data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69 (4):679–699, 2007. ISSN 1467-9868. doi: 10.1111/j.1467-9868.2007.00605.x. [Cited on page 46.]

C. Clingerman and E. Eaton. Lifelong Learning with Gaussian Processes. In M. Ceci, J. Hollmén, L. Todorovski, C. Vens, and S. Džeroski, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 10535, pages 690–704. Springer International Publishing, Cham, 2017. ISBN 978-3-319-71245-1 978-3-319-71246-8. doi: 10.1007/978-3-319-71246-8_42. [Cited on pages 20, 30, and 62.]

C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, Sept. 1995. ISSN 1573-0565. doi: 10.1007/BF00994018. [Cited on page 13.]

C. M. Crainiceanu and A. J. Goldsmith. Bayesian Functional Data Analysis Using WinBUGS. *Journal of statistical software*, 32(11), 2010. ISSN 1548-7660. [Cited on pages 11 and 63.]

N. Cressie. *Statistics for Spatial Data*. John Wiley & Sons, 1993. ISBN 978-1-119-11518-2. [Cited on page 14.]

L. Csató and M. Opper. Sparse on-line gaussian processes. *Neural Computation*, 14(3): 641–668, Mar. 2002. ISSN 0899-7667. doi: 10.1162/089976602317250933. [Cited on page 20.]

J. A. Cuesta-Albertos and R. Fraiman. Impartial trimmed k-means for functional data. *Computational Statistics & Data Analysis*, 51(10):4864–4877, June 2007. ISSN 0167-9473. doi: 10.1016/j.csda.2006.07.011. [Cited on page 9.]

C. de Boor. On calculating with B-splines. *Journal of Approximation Theory*, 6(1):50–62, July 1972. ISSN 0021-9045. doi: 10.1016/0021-9045(72)90080-9. [Cited on pages 3 and 5.]

A. Delaigle and P. Hall. Defining probability density for a distribution of random functions. *The Annals of Statistics*, 38(2):1171–1193, Apr. 2010. ISSN 0090-5364. doi: 10.1214/09-AOS741. [Cited on pages 8 and 9.]

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977. ISSN 0035-9246. [Cited on pages 21 and 66.]

D. Duvenaud. *Automatic Model Construction with Gaussian Processes*. Thesis, University of Cambridge, Nov. 2014. [Cited on pages 14, 18, 19, 65, and 94.]

K. A. Ericsson, R. R. Hoffman, A. Kozbelt, and A. M. Williams. *The Cambridge Handbook of Expertise and Expert Performance*. Cambridge University Press, May 2018. ISBN 978-1-108-62570-8. [Cited on page 2.]

T. Evgeniou, C. A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *Journal of machine learning research*, 6(Apr):615–637, 2005. [Cited on page 28.]

F. Ferraty and P. Vieu. *Nonparametric Functional Data Analysis: Theory and Practice*. Springer Science & Business Media, 2006. ISBN 978-0-387-36620-3. [Cited on pages 7, 9, and 63.]

S. E. Forrester and J. Townend. The effect of running velocity on footstrike angle – A curve-clustering approach. *Gait & Posture*, 41(1):26–32, Jan. 2015. ISSN 0966-6362, 1879-2219. doi: 10.1016/j.gaitpost.2014.08.004. [Cited on page 3.]

T. Gasser, H.-G. Muller, W. Kohler, L. Molinari, and A. Prader. Nonparametric Regression Analysis of Growth Curves. *The Annals of Statistics*, 12(1):210–229, 1984. ISSN 0090-5364. [Cited on page 4.]

M. G. Genton. Classes of kernels for machine learning: A statistics perspective. *The Journal of Machine Learning Research*, 2:299–312, Mar. 2002. ISSN 1532-4435. [Cited on page 13.]

M. G. Genton and W. Kleiber. Cross-Covariance Functions for Multivariate Geostatistics. *Statistical Science*, 30(2):147–163, 2015. ISSN 0883-4237. [Cited on page 30.]

M. Ghassemi, M. A. Pimentel, T. Naumann, T. Brennan, D. A. Clifton, P. Szolovits, and M. Feng. A Multivariate Timeseries Modeling Approach to Severity of Illness Assessment and Forecasting in ICU with Sparse, Heterogeneous Clinical Data. *Proceedings of the ... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence*, 2015:446–453, Jan. 2015. ISSN 2159-5399. [Cited on page 28.]

M. Giacofci, S. Lambert-Lacroix, G. Marot, and F. Picard. Wavelet-Based Clustering for Mixed-Effects Functional Models in High Dimension. *Biometrics*, 69(1):31–40, 2013. ISSN 1541-0420. doi: 10.1111/j.1541-0420.2012.01828.x. [Cited on pages 5, 9, and 47.]

R. Gómez-Bombarelli, J. N. Wei, D. Duvenaud, J. M. Hernández-Lobato, B. Sánchez-Lengeling, D. Sheberla, J. Aguilera-Iparraguirre, T. D. Hirzel, R. P. Adams, and A. Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018. [Cited on page 19.]

H. Goto, J. G. Morris, and M. E. Nevill. Influence of biological maturity on the match performance of 8 to 16 year old elite male youth soccer players. *Journal of Strength and Conditioning Research*, Feb. 2018. ISSN 1064-8011. doi: 10.1519/jsc.0000000000002510. [Cited on page 2.]

B. Guedj. A Primer on PAC-Bayesian Learning. In *Proceedings of the Second Congress of the French Mathematical Society*, May 2019. [Cited on page 11.]

B. Guedj and L. Li. Sequential Learning of Principal Curves: Summarizing Data Streams on the Fly. *arXiv:1805.07418 [cs, math, stat]*, May 2019. [Cited on page 7.]

T. Hastie and W. Stuetzle. Principal Curves. *Journal of the American Statistical Association*, 84(406):502–516, June 1989. ISSN 0162-1459. doi: 10.1080/01621459.1989.10478797. [Cited on page 7.]

R. J. Hathaway. Another interpretation of the EM algorithm for mixture distributions. *Statistics & Probability Letters*, 4(2):53–56, Mar. 1986. ISSN 0167-7152. doi: 10.1016/0167-7152(86)90016-7. [Cited on page 22.]

K. Hayashi, T. Takenouchi, R. Tomioka, and H. Kashima. Self-measuring Similarity for Multi-task Gaussian Process. *Transactions of the Japanese Society for Artificial Intelligence*, 27(3):103–110, 2012. ISSN 1346-8030, 1346-0714. doi: 10.1527/tjsai.27.103. [Cited on pages 29 and 62.]

N. E. Helwig, K. A. Shorter, P. Ma, and E. T. Hsiao-Wecksler. Smoothing spline analysis of variance models: A new tool for the analysis of cyclic biomechanical data. *Journal of Biomechanics*, 49(14):3216–3222, Mar. 2016. ISSN 1873-2380. doi: 10.1016/j.jbiomech.2016.07.035. [Cited on page 3.]

J. Hensman, N. Fusi, and N. D. Lawrence. Gaussian Processes for Big Data. *arXiv:1309.6835 [cs, stat]*, Sept. 2013. [Cited on pages 42, 62, 83, and 95.]

M. R. Hestenes and E. Stiefel. Methods of conjugate gradients for solving linear systems. *Journal of research of the National Bureau of Standards*, 49(6):409–436, 1952. [Cited on pages 17 and 98.]

L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, Dec. 1985. ISSN 1432-1343. doi: 10.1007/BF01908075. [Cited on page 107.]

F. Ieva, A. M. Paganoni, D. Pigoli, and V. Vitelli. Multivariate functional clustering for the morphological analysis of electrocardiograph curves. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 62(3):401–418, 2013. ISSN 1467-9876. doi: 10.1111/j.1467-9876.2012.01062.x. [Cited on page 9.]

J. Jacques and C. Preda. Funclust: A curves clustering method using functional random variables density approximation. *Neurocomputing*, 112:164–171, July 2013a. ISSN 0925-2312. doi: 10.1016/j.neucom.2012.11.042. [Cited on page 46.]

J. Jacques and C. Preda. Funclust: A curves clustering method using functional random variables density approximation. *Neurocomputing*, 112:164–171, July 2013b. ISSN 09252312. doi: 10.1016/j.neucom.2012.11.042. [Cited on page 10.]

J. Jacques and C. Preda. Functional data clustering: A survey. *Advances in Data Analysis and Classification*, 8(3):231–255, Sept. 2014. ISSN 1862-5347, 1862-5355. doi: 10.1007/s11634-013-0158-y. [Cited on pages 8, 9, and 55.]

A. Jalali, S. Sanghavi, C. Ruan, and P. K. Ravikumar. A Dirty Model for Multi-task Learning. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 964–972. Curran Associates, Inc., 2010. [Cited on page 28.]

G. M. James and C. A. Sugar. Clustering for Sparsely Sampled Functional Data. *Journal of the American Statistical Association*, 98(462):397–408, June 2003. ISSN 0162-1459, 1537-274X. doi: 10.1198/016214503000189. [Cited on pages 9 and 46.]

P. Jawanpuria and J. S. Nath. A convex feature learning formulation for latent task structure discovery. In *International Conference on Machine Learning*, 2012. [Cited on page 28.]

H. Jiang and N. Serban. Clustering Random Curves Under Spatial Interdependence With Application to Service Accessibility. *Technometrics*, 54(2):108–119, May 2012. ISSN 0040-1706, 1537-2723. doi: 10.1080/00401706.2012.657106. [Cited on page 47.]

K. Johnston, N. Wattie, J. Schorer, and J. Baker. Talent Identification in Sport: A Systematic Review. *Sports Medicine*, 48(1):97–109, Jan. 2018. ISSN 1179-2035. doi: 10.1007/s40279-017-0803-2. [Cited on page 2.]

Z. Kang, K. Grauman, and F. Sha. Learning with Whom to Share in Multi-task Feature Learning. In *ICML*, volume 2, page 4, 2011. [Cited on page 28.]

K. Karhunen. *Über Lineare Methoden in Der Wahrscheinlichkeitsrechnung*, volume 37. Sana, 1947. [Cited on page 6.]

P. E. Kearney and P. R. Hayes. Excelling at youth level in competitive track and field athletics is not a prerequisite for later success. *Journal of Sports Sciences*, pages 1–8, Apr. 2018. ISSN 0264-0414, 1466-447X. doi: 10.1080/02640414.2018.1465724. [Cited on pages x, 2, 32, and 55.]

P. Latouche, E. Birmelé, and C. Ambroise. Variational Bayesian inference and complexity control for stochastic block models. *Statistical Modelling*, 12(1):93–115, Feb. 2012. ISSN 1471-082X. doi: 10.1177/1471082X1001200105. [Cited on page 27.]

A. Leroy, A. Marc, O. Dupas, J. L. Rey, and S. Gey. Functional Data Analysis in Sport Science: Example of Swimmers' Progression Curves Clustering. *Applied Sciences*, 8(10): 1766, Oct. 2018. doi: 10.3390/app8101766. [Cited on pages xiv, 40, and 45.]

A. Leroy, P. Latouche, B. Guedj, and S. Gey. Cluster-Specific Predictions with Multi-Task Gaussian Processes. *PREPRINT arXiv:2011.07866 [cs, LG]*, Nov. 2020a. [Cited on pages xiv, 40, and 90.]

A. Leroy, P. Latouche, B. Guedj, and S. Gey. MAGMA: Inference and Prediction with Multi-Task Gaussian Processes. *PREPRINT arXiv:2007.10731 [cs, stat]*, July 2020b. [Cited on pages xiv, 40, 62, and 118.]

L. Li, B. Guedj, and S. Loustau. A quasi-Bayesian perspective to online clustering. *Electronic Journal of Statistics*, 12(2):3071–3113, 2018. ISSN 1935-7524. doi: 10.1214/18-EJS1479. [Cited on page 10.]

D. Liebl. Modeling and forecasting electricity spot prices: A functional data perspective. *The Annals of Applied Statistics*, 7(3):1562–1592, Sept. 2013. ISSN 1932-6157. doi: 10.1214/13-AOAS652. [Cited on page 4.]

D. Liebl, S. Willwacher, J. Hamill, and G.-P. Brüggemann. Ankle plantarflexion strength in rearfoot and forefoot runners: A novel clusteranalytic approach. *Human Movement Science*, 35:104–120, June 2014. ISSN 0167-9457. doi: 10.1016/j.humov.2014.03.008. [Cited on page 3.]

D. S. Lima-Borges, P. F. Martinez, L. C. M. Vanderlei, F. S. S. Barbosa, and S. A. Oliveira-Junior. Autonomic modulations of heart rate variability are associated with sports injury incidence in sprint swimmers. *The Physician and Sportsmedicine*, pages 1–11, Mar. 2018. ISSN 2326-3660. doi: 10.1080/00913847.2018.1450606. [Cited on page 2.]

R. J. A. Little and D. B. Rubin. *Statistical Analysis with Missing Data*. John Wiley & Sons, Apr. 2019. ISBN 978-0-470-52679-8. [Cited on page 22.]

H. Liu, J. Cai, and Y.-S. Ong. Remarks on multi-output Gaussian process regression. *Knowledge-Based Systems*, 144:102–121, Mar. 2018. ISSN 0950-7051. doi: 10.1016/j.knosys.2017.12.034. [Cited on page 20.]

M. Loève. Functions aleatoire de second ordre. *Revue science, 84, 195-206.*, 1946. [Cited on page 6.]

F. Mallor, T. Leon, M. Gaston, and M. Izquierdo. Changes in power curve shapes as an indicator of fatigue during dynamic contractions. *Journal of Biomechanics*, 43(8): 1627–1631, May 2010. ISSN 1873-2380. doi: 10.1016/j.jbiomech.2010.01.038. [Cited on page 3.]

F. Martínez-Álvarez, A. Schmutz, G. Asencio-Cortés, and J. Jacques. A Novel Hybrid Algorithm to Forecast Functional Time Series Based on Pattern Sequence Similarity with Application to Electricity Demand. *Energies*, 12(1):94, Jan. 2019. doi: 10.3390/en12010094. [Cited on page 53.]

B. Matérn. *Spatial Variation*, volume 36. Springer Science & Business Media, 2013. [Cited on page 14.]

G. Matheron. The intrinsic random functions and their applications. *Advances in Applied Probability*, 5(3):439–468, Dec. 1973. ISSN 0001-8678, 1475-6064. doi: 10.2307/1425829. [Cited on page 14.]

A. Maurer, M. Pontil, and B. Romera-Paredes. Sparse coding for multitask and transfer learning. In *International Conference on Machine Learning*, pages 343–351, 2013. [Cited on page 27.]

G. J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. John Wiley & Sons, Nov. 2007. ISBN 978-0-470-19160-6. [Cited on pages 21 and 68.]

G. J. McLachlan and D. Peel. *Finite Mixture Models*. John Wiley & Sons, Mar. 2004. ISBN 978-0-471-65406-3. [Cited on page 24.]

J. Mercer and A. R. Forsyth. Functions of positive and negative type, and their connection the theory of integral equations. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 209(441-458): 415–446, Jan. 1909. doi: 10.1098/rsta.1909.0016. [Cited on page 12.]

B. Minasny and A. B. McBratney. The Matérn function as a general model for soil variograms. *Geoderma*, 128(3):192–207, Oct. 2005. ISSN 0016-7061. doi: 10.1016/j.geoderma.2005.04.003. [Cited on page 14.]

K. Moesch, A.-M. Elbe, M.-L. T. Hauge, and J. M. Wikman. Late specialization: The key to success in centimeters, grams, or seconds (cgs) sports. *Scandinavian Journal of Medicine & Science in Sports*, 21(6):e282–290, Dec. 2011. ISSN 1600-0838. doi: 10.1111/j.1600-0838.2010.01280.x. [Cited on page 2.]

H. Mohamed, R. Vaeyens, S. Matthys, M. Multael, J. Lefevre, M. Lenoir, and R. Philippaerts. Anthropometric and performance measures for the development of a talent detection and identification model in youth handball. *Journal of Sports Sciences*, 27(3): 257–266, Feb. 2009. ISSN 0264-0414. doi: 10.1080/02640410802482417. [Cited on page 2.]

J. L. Morales and J. Nocedal. Remark on algorithm L-BFGS-B: Fortran subroutines for large-scale bound constrained optimization. *ACM Transactions on Mathematical Software*, 38(1):7:1–7:4, Dec. 2011. ISSN 0098-3500. doi: 10.1145/2049662.2049669. [Cited on page 98.]

P. Moreno-Muñoz, A. Artés-Rodríguez, and M. A. Álvarez. Continual Multi-task Gaussian Processes. *arXiv:1911.00002 [cs, stat]*, 2019. [Cited on pages 20 and 62.]

I. Moussa, A. Leroy, G. Sauliere, J. Schipman, J.-F. Toussaint, and A. Sedeaud. Robust Exponential Decreasing Index (REDI): Adaptive and robust method for computing cumulated workload. *BMJ Open Sport & Exercise Medicine*, 5(1):e000573, Oct. 2019. ISSN 2055-7647. doi: 10.1136/bmjsem-2019-000573. [Cited on pages xv and 40.]

R. M. Neal. Monte Carlo Implementation of Gaussian Process Models for Bayesian Regression and Classification. *arXiv:physics/9701026*, Jan. 1997. [Cited on page 20.]

J. Nocedal. Updating quasi-Newton matrices with limited storage. *Mathematics of Computation*, 35(151):773–782, 1980. ISSN 0025-5718, 1088-6842. doi: 10.1090/S0025-5718-1980-0572855-7. [Cited on pages 17 and 98.]

A. O'Hagan. Curve Fitting and Optimal Design for Prediction. *Journal of the Royal Statistical Society. Series B (Methodological)*, 40(1):1–42, 1978. ISSN 0035-9246. [Cited on page 15.]

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, Sept. 2008. ISSN 1932-6157. doi: 10.1214/08-AOAS172. [Cited on page 46.]

R. Pla, A. Leroy, R. Massal, M. Bellami, F. Kaillani, P. Hellard, J.-F. Toussaint, and A. Sedeaud. Bayesian approach to quantify morphological impact on performance in international elite freestyle swimming. *BMJ Open Sport & Exercise Medicine*, 5(1):e000543, Oct. 2019. ISSN 2055-7647. doi: 10.1136/bmjsem-2019-000543. [Cited on pages xv, 2, and 40.]

J. Quiñonero-Candela and C. E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6(Dec):1939–1959, 2005. [Cited on page 19.]

J. Quiñonero-Candela, C. E. Rasmussen, and C. K. I. Williams. *Approximation Methods for Gaussian Process Regression*. MIT Press, 2007. ISBN 978-0-262-02625-3. [Cited on page 62.]

B. Rakitsch, C. Lippert, K. Borgwardt, and O. Stegle. It is all in the noise: Efficient multi-task Gaussian process inference with structured residuals. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 1466–1474. Curran Associates, Inc., 2013. [Cited on pages 29 and 62.]

J. O. Ramsay and C. J. Dalzell. Some Tools for Functional Data Analysis. *Journal of the Royal Statistical Society. Series B (Methodological)*, 53(3):539–572, 1991. ISSN 0035-9246. [Cited on page 3.]

J. O. Ramsay and B. W. Silverman. *Applied Functional Data Analysis: Methods and Case Studies*, volume 77. Springer New York, 2002. [Cited on pages 5 and 6.]

J. O. Ramsay and B. W. Silverman. *Functional Data Analysis*. Springer, 2005. [Cited on pages 5, 7, 31, and 63.]

J. O. Ramsay, G. Hooker, and S. Graves. *Functional Data Analysis with R and MATLAB*. Use R! Springer-Verlag, New York, 2009. ISBN 978-0-387-98184-0. [Cited on page 5.]

W. M. Rand. Objective Criteria for the Evaluation of Clustering Methods. *Journal of the American Statistical Association*, 66(336):846–850, Dec. 1971. ISSN 0162-1459. doi: 10.1080/01621459.1971.10482356. [Cited on pages 47 and 52.]

C. R. Rao. Some statistical methods for comparison of growth curves. *Biometrics*, 14:1–17, 1958. ISSN 1541-0420(Electronic),0006-341X(Print). doi: 10.2307/2527726. [Cited on page 6.]

C. E. Rasmussen and Z. Ghahramani. Infinite Mixtures of Gaussian Process Experts. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 881–888. MIT Press, 2002. [Cited on page 20.]

C. E. Rasmussen and H. Nickisch. Gaussian processes for machine learning (GPML) toolbox. *The Journal of Machine Learning Research*, 11:3011–3015, 2010. [Cited on page 20.]

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, Mass, 2006. ISBN 978-0-262-18253-9. [Cited on pages 11, 14, 18, 19, 62, 65, 66, 73, 83, 94, 95, and 104.]

J. A. Rice and B. W. Silverman. Estimating the Mean and Covariance Structure Nonparametrically When the Data are Curves. *Journal of the Royal Statistical Society. Series B (Methodological)*, 53(1):233–243, 1991. ISSN 0035-9246. [Cited on pages 11 and 63.]

D. J. Rogers and T. T. Tanimoto. A Computer Program for Classifying Plants. *Science*, 132(3434):1115–1118, Oct. 1960. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.132.3434.1115. [Cited on page 52.]

P. J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, Nov. 1987. ISSN 0377-0427. doi: 10.1016/0377-0427(87)90125-7. [Cited on page 52.]

A. Schmutz, J. Jacques, C. Bouveyron, L. Cheze, and P. Martin. Clustering multivariate functional data in group-specific functional subspaces. *HAL*, 2018. [Cited on pages xi, 10, 34, 53, and 110.]

A. Schwaighofer, V. Tresp, and K. Yu. Learning Gaussian Process Kernels via Hierarchical Bayes. *NIPS*, page 8, 2004. [Cited on page 62.]

A. Schwaighofer, V. Tresp, and K. Yu. Learning Gaussian Process Kernels via Hierarchical Bayes. In L. K. Saul, Y. Weiss, and L. Bottou, editors, *Advances in Neural Information Processing Systems 17*, pages 1209–1216. MIT Press, 2005. [Cited on page 29.]

G. Schwarz. Estimating the Dimension of a Model. *Annals of Statistics*, 6(2):461–464, Mar. 1978. ISSN 0090-5364, 2168-8966. doi: 10.1214/aos/1176344136. [Cited on pages 10, 24, 53, and 116.]

M. Seeger, C. Williams, and N. Lawrence. Fast forward selection to speed up sparse Gaussian process regression. *Ninth International Workshop on Artificial Intelligence.*, 2003. [Cited on page 19.]

J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge university press, 2004. [Cited on page 13.]

M. Shen, H. Tan, S. Zhou, G. N. Smith, M. C. Walker, and S. W. Wen. Trajectory of blood pressure change during pregnancy and the role of pre-gravid blood pressure: A functional data analysis approach. *Scientific Reports*, 7(1):6227, July 2017. ISSN 2045-2322. doi: 10.1038/s41598-017-06606-0. [Cited on page 4.]

J. Shi, R. Murray-Smith, and D. Titterington. Hierarchical Gaussian process mixtures for regression. *Statistics and Computing*, 15(1):31–41, 2005. ISSN 0960-3174, 1573-1375. doi: 10.1007/s11222-005-4787-7. [Cited on pages 31 and 62.]

J. Q. Shi and Y. Cheng. Gaussian Process Function Data Analysis R Package 'GPFDA'. *Manual of the GPFDA Package*, page 33, 2014. [Cited on pages 69 and 77.]

J. Q. Shi and T. Choi. *Gaussian Process Regression Analysis for Functional Data*. CRC Press, 2011. ISBN 978-1-4398-3773-3. [Cited on pages 31, 37, 42, 69, 74, and 83.]

J. Q. Shi and B. Wang. Curve prediction and clustering with mixtures of Gaussian process functional regression models. *Statistics and Computing*, 18(3):267–283, 2008. ISSN 0960-3174, 1573-1375. doi: 10.1007/s11222-008-9055-1. [Cited on pages 20 and 31.]

J. Q. Shi, B. Wang, R. Murray-Smith, and D. M. Titterington. Gaussian Process Functional Regression Modeling for Batch Data. *Biometrics*, 63(3):714–723, 2007. ISSN 1541-0420. doi: 10.1111/j.1541-0420.2007.00758.x. [Cited on pages 30, 31, 42, 69, and 74.]

E. Snelson and Z. Ghahramani. Sparse Gaussian Processes using Pseudo-inputs. *NIPS*, page 8, 2006. [Cited on pages 20, 42, and 62.]

M. L. Stein. *Interpolation of Spatial Data: Some Theory for Kriging*. Springer Series in Statistics. Springer-Verlag, New York, 1999. ISBN 978-0-387-98629-6. doi: 10.1007/978-1-4612-1494-6. [Cited on page 13.]

K. Swersky, J. Snoek, and R. P. Adams. Multi-Task Bayesian Optimization. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2004–2012. Curran Associates, Inc., 2013. [Cited on pages 30 and 62.]

Y. W. Teh, M. Seeger, and M. I. Jordan. Semiparametric Latent Factor Models. *AISTATS 2005*, page 8, 2005. [Cited on page 29.]

P. D. Thompson. Optimum Smoothing of Two-Dimensional Fields1. *Tellus*, 8(3):384–393, 1956. ISSN 2153-3490. doi: 10.1111/j.2153-3490.1956.tb01236.x. [Cited on page 14.]

W. K. Thompson and O. Rosen. A Bayesian Model for Sparse Functional Data. *Biometrics*, 64(1):54–63, 2008. ISSN 1541-0420. doi: 10.1111/j.1541-0420.2007.00829.x. [Cited on pages 11 and 63.]

S. Thrun and J. O'Sullivan. Discovering structure in multiple learning tasks: The TC algorithm. In *ICML*, volume 96, pages 489–497, 1996. [Cited on page 28.]

M. K. Titsias. Variational Learning of Inducing Variables in Sparse Gaussian Processes. *AISTATS*, page 8, 2009. [Cited on pages 20, 42, 62, 95, and 116.]

V. Tresp. Mixtures of Gaussian Processes. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 654–660. MIT Press, 2001. [Cited on page 20.]

L. R. Tucker. Determination of parameters of a functional relation by factor analysis. *Psychometrika*, 23(1):19–23, Mar. 1958. ISSN 1860-0980. doi: 10.1007/BF02288975. [Cited on page 6.]

N. Ueda and R. Nakano. Deterministic annealing EM algorithm. *Neural Networks*, 11(2): 271–282, Mar. 1998. ISSN 0893-6080. doi: 10.1016/S0893-6080(97)00133-0. [Cited on pages 22 and 99.]

R. Vaeyens, M. Lenoir, A. M. Williams, and R. M. Philippaerts. Talent Identification and Development Programmes in Sport. *Sports Medicine*, 38(9):703–714, Sept. 2008. ISSN 1179-2035. doi: 10.2165/00007256-200838090-00001. [Cited on page 2.]

R. Vaeyens, A. Güllich, C. R. Warr, and R. Philippaerts. Talent identification and promotion programmes of Olympic athletes. *Journal of Sports Sciences*, 27(13):1367–1380, Nov. 2009. ISSN 0264-0414. doi: 10.1080/02640410903110974. [Cited on page 2.]

J. Vanhatalo, J. Riihimäki, J. Hartikainen, P. Jylänki, V. Tolvanen, and A. Vehtari. GPstuff: Bayesian Modeling with Gaussian Processes. *Journal of Machine Learning Research*, 14 (Apr):1175–1179, 2013. ISSN ISSN 1533-7928. [Cited on page 20.]

V. M. Velasco Herrera, W. Soon, G. Velasco Herrera, R. Traversi, and K. Horiuchi. Generalization of the cross-wavelet function. *New Astronomy*, 56:86–93, Oct. 2017. ISSN 1384-1076. doi: 10.1016/j.newast.2017.04.012. [Cited on page 4.]

U. von Luxburg, R. C. Williamson, and I. Guyon. Clustering: Science or Art? *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, pages 65–79, June 2012. [Cited on page 55.]

B. Wang and J. Q. Shi. Generalized Gaussian Process Regression Model for Non-Gaussian Functional Data. *arXiv:1401.8189 [stat]*, Jan. 2014. [Cited on page 31.]

T. Warren Liao. Clustering of time series data—a survey. *Pattern Recognition*, 38(11): 1857–1874, Nov. 2005. ISSN 0031-3203. doi: 10.1016/j.patcog.2005.01.025. [Cited on page 3.]

N. Wattie, J. Schorer, and J. Baker. The relative age effect in sport: A developmental systems model. *Sports Medicine (Auckland, N.Z.)*, 45(1):83–94, Jan. 2015. ISSN 1179-2035. doi: 10.1007/s40279-014-0248-9. [Cited on pages 2 and 3.]

N. Wiener. *Extrapolation, Interpolation, and Smoothing of Stationary Time Series, with Engineering Applications*. MIT Press, Cambridge, 1949. ISBN 978-1-61427-517-6. [Cited on page 14.]

C. Williams, S. Klanke, S. Vijayakumar, and K. M. Chai. Multi-task Gaussian Process Learning of Robot Inverse Dynamics. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 265–272. Curran Associates, Inc., 2009. [Cited on pages 28 and 62.]

C. K. I. Williams and C. E. Rasmussen. Gaussian Processes for Regression. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8*, pages 514–520. MIT Press, 1996. [Cited on page 15.]

J. T. Wilson, V. Borovitskiy, A. Terenin, P. Mostowsky, and M. P. Deisenroth. Efficiently sampling functions from Gaussian process posteriors. *arXiv:2002.09309 [cs, stat]*, Feb. 2020. [Cited on page 62.]

H. O. A. Wold. Estimation of principal components and related models by iterative least squares. *Multivariate analysis*, 1966. [Cited on page 7.]

C. F. J. Wu. On the Convergence Properties of the EM Algorithm. *The Annals of Statistics*, 11(1):95–103, 1983. ISSN 0090-5364. [Cited on page 22.]

J. Yang, H. Zhu, T. Choi, and D. D. Cox. Smoothing and Mean–Covariance Estimation of Functional Data with a Bayesian Hierarchical Model. *Bayesian Analysis*, 11(3):649–670, 2016. ISSN 1936-0975, 1931-6690. doi: 10.1214/15-BA967. [Cited on pages 14, 30, 31, 69, and 95.]

J. Yang, D. D. Cox, J. S. Lee, P. Ren, and T. Choi. Efficient Bayesian hierarchical functional data analysis with basis function approximations using Gaussian-Wishart processes. *Biometrics*, 73(4):1082–1091, 2017. ISSN 0006-341X. doi: 10.1111/biom.12705. [Cited on pages 30, 31, and 69.]

K. Yu, V. Tresp, and A. Schwaighofer. Learning Gaussian Processes from Multiple Tasks. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 1012–1019, New York, NY, USA, 2005. ACM. ISBN 978-1-59593-180-1. doi: 10.1145/1102351.1102479. [Cited on pages 30 and 62.]

S. Yu, K. Yu, V. Tresp, and H.-P. Kriegel. Collaborative ordinal regression. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, pages 1089–1096, New York, NY, USA, June 2006. Association for Computing Machinery. ISBN 978-1-59593-383-6. doi: 10.1145/1143844.1143981. [Cited on page 30.]

H. Zhang. Maximum-likelihood estimation for multivariate spatial linear coregionalization models. *Environmetrics*, 18(2):125–139, 2007. [Cited on page 30.]

Y. Zhang and Q. Yang. A Survey on Multi-Task Learning. *arXiv:1707.08114 [cs]*, July 2018. [Cited on page 27.]

Y. Zhang and D.-Y. Yeung. A Convex Formulation for Learning Task Relationships in Multi-Task Learning. *arXiv:1203.3536 [cs, stat]*, Mar. 2012. [Cited on page 28.]

J. Zhu and S. Sun. Multi-task Sparse Gaussian Processes with Improved Multi-task Sparsity Regularization. In *Pattern Recognition*, pages 54–62. Springer, Berlin, Heidelberg, 2014. doi: 10.1007/978-3-662-45646-0_6. [Cited on pages 30 and 62.]