Méthodes statistiques pour données fonctionnelles multivariées

Steven Golovkine

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

ENSAI École nationale de la statistique et de l'analyse de l'information

THÈSE DE DOCTORAT DE

L'Ecole Nationale de la Statistique
et de l'Analyse de l'Information

ÉCOLE DOCTORALE N° 601
Mathématiques et Sciences et Technologies de l'Information et de la Communication
Spécialité : Mathématiques et leurs Interactions

Par

Steven GOLOVKINE

Statistical methods for multivariate functional data

Thèse présentée et soutenue à Bruz, le 18/06/2021
Unité de recherche : CREST

Rapporteurs avant soutenance :

Sophie DABO-NIANG	Professeur, Université de Lille
Alois KNEIP	Professeur, Université de Bonn

Composition du Jury :

Président :	André MAS	Professeur, Université de Montpellier
Examinateurs :	Sophie DABO-NIANG	Professeur, Université de Lille
	Vincent FEUILLARD	Expert Statistique Renault, Technocentre, Guyancourt
	Claire GORMLEY	Professeur, University College Dublin
	Alois KNEIP	Professeur, Université de Bonn
Dir. de thèse :	Valentin PATILEA	Professeur, CREST, ENSAI
Co-dir. de thèse :	Nicolas KLUTCHNIKOFF	Maître de conférences, IRMAR, Université Rennes 2

Remerciements

Au début de ma thèse, on m'a dit qu'il y avait des similarités entre les travaux de thèse et un marathon. N'étant pas un grand fan de course à pied, je ne saurais trop dire. Par contre, je peux faire un parallèle avec un match de basket. Le doctorat est, d'abord, un travail d'équipe et non pas un effort solitaire. Ensuite, la soutenance se dit "defense" en anglais. Il doit bien y avoir une raison ! Bien entendu une passe décisive dans le monde académique correspond bien plus souvent à une discussion autour d'un café qu'à la transmission d'un ballon, mais la finalité est la même. Enfin, je pense, et j'espère, que chaque chercheur a ressenti un phénomène bien connu des basketteurs, "avoir la main chaude". Bon, pour un basketteur, ça correspond à enchainer les paniers, et pour un chercheur, ce serait plutôt l'enchainement des calculs ou bien la fluidité d'écriture du code. Ainsi, c'est avec plaisir que je tiens à remercier toutes les personnes qui ont pu faire parti de l'équipe.

En premier lieu, je voudrais remercier mes directeurs de thèse, Valentin Patilea et Nicolas Klutchnikoff, pour avoir tenu le rôle de coachs. Merci pour votre patience, votre confiance et votre soutien pendant ces trois années et quelques mois. Et comme tout bon coachs, ils m'ont transmis toute leur passion et leur motivation vis à vis du sujet et de leur métier. Leur sens de la formule n'ayant rien à envier aux plus grands entraineurs, je me permets de les citer ici. Par une chaude journée de février 2019 en faisant les cents pas, Valentin m'expliquait comment trouver l'inspiration : "[Le] processus de création nécessite un déplacement continu". Et alors que l'on s'intéressait à un article lors d'un pluvieux jour de juillet 2019, Nicolas me transmettait sa méthode de lecture des articles : "D'abord, tu supposes que c'est faux, et ensuite, tu vois pourquoi, éventuellement, ça pourrait être vrai... méthode russe !"

Merci à Thierry Cembrzynski, sans qui le match n'aurait pas eu lieu.

Merci à Sophie Dabo et Aloïs Kneip d'avoir accepter de rapporter ma thèse. Merci à André Mas, Claire Gormley et Vincent Feuillard de m'avoir fait l'honneur de participer au jury. Merci à Olivier Lopez et Fabien Mangeant d'avoir fait parti du comité de suivi.

Pour que le match se déroule sans accroc, nous avons aussi besoin de personnes qui comptent les points et qui gérent le temps. Merci à Cécile Terrien d'avoir joué ce rôle, probablement le plus important pour le bon déroulé de la thèse. J'ai bien dû la déranger un million de fois depuis mon master pour des problèmes administratifs qu'elle a toujours réglé avec bonne humeur.

L'avantage d'une thèse CIFRE, c'est d'avoir deux équipes très différentes, une dans l'entreprise et une au labo.

Merci à mes collègues chez Renault, Nicolas, Philippe, François, Virginie, Eric, Mathieu, Joan, Ayhan, Johanna, Patrice, Amélie, Samia, Maxime, Virginie, Nathalie et Nicolas de m'avoir soutenu pendant ces presques quatre années.

Merci à mes camarades doctorants à l'ENSAI, Amandine, Edouard, Camille, Max et Elie sans qui ces trois années n'auraient été les mêmes. En tout cas, j'espère vous avoir fait autant de passes décisives que vous avez pu m'en faire.

Merci à ma plus vieille équipe, Quentin, Valentin, Thibaut, Jordan, Antonin, Bertrand et Vivien, ainsi qu'au dernier arrivé, Erwan. Ne vous inquêtez pas les gars, un jour, je l'aurai ce put**** de rebond.

Merci à toutes les personnes que j'ai pu croiser sur un terrain de basket (un vrai, cette fois) et désolé pour mes (légères) sautes d'humeur.

Pour terminer, un match de basket ne saurait être complet sans des supporters en folie qui peuplent les tribunes. Merci à Alain pour m'avoir donné une première expérience avec Renault. Merci à mon oncle et ma tante (et mes cousins) pour m'avoir acceuilli bien trop souvent pendant mes longues années rennaises. Merci à mes grand-parents, à mes parents et à ma soeur qui, même s'ils ne me comprenaient pas toujours, m'ont toujours soutenu. En tout cas, même si le match a été long, il est bientôt fini.

3..2..1.. Fin du match.. Voilà le résultat !

Table of Contents

Acronyms xi

Introduction 1

Contexte industriel . . . 1

Différents niveaux d'autonomie . . . 3

Fiabilité et sécurité fonctionnelle . . . 4

Données de roulage . . . 5

Point de vue véhicule . . . 6

Point de vue extérieur . . . 7

Modèle d'observation . . . 9

Contributions et organisation du manuscrit . . . 11

1 Concepts of Functional Data and Clustering 15

1.1 Functional Data Analysis . . . 16

1.1.1 Univariate functional data . . . 17

1.1.2 Multivariate functional data . . . 17

1.2 From discrete data to functions . . . 18

1.2.1 Basis expansion . . . 19

1.2.2 Nonparametric estimation . . . 21

1.3 Functional Principal Component Analysis . . . 24

1.3.1 Computation of the eigenelements . . . 25

1.4 Clustering . . . 26

1.4.1 Model-based clustering . . . 28

Appendix . . . 30

1.A The EM algorithm . . . 30

2 Learning the Smoothness of Noisy Curves 33

2.1 Introduction . . . 34

2.2 Local regularity estimation . . . 37

2.2.1 The methodology . . . 37

2.2.2 Concentration bounds for the local regularity estimator 40
2.2.3 The case of smooth trajectories 43
2.2.4 The case of conditionally heteroscedastic noise 44
2.3 Adaptive optimal smoothing 46
2.3.1 Local polynomial estimation 47
2.4 Empirical analysis 50
2.4.1 Simulation experiments 52
2.4.2 Real data analysis: the NGSIM Study 55
Appendices 63
2.A Proof of Theorem 1 63
2.B Proofs of Theorems 2 and 3 69
2.C Technical lemmas 76
2.D Moment bounds for spacings 85
2.D.1 The uniform case 86
2.D.2 The general case 87
2.D.3 Wendel's type inequalities for gamma function ratios 90
2.E Additional simulation results 92
2.E.1 The settings for X 92
2.E.2 On the computation time 93
2.E.3 On the estimation of the local regularity 93
2.E.4 On the pointwise risk 96
2.F Traffic flow: Montanino and Punzo methodology 97
2.G Complements on the real-data applications 100
2.G.1 Canadian weather 100
2.G.2 Household Active Power Consumption 102
2.G.3 PPG-Dalia 103

3 Adaptive estimation of irregular mean and covariance function 105
3.1 Introduction 106
3.2 From unfeasible to feasible optimal estimators 111
3.3 Local regularity estimator 116
3.3.1 Local regularity in quadratic mean 116
3.3.2 The local regularity estimation method 117
3.3.3 Concentration properties of the local regularity estimator 119

3.3.4 From process regularity to trajectories regularity . . . 120
3.4 Optimal mean and covariance estimators . . . 121
3.4.1 Adaptive mean estimation . . . 123
3.4.2 Adaptive covariance function estimates . . . 126
3.4.3 Estimator on the diagonal band of the covariance function . . . 128
3.5 Empirical study . . . 129
3.5.1 Implementation aspects . . . 129
3.5.2 Simulation design . . . 130
3.5.3 Mean estimation . . . 131
3.5.4 Covariance estimation . . . 132
3.6 Discussion and conclusions . . . 134
Appendix . . . 137
3.A Details on the definition (3.21) . . . 137
3.B Proofs . . . 139
3.C Additional simulation results . . . 143
3.C.1 Description of the real data set used to build the simulations . . . 143
3.C.2 Construction of the simulation design . . . 143

4 Clustering multivariate functional data using unsupervised binary trees 147
4.1 Introduction . . . 148
4.2 Model and methodology . . . 150
4.2.1 Notion of multivariate functional data . . . 150
4.2.2 A mixture model for curves . . . 151
4.2.3 Multivariate Karhunen-Loève representation . . . 154
4.3 Parameters estimation . . . 157
4.3.1 Estimation of mean and covariance . . . 158
4.3.2 Derivation of the MFPCA components . . . 159
4.4 Multivariate functional clustering . . . 160
4.4.1 Building the maximal tree . . . 160
4.4.2 Joining step . . . 164
4.4.3 Classification of new observations . . . 165
4.5 Empirical analysis . . . 166
4.5.1 Simulation experiments . . . 169
4.5.2 Real data analysis: the rounD dataset . . . 176

4.6 Extension to images . . . 182
4.7 Conclusion . . . 184
Appendix . . . 184
4.A Proofs . . . 185

5 Implementation of the methods **189**
5.1 Introduction . . . 190
5.2 Classes of functional data . . . 192
5.3 Data used in the examples . . . 194
5.4 Manipulation of functional data objects . . . 194
5.4.1 Creation of objects . . . 195
5.4.2 Access to the instance variables . . . 197
5.4.3 Plotting . . . 198
5.4.4 Data simulation . . . 198
5.5 Parameters estimation . . . 203
5.5.1 Curves denoising . . . 203
5.5.2 Mean and covariance estimation . . . 204
5.6 MFPCA . . . 206
5.6.1 Methodological background . . . 206
5.6.2 Implementation . . . 207
5.7 `fCUBT` . . . 210
5.7.1 Methodological background . . . 210
5.7.2 Implementation . . . 211
5.8 Conclusion . . . 214

Conclusion **215**

Bibliography **217**

List of Figures

1 Différents niveaux de conduite autonome . . . 3
2 ASIL pour différents ADAS . . . 5
3 Exemple de véhicule équipé . . . 6
4 Exemple de caméra fixe sur autoroute . . . 8

1.1 Examples of functional data . . . 16

2.1 Estimation of the local regularity for piecewise fBm . . . 54
2.2 Estimation of the local regularity for integrated fBm . . . 55
2.3 Estimation of different risks for piecewise fBm . . . 56
2.4 Estimation of risks for smoothing the noisy trajectories of an integrated fBm 57
2.5 Comparing risks for smoothing the noisy trajectories of a fBm . . . 57
2.6 Comparing risks for smoothing the noisy trajectories of a piecewise fBm . . 58
2.7 Comparing risks for smoothing the noisy trajectories of an integrated fBm 58
2.8 I-80 dataset illustration: a sample of five velocity curves . . . 60
2.9 I-80 dataset clusters: density of sampling points . . . 60
2.10 Estimation of the local regularity of the velocity curves for different t_0 . . . 61
2.11 Densities of the ratio within different groups . . . 62
2.12 Illustrations of simulated data generated according to the different settings 94
2.13 Computational times (log scale) . . . 95
2.14 Estimation of the local regularity for fBm . . . 95
2.15 Estimation of the local regularity for piecewise fBm . . . 96
2.16 Estimation of the risks for piecewise fBm with constant noise variance . . . 98
2.17 Estimation of the risk for smoothing the noisy trajectories of a fBm . . . 99
2.18 Estimation of the risks for piecewise fBm with non-constant noise variance 99
2.19 I-80 dataset illustration of the clusters . . . 101
2.20 Canadian weather dataset illustration . . . 102
2.21 Household active power consumption dataset illustration . . . 102
2.22 PPG-Dalia dataset illustration . . . 103

3.1 The simulation setup in *Experiment* . . . 132
3.2 Results from *Experiment* on the log-scale . . . 133
3.3 Results from *Experiment* on the log-scale . . . 135
3.4 Extracted Household Active Power Consumption data: voltage curves . . . 143

4.1 Illustration of maximal tree for Scenario 1 simulated data . . . 163
4.2 Simulated data for Scenario 1 . . . 170
4.3 Simulated data for Scenario 2 . . . 171
4.4 Simulated data for Scenario 3 . . . 172
4.5 Estimation of ARI for all tested models on 500 simulations . . . 175
4.6 Estimation of ARI for the comparison with supervised models . . . 177
4.7 Estimation of ARI when the tree is used as a supervised classifier . . . 178
4.8 rounD dataset: the considered roundabout . . . 179
4.9 rounD dataset illustration: a sample of five trajectories . . . 180
4.10 rounD dataset: an example of a cluster found using the `fCUBT` method . . . 181
4.11 rounD dataset: two different clusters with similar trajectory shape . . . 181
4.12 Examples of simulated data for Scenario 4 . . . 183

5.1 Representation of the main classes . . . 192
5.2 Results of the `plot` method for functional data object . . . 199
5.3 Links between classes in the simulation toolbox . . . 199
5.4 Example of simulated data . . . 200
5.5 Results for the `add_noise` and `sparsify` functions . . . 201
5.6 Simulation of data with two clusters . . . 203
5.7 Curve and smoothed estimation for the Canadian Temperature data . . . 204
5.8 Mean and covariance estimation for the Canadian Temperature data . . . 205
5.9 Results of the MFPCA for the Canadian Weather data . . . 208
5.10 Plot of the Canadian Temperature dataset by cluster . . . 212
5.11 Grown tree $\mathfrak{T}$ illustration for the Canadian Temperature dataset . . . 213

List of Tables

4.1 Number of clusters selected for each model 173
4.2 Coefficients for $X_1(t)$ and $X_2(s,t)$ 183
4.3 Results for the Scenario 4 184

Acronyms

A

ADAS Advanced Driver Assistance Systems 2, 4, 5

AEB Autonomous Emergency Braking 2, 4

ARI Adjusted Rand Index 168, 172, 174, 176, 182, 184

ASIL Automotive Safety Integrity Level 4, 5

B

BIC Bayesian Information Criterion 29, 60, 100, 149, 161, 162, 165, 172, 210, 211

C

CAN Controller Area Network 6, 18

CUBT Clustering using Unsupervised Binary Trees 149, 160

CV Cross-Validation 52, 53, 55, 110

E

EM Expectation-Maximisation 29–31, 60, 100, 161, 162, 164, 168

F

fBm fractional Brownian motion 53, 54, 93, 96, 97

FCP-TPA Functional Candecomp/Parafac Tensor Power Algorithm 150, 182, 209

fCUBT functional Clustering using Unsupervised Binary Trees 13, 149, 150, 160, 166, 168, 172, 174, 176, 179, 182, 184, 191, 192, 210, 211

FDA Functional Data Analysis 16, 17, 34, 35, 106, 148, 190

fLBM functional Latent Block Model 27

fMRI functional Magnetic Resonance Imaging 16

fPCA functional Principal Component Analysis 17, 20, 24, 25, 27, 148, 157, 159, 182, 206, 207

G

GAM Generalized Additive Models 209

GMM Gaussian Mixture Model 155, 156, 161, 162, 166–168, 210

GPS Global Positioning System 6

I

ICL Integrated Completed Likelihood 29

ISO Organisation Internationale de Normalisation 4

L

LBM Latent Block Model 27

LiDAR Light Detection and Ranging 7

LKA Lane-Keeping Assist 2

M

MFPCA Multivariate Functional Principal Component Analysis 13, 25, 29, 155, 162, 164, 174, 182, 188, 190–192, 206, 207, 210

N

NGSIM Next Generation Simulation 7, 8, 12, 55, 59, 96, 97, 100, 176

O

OICA Organisation Internationale des Constructeurs Automobiles 3

P

PACE Principal component Analysis through Conditional Expectation 26, 159, 209

PCA Principal Component Analysis 17, 24, 25

Introduction

Résumé: *Le sujet de cette thèse est lié à l'analyse de données fonctionnelles et est motivé par l'analyse de données provenant de l'industrie automobile. Nous commençons par présenter le contexte industriel dans lequel cette thèse s'incrit. Le problème initial était de construire des scénarios de conduite qui sont représentatifs des habitudes des conducteurs. Pour cela, nous nous sommes intéressés au problème de groupement des trajectoires de véhicules. Les données peuvent provenir de différentes sources, comme les capteurs embarqués sur la voiture, les caméras sur autoroutes ou bien des vidéos de drones,* etc.

Contents

Contexte industriel **1**
Différents niveaux d'autonomie 3
Fiabilité et sécurité fonctionnelle 4
Données de roulage **5**
Point de vue véhicule 6
Point de vue extérieur 7
Modèle d'observation **9**
Contributions et organisation du manuscrit **11**

Contexte industriel

En 2019, le nombre de morts sur les routes françaises s'est établi à 3239 décès[1]. Ce total est le plus bas depuis l'obligation du port de la ceinture de sécurité sur les places avants décrétée en 1973. Les politiques publiques, telles que les mesures de lutte contre l'alcool au volant ou encore la limitation de la vitesse à 80 km/h dans les villes, ont permis de diminuer les décès ses quarantes dernières années. Cependant, en regardant le bilan annuel de la

1. https://bit.ly/morts_routes_2019

sécurité routière de 2019, les accidents de la route causent toujours près de 70 000 blessés[2]. De la conception du véhicule à l'intégration des aides actives à leur conduite (Advanced Driver Assistance Systems (ADAS)), les constructeurs automobiles ont aussi eu un rôle important dans la diminution d'accidents de la circulation et auront une responsabilité majeure dans le futur de la mobilité. Comme principaux ADAS, nous pouvons citer le freinage automatique d'urgence (Autonomous Emergency Braking (AEB)), qui permet le freinage automatique du véhicule lorsque celui détecte un risque de colision, ou encore l'aide au maintien dans la file de circulation (Lane-Keeping Assist (LKA)). Entre autres, la Commission Européenne a rendu obligatoire ces deux ADAS dès 2021[3]. Cette mesure fait partie d'un plan à long terme de l'Union Européenne visant les zéros morts d'ici 2050 [44].

Cependant, si la voiture individuelle est l'un des symboles de progrès et de libération de l'homme moderne, une large part des accidents est malgré tout causés par des erreurs humaines ! Ainsi, le développement des ADAS a pour but de rendre négligeable l'erreur humaine. Les constructeurs imaginent même à terme des véhicules totalement autonomes où l'humain ne serait plus responsable de la conduite. Ainsi, ce qui semblait, il y a encore quelques années, totalement inconcevable est, aujourd'hui, de l'ordre du possible. Par conséquent, les enjeux sociétaux autour de cette percée technologique majeure sont multiples. Le véhicule autonome devrait donc permettre de faire régresser la mortalité routière de manière décisive mais également de faciliter le déplacement des personnes à mobilités réduites ou malvoyantes et, de manière plus générale, leur donner une plus grande autonomie. Un de leur autre objectif est de décongestionner les routes.

Pour les constructeurs automobiles, en plus de diminuer drastiquement les accidents, leur ambition est d'offrir au propriétaire du véhicule la possibilité de se réapproprier le temps actuellement consacré à la conduite. L'habitacle du véhicule sera ainsi répensé et se transformera en un espace de vie ou de travail confortable et ultramoderne. De plus, une nouvelle vision de la mobilité pourrait être créée avec, par exemple, l'instauration de flottes d'autopartage entièrement électriques et connectées.

2. https://bit.ly/bilan_securite_routiere_2019
3. https://bit.ly/commission_mobilite

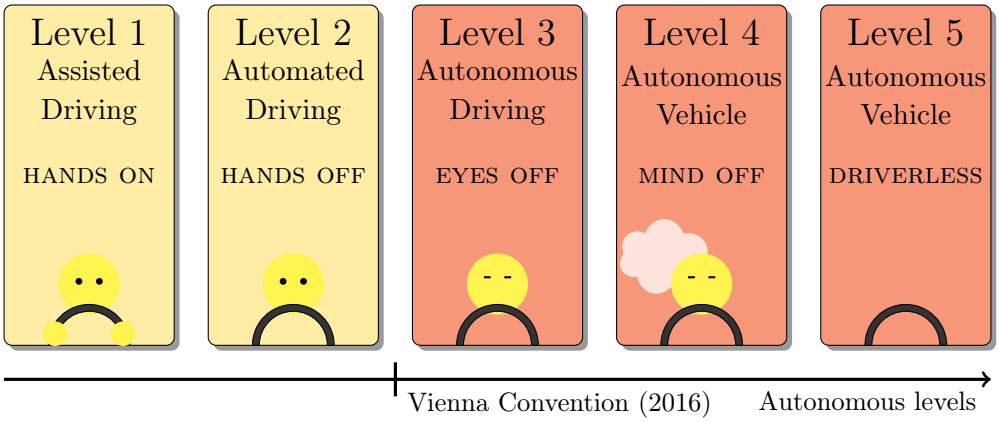


Figure 1: Différents niveaux de conduite autonome.

Différents niveaux d'autonomie

L'Organisation Internationale des Constructeurs Automobiles (OICA) définit cing degrées d'autonomie pour le véhicule autonome[4] (voir Figure 1), sans compter le niveau 0 où aucune assistance n'est disponible. Pour les deux premiers niveaux, certaines fonctions sont déléguées au système autonome, le contrôle latéral du véhicule par exemple. Cependant, le conducteur doit surveiller le bon fonctionnement du système, ainsi que l'environnement du véhicule. Ainsi, il est pleinement conscient et responsable des actions prises par la voiture et de ce qu'il se passe sur la route. En revanche, à partir du niveau 3, la question de la responsabilité du conducteur, et par conséquent de la sécurité, devient critique. Pour le mode *Eyes Off*, le conducteur n'est plus tenu de surveiller le système, mais celui-ci doit rendre la main au conducteur s'il se retrouve dans une situation qu'il ne sait pas gérer. Au dessus de ce niveau, l'automobiliste peut être passif, c'est-à-dire qu'il peut faire une autre activité que la conduite. Ainsi, le niveau 5 défini un système pouvant gérer toutes les situations rencontrées sur la route sans conducteur.

La convention de Vienne sur la circulation routière de 1968 [131] régie la législation et la réglementation routière à travers le monde. Début 2020, 83 états étaient parties contractantes de cette convention[5]. En particulier, elle précise les conditions que doivent vérifier les automobiles pour pouvoir circuler dans les pays signataires. Récemment, la législation a commencé à évoluer pour autoriser la circulation de véhicules de niveau 3 ou 4. Un amendement est ainsi proposé à la convention de Vienne en 2014 et votée en

4. https://bit.ly/autonomous_levels
5. https://bit.ly/parties_contractantes_vienne_1968 - accès 05/05/2020

2016[6], autorisant les systèmes autonomes tant que le conducteur peut, à tout moment, désactiver le système et en reprendre le contrôle et être en conformité avec les régles des Nations Unis [79].

Sécurité fonctionnelle

Pour pouvoir commercialiser un véhicule, quelque soit son niveau d'autonomie, le constructeur doit pouvoir démontrer la sécurité fonctionnelle du produit. Celle-ci permet de mesurer les risques propres au système et livrer des solutions permettant de réduire leurs occurences pour sécuriser les biens et les personnes [117]. Ainsi, l'Organisation Internationale de Normalisation (ISO) a mis en place un ensemble de normes permettant d'uniformiser les pratiques. En ce qui concerne les systèmes électriques/électroniques du véhicule, et en particulier les ADAS, la norme de référence est la norme ISO 26262 [80]. Cependant, cette norme concerne principalement les défaillances matérielles et les fautes systématiques logicielles des systèmes. Et donc, elle n'inclut pas les erreurs d'interprétation de l'environnement véhicule et les problèmes d'intéraction avec d'autres éléments internes, matériels ou logiciels. Par exemple, dans le cas du système de freinage d'urgence (AEB), un cas de défaillance serait que le système ne détecte pas de piétons, et par conséquent ne déclenche pas le freinage d'urgence, alors que le contraire aurait été nécessaire.

L'ISO 26262 définit un système de classification des risques (ASIL) permettant de caractériser le risque suivant trois critères: la sévérité, l'exposition et la contrôlabilité des situations rencontrées. Cette échelle de risques est élaboré en quatre niveaux de A à D, D étant le niveau le plus exigeant. Chacun des ADAS doit répondre à un niveau de risque différent (voir Figure 2). Dans le cas d'un véhicule autonome, le conducteur n'ayant plus à contrôler la majeure partie des fonctionnalités du système, le critère de contrôlabilité doit être maximal. Ainsi, l'ASIL pris pour référence est l'ASIL D, qui impose une probabilité limite de défaillance crtitique de l'ordre de 10^{-8}/h, soit une panne grave toutes les cent millions d'heures de conduite.

Actuellement, des véhicules tests sont équipés et roulent des millions de kilomètres en conditions réelles pour valider la sécurité des ADAS et satisfaire aux exigences de l'ISO [107]. Dans le cas de systèmes totalement autonomes, le nombre de kilomètres serait

6. http://bit.ly/UNECE_modif_vienne
7. Source: https://bit.ly/asil_adas

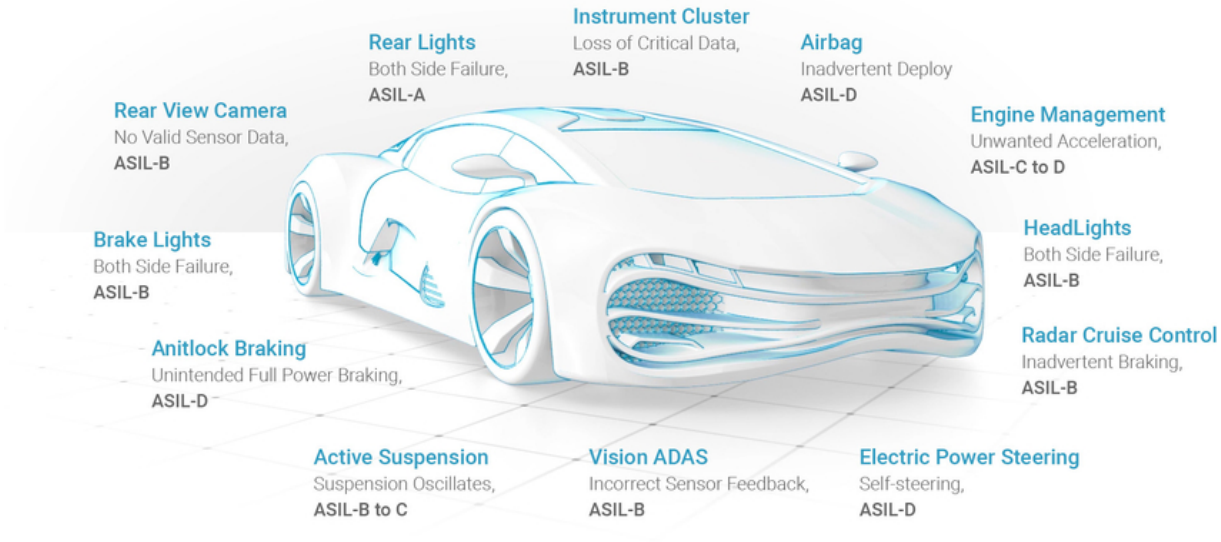


Figure 2: Automotive Safety Integrity Level (ASIL) pour différents ADAS.[7]

trop important pour que cela soit réalisable. Raffaelli et al. [107] proposent notamment une approche statistique associée à des simulations pour réduire le nombre de kilomètres physiquement fait. Celle-ci se base sur la génération de « cas tests ». Un besoin similaire a été exprimé par Cherfi et al. [32] pour l'identification de scénarios de roulages représentatifs du comportement humain.

Ces travaux de thèse s'inscrivent donc dans ce contexte industriel en visant entre autre à proposer une méthode de création de bases de scénarios de roulages représentatifs en s'appuyant sur l'analyse statistique de bases de roulages réels. Ces bases pourront ensuite être intégrées dans les moteurs de simulation.

Données de roulage

Nous définissons les données de roulage comme étant un ensemble de caractéristiques représentatives du traffic routier et des interactions entre les usagers de la route évoluant pour un temps et/ou un espace donné.

Un scénario de conduite peut être décrit dans différents référentiels. Premièrement, nous observons la scène de l'intérieur du véhicule. Ainsi, nous nous situons dans le référentiel véhicule et donc la trajectoire des objets est enregistrée par rapport à celui-ci. Deuxièmement, nous considérons un observateur extérieur à la scène. Dans ce cas là, les tra-

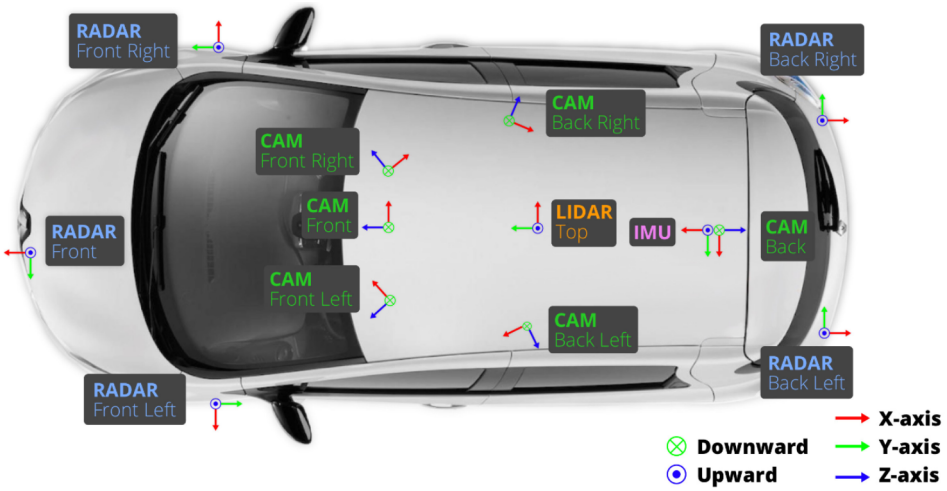


Figure 3: Exemple de véhicule équipé.[8]

jectoires sont enregistrées par rapport à un point fixe de la scène.

Les signaux mesurés auxquels nous nous intéressons particulièrement dans cette thèse correspondent aux positions longitudinale x et latérale y, aux vitesses longitudinale v_x et latérale v_y et aux accélérations longitudinale a_x et latérale a_y de tous les véhicules présents dans la scène. Ces différentes mesures ne sont pas évaluées de la même manière suivant le point de vue considéré. Les différents signaux enregistrés sont généralement disponibles à une fréquence de 20 Hz.

Point de vue véhicule

Pour construire des jeux de données de roulages d'un point de vue véhicule, des voitures sont spécialement équipées pour rouler des milliers de kilomètres et enregistrer les paramètres intrinsèques du véhicule et de son environnement (voir Figure 3 pour l'instrumentation véhicule pour les données *nuScenes*).

Les variables véhicules sont pour la plupart mesurées par le bus de données CAN (Controller Area Network). Cependant, certains capteurs peuvent être ajoutés en fonction des données voulues. Ainsi, un odomètre combiné à un GPS (Global Positioning System) permet de connaître notre position. Les avantages et inconvénients de ces deux systèmes se contrebalançant, ils permettent d'avoir une précision correcte accompagnée d'une faible erreur de mesure. Généralement, notre vitesse est directement lue sur le bus CAN. Sa

8. Source: *nuScenes dataset* Caesar et al. [17]

précision est dépendante des choix du constructeur, qui ne sont pas public.

Concernant la position et la vitesse des véhicules environnants, des caméras, des radars et un scanner LiDAR (Light Detection and Ranging) sont utilisés. L'estimation de la position et de la vitesse des objets basé sur l'utilisation de vidéos extraites de caméras, les images haute-résolutions des radars ou les nuages de points 3D issus du LiDAR est un vaste sujet de recherche dérivé des recherches sur la vision par ordinateur. Elle se décompose en deux étapes : la détection d'objets dans une image, appeler segmentation sémantique, et le suivi de ces objets à travers les différentes images des vidéos. L'état de l'art sur la détection d'objet dans une scène se base sur des architectures de réseaux de neurones convolutifs [90]. Aujourd'hui, l'architecture donnant les meilleurs résultats dans le domaine automobile est le *Mask R-CNN* [70]. Le *tracking* se fait en calculant les distances entre tous les objets de deux images consécutives.

À titre d'exemples, nous pouvons citer les jeux de données suivant :

- Le *KITTI dataset* [52], un des premiers jeu de données *open-source* relatif aux scénarios de conduite d'un point de vue véhicule, est un projet commun de *Karlsruher Institut für Technologie* et de *Toyota Technological Institute at Chicago*. Il contient environ 1 h30 min de roulages dans les environs de Karlsruhe en Allemagne. Il est utilisé comme *benchmark* dans la plupart des études traitant de la vision par ordinateur en automobile.

- Le jeu de données *nuScenes* [17] a été construit à Boston et Singapore, deux villes connues pour leur traffic dense, par l'entreprise privé *Aptiv Autonomous Mobility*[9]. Il inclut les enregistrements des différents capteurs de 5 h30 min de conduite.

- Le *Waymo dataset* [127] est un enregistrement de 6 h30 min de routes californiennes. Il a été développé par l'entreprise privée *Waymo*[10], filiale de Google. Il se distingue par son très grand nombre d'objet annotés comparé aux autres *datasets*.

Point de vue extérieur

Les données d'un point de vue extérieur à la scène sont enregistrés grâce à l'installation de caméras fixes sur différentes routes (voir Figure 4 pour les données *NGSIM*). La détection et le tracking des objets se fait de la même manière que pour le point de vue véhicule,

9. https://www.aptiv.com/
10. https://waymo.com/

Figure 4: Exemple de caméra fixe sur autoroute.[11]

autrement dit, par réseaux de neurones convolutifs pour la segmentation et calculs de distance pour le tracking. La qualité et la précision des résultats dépendent grandement de la caméra, ainsi que de l'angle de celle-ci avec la route.

À titre d'exemples, nous pouvons mentionner les jeux de données suivant :

- Le *NGSIM dataset* [47], fourni par le département des transports américain, est une référence dans ce domaine. Il est composé de quatre enregistrements d'autoroutes américaines. Généralement, les algorithmes de prédiction de trajectoire de véhicule sont étalonnés sur celui-ci. Cependant, Punzo et al. [106] ont montré des incohérences de mesures dans les données brutes. Ainsi, ces données ne devraient pas être utilisées sans un traitement préalable pour diminuer ces erreurs. Nous reviendrons dessus comme cas d'application lors du Chapitre 2.

- Les *highD, rounD* et *inD datasets* [89, 8] sont des données mises à disposition par une université allemande, l'*Institute for Automotive Engineering (ika)* de *RWTH Aachen University*, pour permettre la recherche sur la conduite automobile. Ces données sont récoltées grâce à des drones en stationnement au dessus de routes allemandes ayant des caractéristiques particulières (autoroutes, intersections, ronds-points). En particulier, nous utiliserons *rounD* comme jeu de données réelles sur lequel notre algorithm de *clustering* sera applqué lors du Chapitre 4.

11. Source: *NGSIM dataset* FHWA, U.S. Department of Transportation [47]

Modèle d'observation

Nous pouvons définir un cadre commun pour les deux sources de données et donc les deux points de vue. Ce cadre sera celui de l'analyse de données fonctionnelles. En effet, l'évolution des caractéristiques de voitures au cours du temps s'apparentent à des courbes, et théoriquement, l'évolution de ces caractéristiques est continue. Enfin, en considérant, le début de l'observation comme étant le premier instant où l'objet est détecté et la fin de l'observation comme étant le dernier instant où l'objet est détecté, nous obtenons des données fonctionnelles. Cette approche nous permet de lier les deux sources de données et de pouvoir y appliquer des méthodes similaires.

Pour formaliser le cadre, considérons $I \subset \mathbb{R}$ un ensemble compact de $\mathbb{R}$. Soit N fonctions $X^{(1)}, \ldots, X^{(n)}, \ldots, X^{(N)}$ un échantillon aléatoire d'un processus stochastique $X = (X_t : t \in I)$ ayant des trajectoires continues. Pour chaque $1 \leq n \leq N$ et un entier positif M_n, notons $T_m^{(n)}$, $1 \leq m \leq M_n$, les temps aléatoires d'observation pour la courbe $X^{(n)}$. Ceux-ci sont obtenus comme des réalisations indépendantes d'une variable aléatoire T prenant valeur dans I. Les entiers $M_1, \ldots, M_N$ sont des réalisations indépendantes d'une variable aléatoire M prenant valeur dans $\mathbb{N}^\star$ avec pour moyenne μ croissant avec N. De plus, nous faisons l'hypothèse que les réalisations de X, M et T sont mutuellement indépendants. Ainsi, pour chaque courbe, qu'on l'on nomme aussi trajectoire, $X^{(n)}$, nous observons les paires $(Y_m^{(n)}, T_m^{(n)}) \in \mathbb{R} \times I$ tel que $Y_m^{(n)}$ soit définit comme

$$Y_m^{(n)} = X^{(n)}(T_m^{(n)}) + \varepsilon_m^{(n)}, \qquad 1 \leq n \leq N, \;\; 1 \leq m \leq M_n, \tag{1}$$

où les $\varepsilon_m^{(n)}$ sont des réalisations indépendante d'une variable aléatoire ε. À moins que ce ne soit précisé autrement, ε est supposée être un bruit gaussien centré et de variance inconnue σ^2. Ce modèle peut s'étendre aux processus stochastiques multidimensionnels en considérant I comme un ensemble compact de $\mathbb{R}^d, d \in \mathbb{N}^\star$, et en faisant les modifications nécesaires pour les variables aléatoires T et M. Dans le cas particulier où $d = 2$, nous observerons donc des images (ou surfaces).

Ce modèle sera utilisé pour les applications univariées, en particulier pour l'estimation de la régularité des courbes (*cf.* Chapitre 2), ainsi que celle des fonctions moyenne et covariance (*cf.* Chapitre 3). Cependant, nous nous interesserons surtout à l'aspect multivarié des données, exprimé avec les signaux mesurés pour chaque observation $(x, y, v_x, v_y, a_x, a_y)$. Dans ce cas un modèle multivarié sera préférable.

Pour formaliser ce cadre multivarié, considérons $I_1, \ldots, I_P \subset \mathbb{R}$, P ensembles com-

pacts de $\mathbb{R}$, et notons $\boldsymbol{I} \coloneqq I_1 \times \cdots \times I_P$. Soit N fonctions $X^{(1)}, \ldots, X^{(n)}, \ldots, X^{(N)}$ un échantillon aléatoire d'un processus stochastique P-dimensionnel $X = (X_{\boldsymbol{t}} : \boldsymbol{t} \in \boldsymbol{I})$ ayant des trajectoires continues. Pour chaque $1 \leq n \leq N$ et un vecteur d'entier positif $\boldsymbol{M}_n = (M_{n,1}, \ldots, M_{n,P}) \in \mathbb{R}^P$, notons $T^{(n)}_{\boldsymbol{m}} = (T^{(n)}_{m_1}, \ldots, T^{(n)}_{m_P}), 1 \leq m_p \leq M_{n,p}, 1 \leq p \leq P$ les temps aléatoires d'observation pour la courbe $X^{(n)}$. Ceux-ci sont obtenus comme des réalisations indépendantes d'un vecteur aléatoire $\boldsymbol{T}$ prenant valeur dans $\boldsymbol{I}$. Les vecteurs $\boldsymbol{M}_1, \ldots, \boldsymbol{M}_N$ sont des réalisations indépendantes d'un vecteur d'entiers aléatoire $\boldsymbol{M}$ de moyenne $\boldsymbol{\mu}$ croissant avec N. Nous supposons que les réalisations de X, $\boldsymbol{M}$ et $\boldsymbol{T}$ sont mutuellement indépendants. Ainsi, pour chaque courbe $X^{(n)}$, nous observons les paires $(Y^{(n)}_{\boldsymbol{m}}, T^{(n)}_{\boldsymbol{m}}) \in \mathbb{R}^P \times \boldsymbol{I}$ tel que $Y^{(n)}_{\boldsymbol{m}}$ soit définit comme

$$Y^{(n)}_{\boldsymbol{m}} = X^{(n)}(T^{(n)}_{\boldsymbol{m}}) + \varepsilon^{(n)}_{\boldsymbol{m}}, \tag{2}$$

où les $\varepsilon^{(n)}_{\boldsymbol{m}}$ sont des réalisations indépendantes d'un vecteur aléatoire $\boldsymbol{\varepsilon}$. À moins que ce soit précisé autrement, $\boldsymbol{\varepsilon}$ est supposé être un vecteur gaussien centré et de matrice de variance-covariance $\sigma^2 \mathbf{1}_P$. De même que pour le cas univarié, nous pouvons étendre ce modèle aux processus multidimensionnels. en considérant les ensembles I_p comme des ensembles compacts de $\mathbb{R}^{d_p}, d_p \in \mathbb{N}^\star$ et en faisant les modifications nécessaires pour $\boldsymbol{T}$ et $\boldsymbol{M}$. Nous faisons une différence entre les termes "multidimensionnel" et "multivarié". Le premier fait référence à la dimension intrinsèque d'un unique processus stochastique (courbe, image, ...) définie par d, alors que le deuxième exprime le fait que l'on observe un vecteur de processus de dimension P. Ainsi, dans le cadre le plus général, nous pouvons considérer des réalisations d'un processus qui génére à la fois des courbes et des images. Ce modèle sera, en particulier, utlisé dans le Chapitre 4.

Suivant le contexte, le processus stochastique X pourra référer à un processus univarié ou multivarié et les observations au modèle (1) ou (2). L'échantillon des N réalisations de X est composé de deux sous-populations: un jeu d'entrainement (*learning set*) composé de N_0 courbes et un jeu de test (*online set*) de taille N_1. Ainsi, $1 \leq N_0, N_1 \leq N$ et $N_0 + N_1 = N$. Notons $X^{(1)}, \ldots, X^{(N_0)}$ les courbes venant du jeu d'entrainement et $X^{[1]} = X^{(N_0+1)}, \ldots, X^{[N_1]} = X^{(N)}$ celle venant du jeu de test. Le jeu d'entrainement sera utilisé pour estimer la régularité des courbes (Chapitre 2) et apprendre une partition des réalisations de X (Chapitre 4). Le jeu de test servira à mesurer la qualité d'estimation de la régularité (Chapitre 2), ainsi que la capacité de généralisation de l'algorithme de groupement (Chapitre 4).

Contributions et organisation du manuscrit

Au regard du contexte industriel, le problème initial était de construire des scénarios de roulage representatifs des habitudes de conduite. Ceux-ci pourront ensuite être utilisé pour prouver la fiabilité du véhicule. Pour cela, nous nous sommes intéressés au problème de *clustering* des trajectoires de véhicules. Les données peuvent provenir de différentes sources, comme les capteurs embarqués sur la voiture, les caméras fixes sur autoroutes ou bien des vidéos de drones, *etc.* L'approche suivie est donc de considérer les informations disponibles comme étant des réalisations de processus stochastiques. Ainsi, nous commençons par présenter, dans le Chapitre 1, la littérature sur différents aspects propres à l'analyse de données fonctionnelles. En particulier, nous nous concentrons sur la transformation de données échantillonnées en données fonctionnelles; ainsi que sur l'analyse en composantes principales fonctionnelles que ce soit le cas univarié (modèle (1)) ou bien multivarié (modèle (2)). Nous nous intéressons aussi à quelques méthodes de groupement spécifique à ce genre de données, et notamment celles du type *model-based*.

La majorité des méthodes standards concernant les données fonctionnelles sont basées sur l'hypothèse que les courbes (que l'on appelle aussi trajectoires ou signaux) sont observées de façon continue et sans erreur. Utilisant ces courbes, complétement observées et sans bruit, nous pouvons facilement construire des estimateurs des fonctions moyenne et covariance du processus stochastique qui a généré les données. Cependant, en général, les données réelles ne sont jamais ni continûment, ni exactement observées. Pour cette raison, une étape cruciale sera de reconstuire les trajectoires à partir de mesures bruitées ayant des points d'observations discrets et éventuellement aléaoires. Pour cette tâche, nous proposons un point de vue original qui fournit la ligne directrice de nos contributions methodologiques statistiques : l'utilisation de la régularité locale du processus générant les courbes.

Utilisant le grand nombre de trajectoires, ainsi que leur variabilité intrinsèque, nous proposons un estimateur simple de la régularité locale d'un processus stochastique dans le Chapitre 2. Les trajectoires, supposées indépendantes, peuvent être mesurées avec erreurs et échantillonnées aléatoirement. Des bornes non-asympotiques de la concentration de l'estimateur sont calculées. Munis de l'estimation de la régularité locale, nous construisons un estimateur, quasiment optimal, des courbes utilisant les polynômes locaux à partir d'un nouveau, possiblement très grand, échantillon de trajectoires bruitées. Des bornes uniformes du risque ponctuel non-asymptotique sont calculées sur le nouvel en-

semble de courbes. Notre estimateur montre de bonnes performances sur des simulations. L'application sur les données Next Generation Simulation (NGSIM) illustre l'efficacité de cette approche.

L'approche par estimation de la régularité locale du processus nous permet donc de proposer des estimateurs simples et non-paramétriques des fonctions moyenne et covariance de données fonctionnelles dans le Chapitre 3. Les trajectoires aléatoires ne sont pas nécessairement différentiable et leur régularité n'est pas connue. De plus, elles sont mesurées avec une erreur aléatoire pouvant être hétéroscédastique. Enfin, les points d'échantillonnages des courbes peuvent aussi être aléatoires (*independent design*) ou fixes (*common design*). Premièrement, nous proposons un deuxième estimateur simple de la régularité locale utilisant les caractéristiques *replication and regularization* propres aux données fonctionnelles. Ensuite, nous utilisons l'approche "*smoothing first, then estimation*" pour l'estimation des fonctions moyenne et covariance. Les nouveaux estimateurs non-paramétriques ont des vitesses de convergence quasiment optimales au sens minimax. Les résultats théoriques couvrent une large partie des situations pratiques, sont facilement calculables et actualisables, et ont de bonnes performances lors de simulations. Quelques exemples sur des jeux de données réels illustrent la qualité de cette nouvelle approche.

La troisième contribution méthodologique est liée aux algorithmes de *clustering*. Plus précisement, nous proposons un algorithme de groupement *model-based* pour une classe générale de données fonctionnelles pour laquelle les composantes peuvent être des courbes ou des images dans le Chapitre 4. De même que pour les chapitres précédents, les réalisations alétoires de données fonctionnelles peuvent être mesurées avec erreurs à des instants discrets et eventuellement aléatoires dans leur domain de définition. L'idée de l'algorithme de partitionnement est construire un ensemble d'arbres binaires par découpage récursif des observations. Le nombre de groupes est déterminé de façon *data-driven* et n'a donc pas besoin d'être spécifié par l'utilisateur. Ce nouvel algorithme fournit des résultats facilement interprétables, ainsi que de rapides prédictions pour de nouvelles observations. Les résultats sur des données simulées montrent de bonnes performances dans plusieurs cas complexes. La méthodologie est appliquée pour l'analyse de trajectoires de véhicules sur un rond-point en Allemagne. De cette analyse, nous pouvons en extraire une base de scénarios de roulage représentatifs des utilisateurs de ce rond-point.

Comme contribution pratique, nous proposons le package **Python**, `FDApy`, comme outil implémentant les méthodes liées aux données fonctionnelles que nous avons développé. La Chapitre 5 décrit les différentes fonctionnalités du package. Ainsi, ce package fournit

différents modules pour l'analyse de données fonctionnelles, incluant des classes pour des données de dimension différente ainsi que pour des données fonctionnelles irrégulièrement échantillonées. De plus, nous fournissons une boîte d'outils de simulation. Celle-ci peut être utilisée pour simuler différent groupes de données fonctionnelles. Certaines méthodologies pour analyser ce type de données sont implémentées, comme des méthodes de réduction de dimension (Multivariate Functional Principal Component Analysis (MFPCA)) et de *clustering* (`fCUBT`). De nouvelles méthodes peuvent être facilement ajoutées. Le package est disponible publiquement sur le *Python Package Index*, ainsi que sur Github.

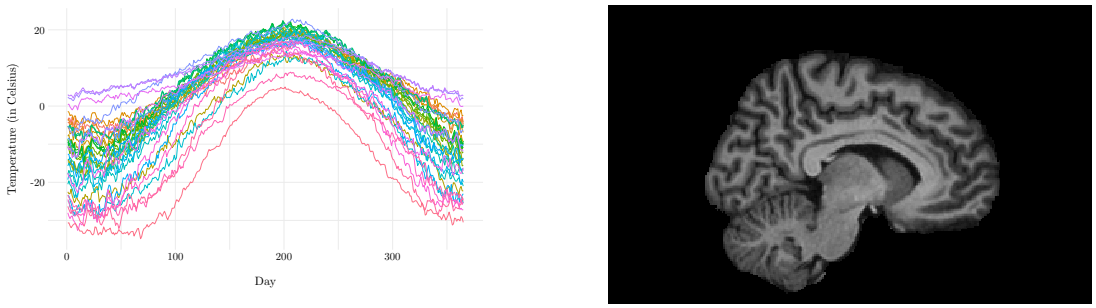
Chapter 1

Concepts of Functional Data and Clustering

Abstract: *The approach taken is to consider the available information as functional dataset of a general type. Therefore, we start our manuscript by reviewing several aspects from functional data analysis. In particular, there is a focus on the conversion from discrete data to functional data and on functional principal component analysis on both univariate and multivariate functional data. We also review some clustering methods specific to functional data, and especially the model-based ones.*

Contents

1.1 Functional Data Analysis 16
1.1.1 Univariate functional data 17
1.1.2 Multivariate functional data 17
1.2 From discrete data to functions 18
1.2.1 Basis expansion 19
1.2.2 Nonparametric estimation 21
1.3 Functional Principal Component Analysis 24
1.3.1 Computation of the eigenelements 25
1.4 Clustering 26
1.4.1 Model-based clustering 28
Appendix 30
1.A The EM algorithm 30



(a) Weather data (b) fMRI data

Figure 1.1: Examples of functional data: (a) average daily temperature for each day of the year for 35 canadian station ($N = 35$) and (b) sagittal slice of one brain from an fMRI image ($N = 1$).

1.1 Functional Data Analysis

Functional Data Analysis has been introduced by Ramsay [109] and deals with discrete observations of continuous multidimensional functions. This section briefly presents univariate and multivariate functional data as well as major statistical tools for them used along the thesis. An overall picture of this domain is accessible in Ramsay and Silverman [108], Ferraty and Vieu [46], Horvàth and Kokoszka [73], Hsing and Eubank [74] and Wang et al. [138].

Functional data may be found in many research and applied areas. These domains include, but are not limited to, meteorology (such as weather data), medicine (such as fMRI data), biology (such as growth curves) and, of course, automotive industry (such as sensors data). The Figure 1.1 shows two examples of such functional data. The Figure 1.1a represents the daily temperature for each day of the year averaged over 1960 to 1994 for 35 different canadian station[1] [110, 108]. These curves are used to quantify differences in temperature evolution within Canadian regions. This can be seen as univariate functional data with $N = 35$ observations. The Figure 1.1b concerns the study of fMRI images. We show $N = 1$ observation that represents a sagittal slice of one brain [101]. These data may be used to analyze human activity from a brain perspective (see [34] for example). The fMRI images may be express as multivariate functional data.

1. https://bit.ly/temperature_data

1.1.1 Univariate functional data

As previously said, Functional Data Analysis (FDA) refers to the analysis of data generated from an underlying continuous process. Thus, rather having a finite-dimensional vector as object to analyze, we would like to find insights from a set of functions which is infinite-dimensional. Let I be a compact interval of $\mathbb{R}^d$ with $d \in \mathbb{N}$, we observe N independant realizations $X^{(1)}, \dots, X^{(n)}, \dots, X^{(N)}$ of an underlying stochastic process $X = (X_t : t \in I)$. It is important to note that by definition $X^{(n)}, 1 \leq n \leq N$ is a continuous function on I.

It is usually assume that realizations of the process X belong to the space of square integrable functions over I, denoted by $\mathcal{L}^2(I)$, which is a Hilbert space (see *e.g.* Young [147, section 3.1]). The associated inner product in $\mathcal{L}^2(\mathcal{T})$ is defined by

$$\langle f, g \rangle = \int_I f(t)g(t)dt, \quad f, g \in \mathcal{L}^2(I).$$

From that, the norm follows as $\|f\|^2 = \langle f, f \rangle$, $f \in \mathcal{L}^2(I)$. Authors commonly assume smooth realizations of the process X. We will return on this assumption in Section 1.2.

The nice properties of the Hilbert space $\mathcal{L}^2(I)$ permit to extend most of the statistical tools used in multivariate analysis to the functional analysis. Thus, multivariate summary statistics such as mean and covariance can be directly extended to the functional case as pointwise mean and covariance functions. Let $\mu : I \mapsto \mathbb{R}$ denote the mean function of the process X and $C : I \times I \mapsto \mathbb{R}$ the covariance function, that is

$$\mu(t) \coloneqq \mathbb{E}(X(t)), \quad C(s,t) \coloneqq \mathbb{E}(\{X(s) - \mathbb{E}(X(s))\}\{X(t) - \mathbb{E}(X(t))\}), \quad s, t \in I.$$

The extension to the functional case of the Principal Component Analysis (PCA), named functional Principal Component Analysis (fPCA), is developped in the section 1.3. The infinite dimensionality of the data presents new challenges for these methods. In fact, most of them also returns infinite-dimensional results. One way to deal with it is to expand the data into a finite basis of functions, which is discussed in section 1.2.1.

1.1.2 Multivariate functional data

Regarding the application, we are indeed interested by multivariate functional data. So, we extend the concept of functional data to P-dimensional process. Let I_p, $1 \leq p \leq P$ be a compact interval of $\mathbb{R}^{d_p}$ with $d_p \in \mathbb{N}^\star$ and define $\boldsymbol{I} \coloneqq I_1 \times \cdots \times I_P$. We observe N

independant realizations $X^{(1)}, \ldots, X^{(n)}, \ldots, X^{(N)}$ of an underlying P-dimensional process $X = (X_1, \ldots, X_P)^\top$. The realizations of each coordinate $X_p : I_p \mapsto \mathbb{R}$ are assumed to belong to $\mathcal{L}^2(I_p)$.

Define $\mathcal{H} \coloneqq \mathcal{L}^2(I_1) \times \cdots \times \mathcal{L}^2(I_P)$. Ramsay and Silverman [108] propose to define an inner product on the space of P-dimensional functions as

$$\langle\langle f, g \rangle\rangle = \sum_{p=1}^{P} \langle f_p, g_p \rangle, \quad f, g \in \mathcal{H}.$$

From this definition, the induced norm is $\|\!|f|\!\|^2 = \sum_{p=1}^{P} \|f\|^2$, $f \in \mathcal{H}$. $\mathcal{H}$ is a Hilbert space with respect to the inner product $\langle\langle \cdot, \cdot \rangle\rangle$ (see Happ and Greven [64] for a proof). Let $\mu : \boldsymbol{I} \mapsto \mathbb{R}$ denote the mean function of the process X and $C : \boldsymbol{I} \times \boldsymbol{I} \mapsto \mathbb{R}$ the covariance function, that is

$$\mu(\boldsymbol{t}) = \mathbb{E}(X(\boldsymbol{t})), \quad C(\boldsymbol{s}, \boldsymbol{t}) = \mathbb{E}\left(\{X(\boldsymbol{s}) - \mu(\boldsymbol{s})\}\{X(\boldsymbol{t}) - \mu(\boldsymbol{t})\}^\top\right), \quad \boldsymbol{s}, \boldsymbol{t} \in \boldsymbol{I}.$$

This another kind of dimensionality, defined by the multivariate process, add another challenges for the analysis of this data. A large part of the state-of-art around multivariate functional data refers to processes that are defined on the same unidimensional domain, which is some compact subset of $\mathbb{R}$ (*e.g.* Di et al. [41], Dai and Genton [35], Schmutz et al. [120], Carroll et al. [23], Zhang et al. [150]). Only a very few of them consider processes defined on different domains and not necesserily unidimensional (*e.g.* Happ [65], Wong et al. [142]).

1.2 From discrete data to functions

From a practical point of view, due to finite resolution, data are recorded only on a finite grid of points. In particular, the information in the Controller Area Network (CAN) bus is transmitted as, at least, 20 Hz, which corresponds to 20 measures every second. However, for example, in the context of vehicle trajectory, the position or velocity of a vehicle does exist at every location, so it is quite natural to view it as functions defined over a continuous set. Moreover, these recording points can be different from one realization to another. The first step when applying functional data concepts is then to go from this discrete set of measurements to functional data. In addition, data may be contamined

with, not necessarily homoscedatic, measurement errors. For example, from the vehicle point of view, the detection of objects that are far away from our vehicle will likely lead to higher error measurements than for closer objects.

The Chapter 2 is dedicated to the smoothing of functional data. We consider to be in the univariate case and assume $X^{(1)}, \ldots, X^{(N)}$ to be an independant sample of a random process X defined over I. For each $1 \leq n \leq N$, and given a positive integer M_n, let $T_m^{(n)}$, $1 \leq m \leq M_n$, be the random observation times for the curve $X^{(n)}$. For each $1 \leq n \leq N$, the observations consist of the pairs $(Y_m^{(n)}, T_m^{(n)}) \in \mathbb{R} \times I$ defined in (1).

Up to now, smoothing methods for functional data usually involve a decomposition of each of the realizations of the process into a common basis of function, such as Fourier or splines bases. However, one may use kernel estimators on individual curve independently and estimate the bandwidth by cross-validation. But, none of these ideas capitalizes on the large number of curves to estimate the bandwidth and thus perform kernel regression on each of the curve using a bandwidth learned using all the observations. Chapter 2 introduces a new estimator for the bandwidth used in kernel regression methods which take into account the number of curves. Properties of this estimator have been investigated.

The smoothing of the data has mainly two purposes. On the one hand, in the case of data recorded at different sampled times, the smoothing is used to resample all the curves on a common grid. On the other hand, sensor measurements may have noise and thus, the smoothing will remove this noise. In the general case, data are recorded at different sampling points and with measurement errors.

1.2.1 Basis expansion

Let $\Phi = \{\phi_j(\cdot) : j \in \mathbb{N}\}$ be an infinite basis of $\mathcal{L}^2(I)$. The elements of Φ are linearly independent. And, every element of $\mathcal{L}^2(I)$ can be written as a linear combination of the elements of Φ. In particular, a realization, $X^{(n)}$, of the stochastic process X expands into

$$X^{(n)}(\cdot) = \sum_{j \geq 1} c_{j,n} \phi_j(\cdot),$$

where $\{c_{j,n}\}_{j \geq 1}$ is an infinite set of coefficients. Moreover, elements of Φ may be orthonormal or not. The underlying idea of basis expansion is the approximation of a realization $X^{(n)}$ of the process X by its projection on the span of a finite basis of functions

$\Phi_J = \{\phi_j : 1 \leq j \leq J\}$, $J \in \mathbb{N}$, which is a finite subset of Φ,

$$X^{(n)}(\cdot) \approx \sum_{j=1}^{J} c_{j,n}\phi_j(\cdot), \tag{1.1}$$

where $\{c_{j,n}\}_{1\leq j\leq J}$ is a subset of $\{c_{j,n}\}_{j\geq 1}$. Here, the term *basis*, to characterize the set Φ_J, is not exactly correct because the finite number of functions within Φ_J cannot span the entirety of the infinite space $\mathcal{L}^2(I)$. However, to simplify the terminology, we will continue to refer to Φ_J as a basis. Returning to the model (1), in this basis, we observe:

$$Y_m^{(n)} = \sum_{j=1}^{J} c_{j,n}\phi_j(T_m^{(n)}) + \varepsilon_m^{(n)}, \quad 1 \leq m \leq M_n. \tag{1.2}$$

The equation (1.2) is a standard linear model in the coefficient $c_{j,n}$ which can be solved by a least-squares approach:

$$\widehat{c}_n = \arg\min_{c_n} \left(Y^{(n)} - \Phi_n c_n\right)^\top \left(Y^{(n)} - \Phi_n c_n\right) = \left(\Phi_n^\top \Phi_n\right)^{-1} \Phi_n^\top Y^{(n)},$$

where $Y^{(n)} = (Y_1^{(n)}, \ldots, Y_{M_n}^{(n)})^\top$, $\Phi_n \in \mathbb{R}^{M_n \times J}$ with entries $\phi_j(T_m^{(n)})$ and $c_n = (c_{1,n}, \ldots, c_{J,n})^\top$. This expression exists and is unique if and only the number of elements in the basis Φ_J is smaller than the number of observation points M_n. Thus, an estimation of $X^{(n)}(\cdot)$ is given by

$$\widehat{X}^{(n)}(\cdot) = \sum_{j=1}^{J} \widehat{c}_{j,n}\phi_j(\cdot).$$

Herein, all the observations $X^{(n)}$ can be summarized by a J-dimensional vector. And, thus, all the classical statistical methods for multivariate data can be applied on the vector $\widehat{c}_n$ and returning to the original space using the equation (1.1). There are numerous methods using this approach in different context, *e.g* for classification (see *e.g.* [108, 138]) or clustering (see Section 1.4). The accuracy of results will depends on the goodness of the basis representation which is influenced by the type of the basis (Fourier, B-Splines, *etc.*) along the number of the functions in the basis. In the following, Fourier and B-Splines basis are presented as common ones used in the functional data context. Other basis can be found in the literature. We may cite polynomial basis or wavelets as example. One possibility is to build data-based basis derived using principal components, named fPCA presented in the Section 1.3.

Fourier basis

In the case of periodic observations, a particularly suitable basis is the Fourier basis. The basis functions ϕ_j's take the following form:

$$\phi_0(t) = 1, \quad \phi_{2j-1}(t) = \sin(j\omega t) \quad \text{and} \quad \phi_{2j}(t) = \cos(j\omega t), \quad t \in I,\ j = 1, 2, \ldots$$

The constant ω defines the period of oscillation of ϕ_{2j-1} and ϕ_{2j}. It determines the period and the length of the interval $|I| = 2\pi/\omega$. These functions have excellent computational properties; in particular, the derivatives are easy to compute but retain complexity. This basis is very popular in signal analysis.

Splines

When the curves exhibit no particular shape, B-Splines basis functions is very popular in the functional data context. Splines are polynomial segments joined end-to-end constrained to be smooth at the join. The joining points are called *knots*. Constraints are easily implemented by placing more knots to strong curvatures of the curves. As the splines are defined by polynomials, the derivatives are fast to compute. To compute higher derivative degrees, one just needs higher polynomial degrees. A popular choice is to use 4-order polynomials such that the second derivatives are continuous.

1.2.2 Nonparametric estimation

One might not want to specify a basis expansion to the data and build the underlying function in a data-driven fashion. Thus, no particular form (polynomial or periodic for example) are assumed for the functions in this case. This approach has a major advantage compare to basis expansion: one is not limited to a particular set of functions, and so, it is possible to fit a wider set of functions. However, here, we are not limited to the estimation of a small set of parameters but we have to estimate a complete function which is more computationally demanding.

Returning to the model defined in (1), the objective is to estimate the function $X^{(n)}(\cdot)$ using the available sample points. One of the most popular nonparametric smoother is the local polynomial estimator [45] and particularly, the local constant smoother, the Nadaraya-Watson estimator [100, 139]. This type of estimator crucially depends on a tuning parameter, the so-called *bandwidth*, denoted by h.

Nadaraya-Watson estimator

Let $t_0 \in I$ be the evaluation point at which we want to estimate $X^{(n)}$. The Nadaraya-Watson estimate is a weighted average of $\left(Y_1^{(n)}, \ldots, Y_{M_n}^{(n)}\right)$. The weights are defined as

$$w_m^{(n)}(t_0) \coloneqq \frac{K_h(T_m^{(n)} - t_0)}{\sum_{m=1}^{M_n} K_h(T_m^{(n)} - t_0)}, \quad 1 \leq m \leq M_n,$$

where $K_h(\cdot) = h^{-1}K(\cdot/h)$ is a positive kernel function. Thus, an estimation of $X^{(n)}(\cdot)$ at t_0 is given by

$$\widehat{X}^{(n)}(t_0) = \sum_{m=1}^{M_n} w_m^{(n)}(t_0) Y_m^{(n)}.$$

Local polynomial estimator

Let $\mathbf{d} \geq 0$ be an integer and $t_0 \in I$ be the evaluation points for the estimation of $X^{(n)}$. For any $u \in \mathbb{R}$, we consider the vector $U(u) = (1, u, \ldots, u^{\mathbf{d}}/\mathbf{d}!)$ and note $U_h(\cdot) = U(\cdot/h)$. Let $K : \mathbb{R} \to \mathbb{R}$ be a positive kernel and define $K_h(\cdot) = h^{-1}K(\cdot/h)$. Moreover, we define:

$$\vartheta_{M_n,h} \coloneqq \underset{\vartheta \in \mathbb{R}^{\mathbf{d}+1}}{\arg\min} \sum_{m=1}^{M_n} \left\{ Y_m^{(n)} - \vartheta^\top U_h\left(T_m^{(n)} - t_0\right) \right\}^2 K_h\left(T_m^{(n)} - t_0\right),$$

where h is the bandwidth. The vector $\vartheta_{M_n,h}$ satisfies the normal equations $A\vartheta_{M_n,h} = a$ with

$$A = A_{M_n,h} = \frac{1}{M_n} \sum_{m=1}^{M_n} U_h\left(T_m^{(n)} - t_0\right) U_h^\top\left(T_m^{(n)} - t_0\right) K_h\left(T_m^{(n)} - t_0\right), \tag{1.3}$$

$$a = a_{M_n,h} = \frac{1}{M_n} \sum_{m=1}^{M_n} Y_m^{(n)} U_h\left(T_m^{(n)} - t_0\right) K_h\left(T_m^{(n)} - t_0\right). \tag{1.4}$$

Let λ be the smallest eigenvalue of the matrix A and remark that, whenever $\lambda > 0$, we have $\vartheta_{M_n,h} = A^{-1}a$. With at hand an estimation of the bandwidth $\widehat{h}$, the local polynomial estimator of $\widehat{X}^{(n)}(t_0)$ of order $\mathbf{d}$ is given by:

$$\widehat{X}^{(n)}(t_0) = U^\top(0)\widehat{\vartheta}, \quad \text{where} \quad \widehat{\vartheta} = \vartheta_{M_n,\widehat{h}}.$$

Bandwidth selection

The quality of the nonparametric estimation of $X^{(n)}$ crucially depends on the bandwidth, which have to be chosen. Let $\widehat{X}_h^{(n)}$ be the estimator of $X^{(n)}$ using the bandwidth h. Before considering the estimation of the bandwidth, we need an error criterion for the estimator $\widehat{X}_h^{(n)}$. One commonly used criterion is the *mean integrated squared error*:

$$\mathrm{MISE}\left[\widehat{X}_h^{(n)}(\cdot)\right] := \int \mathbb{E}\left\{X^{(n)}(t) - \widehat{X}_h^{(n)}(t)\right\}^2 f(t)dt,$$

where f is the marginal probability density function of the observation points. We are interested by the bandwidth that minimizes the MISE,

$$\widehat{h}_{\mathrm{MISE}} := \arg\min_{h>0} \mathrm{MISE}\left[\widehat{X}_h^{(n)}(\cdot)\right].$$

See [118, 135, 55] for some estimators of $\widehat{h}_{\mathrm{MISE}}$.

Moreover, the Cross-Validation method is a general approach commonly used to set *hyperparameters* of statistical models. Hyperparameters are parameters that should be fixed prior the training of the model. Hastie et al. [68, Chapter 7.10] describe the general procedure for the cross-validation. In our case, the *leave-one-out cross validation* method will consists as the following. Let $\mathcal{H}$ be a finite set of bandwidths. An estimation of h by cross-validation is given by

$$\widehat{h}_{CV} := \arg\min_{h\in\mathcal{H}} \frac{1}{M_n}\sum_{m=1}^{M_n}\left(Y_m^{(n)} - \widehat{X}_{h,-m}^{(n)}(T_m^{(n)})\right)^2,$$

where $\widehat{X}_{h,-m}^{(n)}(T_m^{(n)})$ is an estimation of $X^{(n)}(T_m^{(n)})$ without the pair $\left(Y_m^{(n)}, T_m^{(n)}\right)$ using the Nadaraya-Watson or local polynomial estimator with bandwidth h.

The previous methods were designed for the case where one observes only one curve. Thus one has to apply them for each curve separately, which could require large amounts of resources. In Chapter 2, we develop an estimation procedure for the bandwidth based on the regularity of the trajectories of the process that takes strength from the information contained in the whole set of its realizations. As the purpose is the simultaneous denoising of a set of N curves, we will consider the following pointwise risk:

$$\mathcal{R}\left(\widehat{X}_h; t_0\right) := \mathbb{E}\left[\max_{1\le n\le N}\left|X_h^{(n)}(t_0) - \widehat{X}_h^{(n)}(t_0)\right|^2\right], \quad t_0 \in I.$$

1.3 Functional Principal Component Analysis

FPCA is a direct generalization of the PCA, used in multivariate statistical analysis, to continuous data. The motivations of fPCA are multiple. On one hand, this method aims to describe the structure of their covariance by extracting the most important modes of variation. On the other hand, this modes of variation are used to defined a basis of functions that approximate the curves as closely as possible.

Starting from the univariate case, defined in Section 1.1.1. So, $\mathcal{H} = \mathcal{L}^2(I)$. Let $\Gamma : \mathcal{H} \mapsto \mathcal{H}$ denote the covariance operator of X, defined as the integral operator with kernel C. That is, for $f \in \mathcal{H}$ and $t \in I$, $\Gamma f(t)$ is given by

$$\Gamma f(t) := \langle C(\cdot, t), f(\cdot) \rangle, \quad t \in I.$$

By the results in Horvàth and Kokoszka [73, Chapter 2.3], the covariance operator Γ is a linear, self-adjoint and positive operator in $\mathcal{H}$. Moreover, Γ is a compact operator. Using the theory of Hilbert-Schmidt operators, *e.g.* [114, Chapter VI], there exists a complete orthonormal basis $\{\varphi_j\}_{j\geq 1}$ and a sequence of real numbers $\lambda_1 \geq \lambda_2 \geq \ldots \geq 0$ such that

$$\Gamma\varphi_j = \lambda_j\varphi_j \quad \text{and} \quad \lambda_j \to 0 \text{ as } j \to \infty,$$

where $\{\lambda_j\}_{j\geq 1}$ is the set of eigenvalues of the covariance operator Γ associated to $\{\varphi_j\}_{j\geq 1}$ the set of its eigenfunctions. Then, using the Mercer's theorem [96], the covariance function has the decomposition:

$$C(s,t) = \sum_{j\geq 1} \lambda_j\varphi_j(s)\varphi_j(t), \quad s,t \in I.$$

Then, a process X admits the Karhunen-Loève representation [86, 92]

$$X(t) = \mu(t) + \sum_{j\geq 1} \mathfrak{c}_j\varphi_j(t), \quad t \in I, \quad \text{with} \quad \mathfrak{c}_j = \langle X - \mu, \varphi_j \rangle, \tag{1.5}$$

and $\mathbb{E}(\mathfrak{c}_j) = 0$ and $\text{Cov}(\mathfrak{c}_j, \mathfrak{c}_l) = \lambda_j \mathbf{1}_{j=l}$. We call $\{\varphi_j\}_{j\geq 1}$ the fPCA basis.

Let $J \geq 1$ and assume that $\lambda_1 > \lambda_2 > \ldots > \lambda_J > \lambda_{J+1}$, which in particular implies that the first J eigenvalues are nonzero. Using [73, Theorem 3.2], we decuce that, up to

a sign, the elements of the fPCA basis are characterized by the following property:

$$\begin{aligned}\varphi_1 &= \arg\max_{\varphi} \langle \Gamma\varphi, \varphi\rangle \quad \text{such that} \quad \|\varphi\| = 1,\\ \varphi_j &= \arg\max_{\varphi} \langle \Gamma\varphi, \varphi\rangle \quad \text{such that} \quad \|\varphi\| = 1 \text{ and } \langle \varphi, \varphi_l\rangle = 0, \quad \forall l < j \leq J. \end{aligned} \tag{1.6}$$

From a practical point of view, the infinite number of eigenfuntions is impossible to achieve. Moreover, the decreasing of the eigenvalues λ_j is commonly fast and so, a few number of eigenvalues are enough to explain a large proportion of the variability in the data. So, a realization, $X^{(n)}$, of the process X can be approximated by truncating the infinite sum at the first J terms,

$$X^{(n)}(t) \approx \mu(t) + \sum_{j=1}^{J} \mathfrak{c}_{j,n}\varphi_j(t), \quad t \in I, \quad \text{with} \quad \mathfrak{c}_{j,n} = \langle X^{(n)} - \mu, \varphi_j\rangle. \tag{1.7}$$

The approximation (1.7) is a particular basis representation of $X^{(n)} - \mu$ as the one in (1.1) with principal components. The fPCA basis is the one which will induce the most accurate truncation for a given J (see *e.g.* [73, Section 3.2]). Therefore, it should be a privileged one when it comes to choose a basis.

Concerning the multivariate case, defined in 1.1.2, one approach was to stack all the components of the process X into one and perform an univariate fPCA on the resulting observations. Then, the principal components are splitted in multiple parts, each of them corresponding to a component. For example, Ramsay and Silverman [108] use this methodology on gait data. Berrendero et al. [4] propose another method for MFPCA. Thus, they perform a standard PCA on each sampling points. The principal components are rebuilt by interpolating the results of the PCA. Both of theses methods can only be performed if the observations are sampled on a common one-dimensional grid. Recently, Happ and Greven [64] developed a theory for MFPCA, that allows for different domains definition and can be written in a similar way than before by setting $\mathcal{H} = \mathcal{L}^2(I_1) \times \cdots \times \mathcal{L}^2(I_P)$ and replacing $\langle \cdot, \cdot \rangle$ by $\langle\langle \cdot, \cdot \rangle\rangle$. See Chapter 4 for details.

1.3.1 Computation of the eigenelements

The different elements are, in general, not known and have to be estimated from the observations that are possibly observed on different sparse grid points. These quantities are the mean function $\mu(\cdot)$, the covariance function $C(\cdot, \cdot)$, the eigenvalues $\{\lambda_j\}_{j\geq 1}$, the

eigenfunctions $\{\varphi_j\}_{j\geq 1}$ and the scores $\{\mathfrak{c}_j\}_{j\geq 1}$.

In the case of dense functional data that are observed on the same grid, Ramsay and Silverman [108, Chapter 8] expand the data into a common fixed basis, such as B-splines or Fourier, and reduced the estimation of the principal components to a matrix eigendecomposition with eventually weighted entries to consider non orthonormal basis. The number of principal components that can be computed this way is equal to the number of functions in the basis. It results to estimated eigenvalues $\widehat{\lambda}_j$ associated to eigenfunctions $\widehat{\varphi}_j$. Then, the scores can be approximated by numerical integration:

$$\widehat{\mathfrak{c}}_{j,n} = \langle \widehat{X}^{(n)} - \widehat{\mu}, \widehat{\varphi}_j \rangle = \int_I (\widehat{X}^{(n)}(t) - \widehat{\mu}(t))\widehat{\varphi}_j(t)dt, \quad 1 \leq j \leq J, \quad n = 1, \ldots, N.$$

The quality of the estimation clearly depends on the capacity of the basis to fit the data. Moreover, numerical integration relies on the number and location of the sampling points. In the case of data with measurements error, as in model (1), some preprocessing steps will be needed.

Yao et al. [146] propose the alternative Principal component Analysis through Conditional Expectation (PACE) method to obtain the scores for sparse and irregularly sampled functional data with measurement errors. Considering data from the model (1) with Gaussian errors with common variance σ^2, expanding in (1.5), we note $\widetilde{Y}^{(n)} = (Y_1^{(n)}, \ldots, Y_{M_n}^{(n)})^\top$, $\mu^{(n)} = (\mu(T_1^{(n)}), \ldots, \mu(T_{M_n}^{(n)}))^\top$ and $\varphi_{j,n} = (\varphi_j(T_1^{(n)}), \ldots, \varphi_j(T_{M_n}^{(n)}))^\top$. Then, the best prediction of the scores for the nth observation, given the observed data and the sampling points, under the Gaussian assumptions, is the conditional expectation:

$$\widetilde{\mathfrak{c}}_{j,n} = \mathbb{E}\left(\mathfrak{c}_{j,n} | \widetilde{Y}^{(n)}\right) = \lambda_j \varphi_{j,n}^\top \Sigma_n^{-1} \left(\widetilde{Y}^{(n)} - \mu^{(n)}\right),$$

where $\Sigma_n \in \mathbb{R}^{M_n \times M_n}$ with entries $\Sigma_{n,m,m'} = C(T_m^{(n)}, T_{m'}^{(n)}) + \sigma^2 \mathbf{1}_{\{m=m'\}}$. An estimation of the scores is computed by replacing λ_j, $\varphi_j^{(n)}$, Σ_n and $\mu^{(n)}$ by their empirical version. Note that, given the information of the nth observation, the Gaussian assumption does not have to hold for $\widetilde{\mathfrak{c}}_{j,n}$ to be the best linear prediction of $\mathfrak{c}_{j,n}$. See [146] for details.

1.4 Clustering

Let $\mathcal{S}$ be a sample of realizations of the process X, that could be univariate or multivariate. We consider the problem of learning a partition $\mathcal{U}$ from $\mathcal{S}$ such that every element U of $\mathcal{U}$ gathers similar elements of $\mathcal{S}$. The similarity criteria must be specified. Given two

elements of $\mathcal{S}$ and a similarity measure, the larger the measure between the elements is, the higher the similarity is. The clustering problem is therefore to have elements with a high intra-groups similarity and low inter-groups similarity.

Clustering procedures for functional data have been widely studied in the last two decades, see for instance, [38, 39, 25, 83] and references therein. See also Bouveyron et al. [13] for a recent textbook. In particular, for Gaussian processes, Tarpey and Kinateder [129] show that the cluster centers found with k-means are linear combinations of the eigenfunctions from the fPCA. A discriminative functional mixture model is developed in [11] for the analysis of a bike sharing system from cities around the world.

Algorithms built to handle multivariate functional data have gained much attention in the last few years. Some of these methods are based on k-means algorithm with a specific distance function adapted to multivariate functional data. See *e.g* [124, 133, 78, 149]. Let f, g be two elements of $\mathcal{H}$ (defined in Section 1.1.2). A suitable distance between f and g in the space $\mathcal{H}$ is defined as

$$d_r(f,g) := \left(\sum_{p=1}^{P} \int_{I_p} \left\{ \frac{d^r f_p(t_p)}{dt_p^r} - \frac{d^r g_p(t_p)}{dt_p^r} \right\}^2 dt_p \right)^{1/2}, \quad r \geq 0,$$

where $d^r f_p(t_p)/dt_p^r$ (or $d^r g_p(t_p)/dt_p^r$) is the rth derivative of $f_p(t_p)$ (or $g_p(t_p)$).

Some other methodologies are available. Kayano et al. [87] consider Self-Organizing Maps built on the coefficients of the curves into orthonormalized Gaussian basis expansion. The underlying model for these methods usually consider only amplitude variations. Unlike others, Park and Ahn [102] present a specific model for functional data to consider phase variations. Finally, Traore et al. [134] propose a mix between dimension reduction and nonparametric approaches by deriving the envelope and the spectrum from the curves and have applied it to nuclear safety experiments.

In the Section 1.4.1, we focus on the model-based methods for the clustering of functional data. Chapter 4 presents a new model-based clustering algorithm for multivariate functional data defined on different domain dimensions using unsupervised binary trees.

One of the lastest development on the clustering of functional data concerns the so-called co-clustering methodology. The co-clustering refers to algorithms that provides a clustering on the both the observations and features. These algorithms usually relies on an adaptation of the Latent Block Model (LBM) [60] to the functional context, named functional Latent Block Model (fLBM). Bouveyron et al. [12] successfully applied the fLBM to the electricity consumption curves, while it is used by Ben Slimen et al. [3] to

optimize the topology of 4G mobile networks.

The estimation of number of clusters in a dataset is a widely discussed topic in the clustering community. Multiple criteria, for multivariate data, have been introduced to tackle this problem. For example, one may cite stability criteria (see *e.g.* [137] for a review), the Calinski-Hrabasz Index [21] , the Davis-Bouldin Index [36] or the Gap statistic [132] to name a few. In general, the clustering procedure is run for different number of clusters and the best is retained according to the minimisation or maximisation of some criteria. As most of these criteria are distance-based, a direct generalisation to the functional case is to applied them with specific distance for functional data. See *e.g.* [83] for a review. Recently, Zambom et al. [148] proposed a new approach for this problem based on the aggregation of two test statistics, one to test the parallelism of the curves and one that test the difference in means. Due to the tree structure, the method proposed in Chapter 4 allows to estimate the number of clusters in the dataset.

1.4.1 Model-based clustering

Standard model-based clustering approaches consider that data is sampled from a mixture of probability densities on a finite dimensional space. However, this approximation is not directly applicable to functional data since the notion of probability density usually does not exist for functional random variables (see [37, 84]). As a consequence, model-based clustering approaches in the functional data context assume a mixture of parametric distributions on the coefficients of the representation of the realizations of the process X in some basis.

Let K be a positive integer, and let Z be a discrete random variable taking values in $\{1, \dots, K\}$ such that

$$\mathbb{P}(Z = k) = \pi_k \quad \text{with} \quad \pi_k > 0 \quad \text{and} \quad \sum_{k=1}^{K} \pi_k = 1.$$

The variable Z is a latent variable representing the cluster membership of the realizations of the process. Jacques and Preda [84] developed model-based clustering for multivariate functional data. In their model, they assume a cluster-specific Gaussian distribution for the principal component scores, *i.e.*

$$f_k(\mathfrak{c} \mid \Sigma_k) = \prod_{j=1}^{J_k} f_{\mathfrak{c}_j,k}(\mathfrak{c}_j \mid \lambda_{j,k}), \quad 1 \leq k \leq K,$$

where J_k is the number of principal components retained in the Karhunen-Loève approximation (1.7) for the cluster $Z = k$, $\mathfrak{c}_j$ is the jth principal components score defined in (1.5) and $f_{\mathfrak{c}_j,k}$ is the univariate Gaussian density with zero mean and variance $\lambda_{j,k}$ of $\mathfrak{c}_j$. The matrix Σ_k is the diagonal matrix of the principal components variances $\text{diag}(\lambda_{1,k}, \ldots, \lambda_{J_k,k})$. Thus, the density of the principal components $\mathfrak{c}$ is given by

$$f(\mathfrak{c} \mid \theta) = \sum_{k=1}^{K} \pi_k \prod_{j=1}^{J_k} f_{\mathfrak{c}_j,k}(\mathfrak{c}_j \mid \lambda_{j,k}),$$

where $\theta = \{(\pi_k, \lambda_{1,k}, \ldots, \lambda_{J_k,k})_{1 \leq k \leq K}\}$. The estimation of the parameters θ and $(J_1 \ldots, J_k)$ is based on a modification of the Expectation-Maximisation (EM) algorithm (see *e.g* [68, 13] and Appendix 1.A). Between the E-step and the M-step, they update the scores and re-estimate the group specific dimension $(J_1, \ldots, J_k)$. The updating of the scores is performed using a group specific MFPCA weighted by the conditional probability (1.8) computed in the E-step. For each of the group k, the dimension J_k is estimated using the Cattell scree-test [26] on the group specific MFPCA. It results to cluster specific eigenelements estimated using an approximation of the curves into a finite dimensional functional space. This model is also used in [29].

Schmutz et al. [120] have recently extended the previous model by modeling all principal components whose estimated variances are non-null. Considering that the P-dimensional process X is expanding as (1.7) using J_+ basis functions, the matrix Σ_k is assumed to have the following shape:

$$\Sigma_k = \begin{pmatrix} \Lambda_k & \mathbf{0} \\ \mathbf{0} & B_k \end{pmatrix} \in \mathbb{R}^{J_+ \times J_+}, \quad \text{where } \Lambda_k = \text{diag}(\lambda_{1,k}, \ldots, \lambda_{J_k,k}) \text{ and } B_k = \text{diag}(b_k, \ldots, b_k).$$

In fact, in the model from Jacques and Preda [84], the matrix B_k is null for all $k = 1, \ldots, K$. The estimation of the model parameters is performed using a similar modified EM algorithm than Jacques and Preda [84], with minor change to take into account the particular form of the variance matrices Σ_k.

Note that, in both models, the algorithm is launched with a pre-specified number of cluster K. Thus, in order to estimate the number of clusters in the dataset, different values for K have to be tested. Usually, the returned model is the one that maximizes a chosen criterion, such as Bayesian Information Criterion (BIC) [121], Integrated Completed Likelihood (ICL) [6], *etc.*

Appendix

1.A The EM algorithm

Let K be a positive integer, and let Z be a discrete random variable taking values in $\{1, \ldots, K\}$ such that

$$\mathbb{P}(Z = k) = \pi_k \quad \text{with} \quad \pi_k > 0 \quad \text{and} \quad \sum_{k=1}^{K} \pi_k = 1.$$

Let X be a multivariate random variable in $\mathbb{R}^J$. We assume that X has a multivariate Gaussian distribution with K-components:

$$f(\mathbf{x}) = \sum_{k=1}^{K} \pi_k f_k(\mathbf{x} \mid \mu_k, \Sigma_k), \quad \mathbf{x} \in \mathbb{R}^J,$$

where, for each $1 \leq k \leq K$, $f_k(\cdot \mid \mu_k, \Sigma_k)$ is the probability density function of a multivariate Gaussian distribution for the kth component with mean μ_k and variance Σ_k. Usually, the components parameters μ_k and Σ_k are unknown and have to be estimated. This can be done by maximum likelihood estimation. Let $\mathsf{X} = (x_1, \ldots, x_N)$ be a vector of N realizations of the random variable X and $\theta = (\pi_k, \mu_k, \Sigma_k)_{1 \leq k \leq K}$ is the vector of components. The complete log-likelihood of the model is

$$\mathcal{L}(\theta \mid \mathsf{X}) = \sum_{n=1}^{N} \sum_{k=1}^{K} z_{nk} \log \left\{ \pi_k (2\pi)^{-J/2} \det(\Sigma_k)^{-1/2} \exp\left(-\frac{1}{2} (x_n - \mu_k)^\top \Sigma_k^{-1} (x_n - \mu_k) \right) \right\},$$

where $z_{nk} = 1$ if x_n belongs to cluster k, and 0 otherwise. The z_{nk} are not observed. The EM algorithm [40] is an iterative procedure used to estimate the maximum of the log-likelihood function. This is a two-steps method. First, we estimate the conditional expectation of the complete log-likelihood given the observed data and the current estimation of the parameters, which the expectation step (**E-step**). In the case of Gaussian mixture model, the E-step consists in computing

$$\widehat{z}_{nk} = \mathbb{E}\left(z_{nk} \mid x_n, \theta\right) = \frac{\widehat{\pi}_k f_k(x_n \mid \widehat{\mu}_k, \widehat{\Sigma}_k)}{\sum_{k'=1}^{K} \widehat{\pi}_{k'} f_{k'}(x_n \mid \widehat{\mu}_{k'}, \widehat{\Sigma}_{k'})}, \quad 1 \leq k \leq K, \ 1 \leq n \leq N. \tag{1.8}$$

Let $n_k := \sum_{n=1}^{N} \widehat{z}_{nk}$. The second step consists in finding the parameters that maximize the expected log-likelihood found during the previous step, known as the maximisation step (**M-step**). For Gaussian mixture models, it exists a closed-form solution of the maximisation problem as:

$$\widehat{\mu}_k = \frac{1}{n_k} \sum_{n=1}^{N} \widehat{z}_{nk} x_n, \quad \widehat{\Sigma}_k = \frac{1}{n_k} \sum_{n=1}^{N} \widehat{z}_{nk} (x_n - \widehat{\mu}_k)(x_n - \widehat{\mu}_k)^\top \quad \text{and} \quad \widehat{\pi}_k = \frac{n_k}{N}. \tag{1.9}$$

The Algorithm 1 presents a possible implementation of the EM algorithm in the case of a Gaussian mixture model. It has been proved that the EM algorithm converges to a local maximum of the log-likelihood function [40]. To prevent local maximum, the algorithm can be run multiple times, and the one that lead to the best result (for some criteria) is returned. Multiple variants of the EM algorithm have been introduced. For example, the Generalized EM [40], which replace the maximisation of the log-likelihood by an "improvement", the Classification EM [27], build for clustering purpose, and the Stochastic EM [28], to prevent to find a local maximum of the log-likelihood, to name a few.

Algorithm 1: One run of the EM algorithm for Gaussian mixture estimation.

Input: A training sample $x_1, \ldots, x_N$, a threshold ϵ and a maximum number of iterations M.
Initialization: Randomly initialize the components parameters μ_k, Σ_k and π_k.
E-step: Estimate r_{nk} defined in (1.8).
M-step: Estimate the components defined in (1.9).
Recursion: Run the **E-step** and **M-step** until the difference between the log-likelihoods in the last two iterations is less than ϵ or M iterations have been run.
Output: An estimate of the components parameters $\widehat{\mu}_k, \widehat{\Sigma}_k$ and $\widehat{\pi}_k$.

Chapter 2

Learning the Smoothness of Noisy Curves with Application to Online Curve Denoising

Abstract: *Combining information both within and across trajectories, we propose a simple estimator for the local regularity of a stochastic process. Independent trajectories are measured with errors at randomly sampled time points. Non-asymptotic bounds for the concentration of the estimator are derived. Given the estimate of the local regularity, we build a nearly optimal local polynomial smoother from the curves from a new, possibly very large sample of noisy trajectories. We derive non-asymptotic pointwise risk bounds uniformly over the new set of curves. Our estimates perform well in simulations. Real data sets illustrate the effectiveness of the new approaches. Part of this study was presented within the conference* Journées de Statistique *of the French Statistical Society of 2020.*

Contents

2.1 Introduction 34
2.2 Local regularity estimation 37
2.2.1 The methodology 37
2.2.2 Concentration bounds for the local regularity estimator 40
2.2.3 The case of smooth trajectories 43
2.2.4 The case of conditionally heteroscedastic noise 44
2.3 Adaptive optimal smoothing 46
2.3.1 Local polynomial estimation 47
2.4 Empirical analysis 50
2.4.1 Simulation experiments 52

2.4.2 Real data analysis: the NGSIM Study 55
Appendices 63
2.A Proof of Theorem 1 63
2.B Proofs of Theorems 2 and 3 69
2.C Technical lemmas 76
2.D Moment bounds for spacings 85
2.D.1 The uniform case 86
2.D.2 The general case 87
2.D.3 Wendel's type inequalities for gamma function ratios 90
2.E Additional simulation results 92
2.E.1 The settings for X 92
2.E.2 On the computation time 93
2.E.3 On the estimation of the local regularity 93
2.E.4 On the pointwise risk 96
2.F Traffic flow: Montanino and Punzo methodology 97
2.G Complements on the real-data applications 100
2.G.1 Canadian weather 100
2.G.2 Household Active Power Consumption 102
2.G.3 PPG-Dalia 103

2.1 Introduction

More and more phenomena in modern society produce observation entities in the form of a sequence of measurements recorded intermittently at several discrete points in time. Very often the measurements are noisy and the observation points in time are neither regularly distributed nor the same across the entities. Functional Data Analysis considers such data as being values on the trajectories of a stochastic process, recorded with some error, at discrete random times. One of the main purposes of the FDA is to recover the trajectories, also called curves or functions, at any point in time. See, *e.g.*, [108, 73, 138, 152] for some recent references. Whatever the approach for recovering the curve is, in the existing

literature it is usually assumed that, for each curve, a certain number of derivatives exist. However, many applications, some of them presented in the following, indicate that assuming that the curves admit second, third,... order derivatives is not realistic. Assuming that the curves to be reconstructed are smoother than they really are could lead to missing important information carried by the data. In this contribution, we propose a definition of the local regularity of the curves which could be easily estimated from the data and used to estimate the curves.

To formalize the framework, let $I \subset \mathbb{R}$ be a compact interval of time. We consider N functions $X^{(1)}, \ldots, X^{(n)}, \ldots, X^{(N)}$ generated as a random sample of a stochastic process $X = (X_t : t \in I)$ with continuous trajectories. For each $1 \leq n \leq N$, and given a positive integer M_n, let $T_m^{(n)}$, $1 \leq m \leq M_n$, be the random observation times for the curve $X^{(n)}$. These times are obtained as independent copies of a variable T taking values in I. The integers $M_1, \ldots, M_N$ represent an independent sample of an integer-valued random variable M with expectation μ which increases with N. Thus $M_1, \ldots, M_N$ is the Nth line in a triangular array of integer numbers. We assume that the realizations of X, M and T are mutually independent. The observations associated with a curve, or trajectory, $X^{(n)}$ consist of the pairs $(Y_m^{(n)}, T_m^{(n)}) \in \mathbb{R} \times I$ where $Y_m^{(n)}$ is defined as

$$Y_m^{(n)} = X^n(T_m^{(n)}) + \varepsilon_m^{(n)}, \qquad 1 \leq n \leq N, \;\; 1 \leq m \leq M_n, \tag{2.1}$$

and $\varepsilon_m^{(n)}$ are independent copies of a centered error variable ε. For the sake of readability, here and in the following, we use the notation X_t for the value at t of the generic process X and $X^{(n)}(t)$ for the value at t of the realization $X^{(n)}$ of X. The $N-$sample of X is composed of two sub-populations: a *learning* set of N_0 curves and a set of N_1 curves to be recovered that we call the *online* set. Thus, $1 \leq N_0, N_1 < N$ and $N_0 + N_1 = N$. Let $X^{(1)}, \ldots, X^{(N_0)}$ denote the curves corresponding to the *learning* set.

Our first aim is to define a meaningful concept of local regularity for the trajectories of X and to build an estimator for it. The estimator could be computed easily and rapidly from the observations $(Y_m^{(n)}, T_m^{(n)})$ corresponding to the curves in the *learning* set, and does not require a very large number N_0 of curves. Moreover, it could be easily updated if more curves are added to the *learning* set. The problem of estimating the regularity of the trajectories is related to the estimation of the Hausdorff, or fractal, dimension of time series. See, for instance, [33, 30, 54] and the references therein. However, herein, we adopt the FDA point of view and use the so-called *replication* and *regularization* features

of functional data (see [108], ch.22). More precisely, we combine information both across and within curves. Thus, taking strength from the information contained in the whole set of N_0 available time series, we are able to investigate more general situations: X need not to be a Gaussian, or a transformed Gaussian process, it is not necessarily stationary or with stationary increments, it could have a fractal dimension which changes over time, it is observed with possibly heteroscedastic noise, at random moments in time.

Based on the regularity estimates, our second objective is to build an adaptive, nearly optimal smoothing for a possibly very large set of N_1 new curves. Let

$$X^{[1]} = X^{(N_0+1)}, \ldots, X^{[N_1]} = X^{(N)},$$

denote the curves from the *online* set to be recovered from the corresponding observations $(Y_m^{(n)}, T_m^{(n)})$. In the following, $t_0 \in I$ is an arbitrarily fixed point. The aim is thus to estimate $X^{[1]}(t_0), \ldots, X^{[N_1]}(t_0)$. This issue is a nonparametric estimation problem and, if each curve regularity is given, nonparametric estimators of the curves $X^{[1]}, \ldots, X^{[N_1]}$ could be easily built, for instance using the local linear smoother or the series estimator. Nevertheless in applications, there is no reason to suppose that the sample paths of the random process X have a known regularity at t_0. When it is not reasonable to assume a given regularity for the trajectories, one could use one of the existing data-driven procedures for determining the optimal smoothing parameter. However, the existing procedures, such as the cross-validation or the Goldenshluger-Lepski method [55], were designed for the case where one observes only one curve. Thus one has to apply them for each curve separately, which could require large amounts of resources.

In Section 2.2, we define the local regularity and provide concentration bounds for the estimator of the local regularity of the trajectories of X. Our results are new and of non-asymptotic type, in the sense that they hold for any values of the sample sizes N_0 and the mean value of observation times μ, provided these values are sufficiently large. In Section 2.3, we explain the relationship between the probabilistic concept of local regularity for the trajectory of X and the analytic regularity of the curves which usually determines the optimal risk rate in nonparametric estimation. Given the estimate of the local regularity of the trajectories of X, in Section 2.3, we also provide a non-asymptotic bound for the pointwise risk of the local polynomial smoother, uniformly over the *online* set. This uniform bound is obtained using an exponential-type moment bound for the pointwise risk for the local polynomial smoother, a new result of interest in itself. The

pointwise risk bound is optimal, in the nonparametric regression estimation sense, up to some logarithmic factors induced by our stochastic curves model, the concentration of the local regularity estimator, and the uniformity over the *online* set. In Section 2.4, we provide some additional guidance for the implementation of the local polynomial smoother and report results from simulation showing that our estimator perform well. A real data application on vehicle traffic flow analysis illustrates the effectiveness of our approaches. The proofs of our results are postponed to the Appendices 2.A and 2.B. Additional technical proofs, simulation results, and details on traffic flow application are also relegated to the Appendices 2.C–2.F . To further illustrate the irregularity of the curves in applications, we also report in the Appendix 2.G the local regularity estimates for another three functional data sets often analyzed in the literature.

2.2 Local regularity estimation

The new local regularity estimator is introduced and studied in this section. After providing some insight into the ideas behind the construction, we provide a concentration result for our estimator under general mild assumptions which do not impose a specific distribution for X. In particular, X could, but need not, be a Gaussian process. The case where the variance of the noise is not constant is also discussed.

2.2.1 The methodology

Let us present the main ideas behind the construction of the regularity estimate. For this, let us introduce some more notation used throughout the paper. Let K_0 be an integer value which will be defined below, and consider the order statistics of a M-sample $T_1, \ldots, T_M$ distributed as T which admits the density f. Let $t_0 \in I$ such that $f(t_0) > 0$. We extract the subvector of the K_0 closest values to t_0 and denote these values $T_{(1)} \leq \ldots \leq T_{(K_0)}$. If $t_0 = \inf(I)$ then $t_0 \leq T_{(1)}$, while if $t_0 = \sup(I)$, then $T_{(K_0)} \leq t_0$. When t_0 is an interior point of I, t_0 likely lies between $T_{(1)}$ and $T_{(K_0)}$. Next, we define the interval

$$J_\mu(t_0) = \Big(t_0 - |I|/\log(\mu), t_0 + |I|/\log(\mu)\Big) \cap I,$$

where $|I|$ denotes the length of the interval I and, recall, μ is the expectation of M.

We assume that the process X generating the continuous curves $X^{(1)}, \dots, X^{(N)}$ satisfies

$$\mathbb{E}\left[(X_u - X_v)^2\right] \approx L_{t_0}^2 |v-u|^{2H_{t_0}}, \quad u, v \in J_\mu(t_0), \tag{2.2}$$

for some $H_{t_0} \in (0,1]$. Here and in the following, $\approx$ means the left-hand side is equal to the right-hand side times a quantity which tends towards 1 when μ increases. When the trajectories of X are not differentiable, H_{t_0} is what we call the *local regularity of the process X at t_0*. For now, we focus on this case. When, with probability 1, the trajectories of X admits derivatives of order $\mathbf{d} \geq 1$ in a neighborhood of t_0, the property (2.2) will be used for the derivative of order $\mathbf{d}$ of the smooth trajectories. In this smooth case, the local regularity of the process X at t_0 will be $\mathbf{d} + H_{t_0}$. See Section 2.2.3.

To construct our estimator of H_{t_0}, we consider the event

$$\mathcal{B} = \{M \geq K_0, T_{(1)} \in J_\mu(t_0), \dots, T_{(K_0)} \in J_\mu(t_0)\},$$

which is expected to be of high probability. Let $\mathbf{1}_{\mathcal{B}}$ denote the indicator of $\mathcal{B}$ and let us define the expectation operator

$$\mathbb{E}_{\mathcal{B}}(\cdot) = \mathbb{E}(\cdot \mathbf{1}_{\mathcal{B}}).$$

Using (2.2) and the independence between X and T, for any $1 \leq k < l \leq K_0$,

$$\mathbb{E}_{\mathcal{B}}\left[(X_{T_{(l)}} - X_{T_{(k)}})^2\right] \approx L_{t_0}^2 \mathbb{E}_{\mathcal{B}}\left(|T_{(l)} - T_{(k)}|^{2H_{t_0}}\right).$$

From this and the moments of the spacing $T_{(l)} - T_{(k)}$ as given in the Lemma 2, we obtain

$$\mathbb{E}_{\mathcal{B}}\left[(X_{T_{(l)}} - X_{T_{(k)}})^2\right] \approx L_{t_0}^2 \left(\frac{l-k}{f(t_0)(\mu+1)}\right)^{2H_{t_0}}.$$

Now, for any $1 \leq k \leq K_0$, let $\varepsilon_{(k)}$ be a generic error term corresponding to the generic realization $X_{T_{(k)}}$, and denote

$$Y_{(k)} = X_{T_{(k)}} + \varepsilon_{(k)}.$$

Moreover, for k such that $2k-1 \leq K_0$, let

$$\theta_k = \mathbb{E}_{\mathcal{B}}\left[(Y_{(2k-1)} - Y_{(k)})^2\right].$$

We then obtain

$$\frac{\theta_k - 2\sigma^2}{L_{t_0}^2} \approx \left(\frac{k-1}{f(t_0)(\mu+1)}\right)^{2H_{t_0}}. \tag{2.3}$$

We distinguish two situations : the case where σ^2 is known and the case where it is unknown. In the former case, we suppose that $4k-3$ is also less than K_0 and use twice the relationship (2.3) with k and $2k-1$, respectively. We deduce

$$\frac{\theta_{2k-1} - 2\sigma^2}{\theta_k - 2\sigma^2} \approx 4^{H_{t_0}}.$$

Taking the logarithm on both sides, we obtain the proxy value

$$H_{t_0}(k, \sigma^2) = \frac{\log(\theta_{2k-1} - 2\sigma^2) - \log(\theta_k - 2\sigma^2)}{2\log 2},$$

of the local regularity parameter H_{t_0}, when σ^2 is given. In the case where σ^2 is unknown, assuming that $8k-7 \leq K_0$, we use the relationship (2.3) three times with k, $2k-1$ and $4k-3$, respectively, to obtain

$$\frac{\theta_{4k-3} - \theta_{2k-1}}{\theta_{2k-1} - \theta_k} \approx 4^{H_{t_0}}.$$

A natural proxy of H_{t_0} is then given by

$$H_{t_0}(k) = \frac{\log(\theta_{4k-3} - \theta_{2k-1}) - \log(\theta_{2k-1} - \theta_k)}{2\log 2}. \tag{2.4}$$

Our estimator of the local regularity parameter H_{t_0} is the empirical version of the proxy value $H_{t_0}(k)$, or $H_{t_0}(k, \sigma^2)$, built from a random sample of N_0 trajectories of X, the learning set of curves. Formally, we consider the sequence of events

$$\mathcal{B}_n = \mathcal{B}_n(\mu, N_0) = \left\{M_n \geq K_0, T_{(1)}^{(n)} \in J_\mu(t_0), \ldots, T_{(K_0)}^{(n)} \in J_\mu(t_0)\right\}, \quad 1 \leq n \leq N_0, \tag{2.5}$$

and we define

$$\hat{\theta}_{2k-1} = \frac{1}{N_0}\sum_{n=1}^{N_0}\left[Y_{(4k-3)}^{(n)} - Y_{(2k-1)}^{(n)}\right]^2 \mathbf{1}_{\mathcal{B}_n} \quad \text{and} \quad \hat{\theta}_k = \frac{1}{N_0}\sum_{n=1}^{N_0}\left[Y_{(2k-1)}^{(n)} - Y_{(k)}^{(n)}\right]^2 \mathbf{1}_{\mathcal{B}_n}, \tag{2.6}$$

where, for any n and k, $Y_{(k)}^{(n)}$ denotes the noisy measurement of $X^{(n)}(T_{(k)}^{(n)})$. If $H_{t_0}(k, \sigma^2)$ is

indeed a good approximation of H_{t_0}, a simple estimator of H_{t_0} when σ^2 is known is then

$$\widehat{H}_{t_0}(k,\sigma^2) = \begin{cases} \dfrac{\log(\hat{\theta}_{2k-1} - 2\sigma^2) - \log(\hat{\theta}_k - 2\sigma^2)}{2\log 2} & \text{if} \quad \min(\hat{\theta}_{2k-1}, \hat{\theta}_k) > 2\sigma^2 \\ 1 & \text{otherwise.} \end{cases}$$

When σ^2 is unknown the corresponding estimator is

$$\widehat{H}_{t_0}(k) = \begin{cases} \dfrac{\log(\hat{\theta}_{4k-3} - \hat{\theta}_{2k-1}) - \log(\hat{\theta}_{2k-1} - \hat{\theta}_k)}{2\log 2} & \text{if } \hat{\theta}_{4k-3} > \hat{\theta}_{2k-1} > \hat{\theta}_k \\ 1 & \text{otherwise,} \end{cases} \tag{2.7}$$

where $\hat{\theta}_{4k-3}$ is obtained from the formula of $\hat{\theta}_{2k-1}$ after replacing k by $2k-1$.

It is worth noting that our estimator could be easily updated every time new curves are included in the learning sample, without revisiting the learning set already used. Indeed, one should only add new terms in the sums defining $\hat{\theta}_k$, $\hat{\theta}_{2k-1}$ and $\hat{\theta}_{4k-3}$.

2.2.2 Concentration bounds for the local regularity estimator

Below, we focus on the more complicated and realistic case with unknown variance. The case with given variance could be treated after obvious adjustments. The results in this section depend on μ, the mean number of observation times T, and the cardinality N_0 of the learning set of curves. However, they are non-asymptotic in the sense that they hold true for any sufficiently large μ and N_0 satisfying our conditions. Whenever it exists, let $X_u^{(\mathbf{d})}$ denote the $\mathbf{d}-$th derivative, $\mathbf{d} \geq 1$, of the generic curve X_u at the point u. By definition $X_u^{(0)} \equiv X_u$. For deriving our results, we impose the following mild assumptions.

(H1) The data consist of the pairs $(Y_m^{(n)}, T_m^{(n)}) \in \mathbb{R} \times I$ defined as in (3.1), with $I \subset \mathbb{R}$ a compact interval, and the realizations of X, M and T are mutually independent.

(H2) The random variable T admits a density $f : I \to \mathbb{R}$ such that $f(t_0) > 0$. Moreover, there exist $L_f > 0$ and $0 < \beta_f \leq 1$ such that

$$|f(u) - f(v)| \leq L_f |u - v|^{\beta_f}, \qquad \forall u, v \in J_\mu(t_0).$$

(H3) For some integer $\mathbf{d} \geq 0$, there exist a function $\phi_{t_0}(\cdot,\cdot) > 0$, the constants $L_{t_0}, L_\phi > 0$

and $0 < \beta_\phi \leq 1$ such that, for any $u, v \in J_\mu(t_0)$, we have

$$\mathbb{E}\left[(X_u^{(\mathbf{d})} - X_v^{(\mathbf{d})})^2\right] = L_{t_0}^2 |u-v|^{2H_{t_0}} \{1 + \phi_{t_0}(u,v)\} \quad \text{and} \quad |\phi_{t_0}(u,v)| \leq L_\phi |u-v|^{\beta_\phi}.$$

(H4) For $\mathbf{d} \geq 0$ in Assumption (H3), two constants $\mathfrak{a}, \mathfrak{A} > 0$ exist such that

$$\mathbb{E}\left[|X_u^{(\mathbf{d})} - X_v^{(\mathbf{d})}|^{2p}\right] \leq \left(\frac{p!}{2}\mathfrak{a}\mathfrak{A}^{p-2}\right)|u-v|^{2pH_{t_0}}, \qquad \forall p \geq 2,\ \forall u, v \in J_\mu(t_0).$$

(H5) The variables $\varepsilon_m^{(n)}$, $n, m \geq 1$, are independent copies of a centered variable ε, with finite variance σ^2, for which constants $\mathfrak{b} \geq \sigma^2 > 0$ and $\mathfrak{B} > 0$ exist such that

$$\mathbb{E}(|\varepsilon|^{2p}) \leq \frac{p!}{2}\mathfrak{b}\mathfrak{B}^{p-2}, \qquad \forall p \geq 1.$$

(H6) The random variable M is such that $M \geq 9$ and $\gamma_0 > 0$ exists such that, for any $s > 0$, $\mathbb{P}(|M - \mu| > s) \leq \exp(-\gamma_0 s)$.

Assumption (H2) imposes a mild condition on the distribution of the random observation points which provides convenient moment bounds for their spacings. In particular, it implies that, for a sufficiently large μ, $f(t_0)/2 \leq f(t) \leq 2f(t_0)$, $\forall t \in J_\mu(t_0)$. Assumption (H3) is a version of the so-called local stationarity condition, here considered for the $\mathbf{d}-$th derivative process. More precisely, (H3) implies that the trajectories of $X^{(\mathbf{d})} = (X_u^{(\mathbf{d})} : u \in I)$ are Hölder continuous in quadratic mean in the neighborhood of t_0, with exact exponent H_{t_0} and local Hölder constant L_{t_0}. Let us call $\varsigma_{t_0} = \mathbf{d} + H_{t_0}$ the *local regularity of the process* X *at* t_0. Examples include, but are not limited to, stationary or stationary increment processes X. See, *e.g.*, [7] for some examples and references on processes satisfying the mild condition in (H3). Assumptions (H4) and (H5) are needed for deriving exponential bounds for the concentration of our local regularity estimator, while (H6) is a mild condition for controlling the variability of number of observation points on the curves. The lower bound on M guarantees that each curve in the learning set has a sufficient number of observation times for building our estimator.

We first consider the case $\mathbf{d} = 0$, the general case being analyzed in Section 2.2.3. For a real number a, let $\lfloor a \rfloor$ denote the largest integer not exceeding a.

Theorem 1. *Let Assumptions (H1)–(H6) hold true with* $\mathbf{d} = 0$. *Let* μ *and* K_0 *be positive integers such that*

$$(\mu+1)^{\frac{\beta_f \alpha}{4+\beta_f \alpha}} \leq K_0 \leq \frac{\mu}{2\log(\mu)}, \tag{2.8}$$

with $\alpha = 2H_{t_0} + \beta_\phi \in (0, 3]$.

Let

$$\mathfrak{c}\left(\frac{K_0 - 1}{f(t_0)(\mu+1)}\right)^{\min(\beta_\phi, \beta_f H_{t_0}/2)} < \epsilon < \frac{2}{\log 2}, \tag{2.9}$$

with $\mathfrak{c}$ *a constant depending only on* L_f, β_f, β_ϕ, $f(t_0)$ *and* H_{t_0}. *Define* $k = \lfloor (K_0+7)/8 \rfloor$ *and let* $\widehat{H}_{t_0} = \widehat{H}_{t_0}(k)$ *be defined as in (2.7). Then, for a sufficiently large* μ *:*

$$\mathbb{P}\left(\left|\widehat{H}_{t_0} - H_{t_0}\right| > \epsilon\right) \leq 12\exp\left[-\mathfrak{f} N_0 \epsilon^2 \left(\frac{k-1}{f(t_0)(\mu+1)}\right)^{4H_{t_0}}\right],$$

where $\mathfrak{f}$ *is a positive constant depending on* $\mathfrak{a}, \mathfrak{A}, \mathfrak{b}, \mathfrak{B}$ *and the length of the interval* I.

To obtain a non-trivial estimator of H_{t_0}, we need $k \geq 2$, thus the upper bound in (2.8) should be larger than 9, and this happens as soon as $\mu \geq 80$. The exact expressions of the constants $\mathfrak{c}$ and $\mathfrak{f}$ could be traced in the proof of Theorem 1. The condition imposed on K_0 provides a panel of choices depending on N_0 and μ. As a result, up to some constants, and depending on L_f, β_f, β_ϕ, $f(t_0)$ and H_{t_0}, the concentration rate ϵ, one could expect, could be in a range such that $\epsilon\mu \gg 1$ and $\epsilon \log^{1/2}(\mu) \ll 1$. The best possible concentration of $\widehat{H}_{t_0}$ is guaranteed as soon as N_0 is larger than some power of μ, while for a concentration as fast as some negative power of $\log(\mu)$, one only needs a small number N_0 of curves in the learning set, that is larger than some power of $\log(\mu)$.

For the purpose of building an adaptive optimal kernel estimator for the trajectories of X, we will impose $\epsilon \leq \log^{-2}(\mu)$ and an exponential bound equal to $\exp(-\mu)$. The following corollary proposes a data-driven choice of K_0 which guarantees these requirements. This choice is guided by the fact that, for any constants $a, b > 0$, we have the relationship $\log^a(\mu) \leq \exp((\log\log(\mu))^2) \leq \mu^b$, provided μ is sufficiently large.

Corollary 1. *Assume the conditions of Theorem 1 hold true. Let* $\widehat{\mu} = {N_0}^{-1}\sum_{n=1}^{N_0} M_n$,

$$\widehat{K_0} = \lfloor \widehat{\mu}\exp(-(\log\log(\widehat{\mu}))^2) \rfloor,$$

and $\widehat{H}_{t_0} = \widehat{H}_{t_0}(\lfloor(\widehat{K}_0 + 7)/8\rfloor)$, *with* $\widehat{H}_{t_0}$ *defined in* (2.7). *Then, for any constant* $C > 0$,

$$\mathbb{P}\left(\left|\widehat{H}_{t_0} - H_{t_0}\right| > C\log^{-2}(\mu)\right) \leq \exp(-\mu),$$

provided $N_0 \geq \mu^{1+b}$ *for some* $b > 0$ *and* μ *is sufficiently large.*

One could also build $\widehat{H}_{t_0}$ with only one trajectory of a stochastic process X with stationary increments. If the density of T is uniform and sufficiently many measurements are available, it suffices to split the interval $[0, 1]$ into N_0 intervals of the same length and apply our methodology considering the measuring times and the noisy measured values in each block as belonging to a different curve in the learning set. Theorem 1 and Corollary 1 remain valid.

2.2.3 The case of smooth trajectories

We can extend our learning approach for the local regularity of the trajectories of X to the case of smooth trajectories, *i.e.*, when almost all trajectories admit derivatives. For $s \in (0, 1]$, let $\mathcal{X}(\mathbf{d}, s)$ denote the class of stochastic processes X such that, with probability 1, the trajectories admit derivatives of order $\mathbf{d} \geq 1$ and $X^{(\mathbf{d})} = (X_u^{(\mathbf{d})} : u \in I)$ satisfies the Assumptions (H3) and (H4) with $H_{t_0} = s$. Let us point out that if $X \in \mathcal{X}(\mathbf{d}, s)$, for some $\mathbf{d} \geq 1$ and $s \in (0, 1]$, and the first derivative X'_{t_0} has an exponential moment, *i.e.*, $\mathbb{E}[\exp(cX'_{t_0})] < \infty$ for some $c > 0$, then $X \in \mathcal{X}(0, 1)$. Moreover, X cannot belong to any of the classes $\mathcal{X}(0, \overline{s})$ with $\overline{s} \in (0, 1)$. Therefore, when $\mathbf{d} \geq 1$, our estimator $\widehat{H}_{t_0}$ is expected to concentrate to the value 1. Meanwhile, when X belongs to $\mathcal{X}(0, H_{t_0})$ for some $H_{t_0} \in (0, 1)$, $\widehat{H}_{t_0}$ concentrates to H_{t_0}. This remark suggests a simple check of the composite null hypothesis

$$\mathcal{H}_0 : \ X \in \mathcal{X}(0, H_{t_0}) \text{ for some } H_{t_0} \in (0, 1),$$

against the alternative hypothesis

$$\mathcal{H}_1 : \ X \in \mathcal{X}(\mathbf{d}, s) \text{ for some } \mathbf{d} \geq 1 \text{ and } s \in (0, 1],$$

defined by the rule

$$\text{rejects } \ \mathcal{H}_0 \ \text{ if } \ \widehat{H}_{t_0} > 1 - \log^{-2}(\widehat{\mu}),$$

where $\widehat{H}_{t_0} = \widehat{H}_{t_0}(\lfloor(\widehat{K}_0+7)/8\rfloor)$ is defined as in (2.7) with $\widehat{K}_0 = \lfloor\widehat{\mu}\exp(-(\log\log(\widehat{\mu}))^2)\rfloor$ and $\widehat{\mu} = {N_0}^{-1}\sum_{n=1}^{N_0} M_n$. The following result guarantees that this test procedure is consistent.

Corollary 2. *Assume that* $\log^{-1}(\mu)\mu \leq M \leq \mu\log(\mu)$ *almost surely. Under the conditions of Corollary 1, if* $N_0 \geq \mu^{1+b}$*, for some* $b > 0$*, and sufficiently large* μ*, then under* $\mathcal{H}_0$*,*

$$\mathbb{P}\left(\widehat{H}_{t_0} > 1 - \log^{-2}(\widehat{\mu})\right) \leq \exp(-\mu),$$

whereas under $\mathcal{H}_1$*, if* $\mathbb{E}[\exp(cX'^{t_0})] < \infty$ *for some* $c > 0$*,*

$$\mathbb{P}\left(\widehat{H}_{t_0} > 1 - \log^{-2}(\widehat{\mu})\right) \geq 1 - \exp(-\mu).$$

Corollary 2 is a direct consequence of Corollary 1 and of the mild simplifying condition on the support of M, which implies that $\log(\mu)/2 \leq \log(\widehat{\mu}) \leq 2\log(\mu)$. We hence omit the details. In the case of smooth trajectories, the first step is to detect that the trajectories are differentiable. Corollary 1 indicates that, for a suitable choice of $\widehat{K}_0$ as a function of $\widehat{\mu}$, the event $\{\widehat{H}_{t_0}(\lfloor(\widehat{K}_0+7)/8\rfloor) > 1-\log^{-2}(\widehat{\mu})\}$ has a very low probability, provided that X belongs to $\mathcal{X}(0, H_{t_0})$ and N_0 is as large as a power larger than 1 of μ. Based on Corollary 2, in practice, one could simply check the condition

$$\widehat{H}_{t_0}(\lfloor(\widehat{K}_0+7)/8\rfloor) \leq 1 - \log^{-2}(\widehat{\mu}).$$

If this condition fails, ideally one would like to consider the first derivative trajectories of X, build a new estimator $\widehat{H}_{t_0}$ with these trajectories, and again test $\mathcal{H}_0$ with $\mathcal{X}(1, H_{t_0})$ instead of $\mathcal{X}(0, H_{t_0})$. If $\mathcal{H}_0$ is not rejected, one could define $\widehat{\varsigma}_{t_0} = 1 + \widehat{H}_{t_0}$, while, if $\mathcal{H}_0$ is rejected, one would consider the second order derivative trajectories of X, build a new estimator $\widehat{H}_{t_0}$ with these trajectories, and so on. Since the derivatives of the trajectories of X are not available, in Section 2.4.1, we propose a sequential algorithm to estimate them, $\mathbf{d}$ and H_{t_0}, in the case of local regularities ς_{t_0} larger than 1.

2.2.4 The case of conditionally heteroscedastic noise

In some applications, the assumption of constant variance for the error term ε could be unrealistic. Therefore, we consider the following conditional heteroscedastic error extension of model (3.1):

$$Y_m^{(n)} = X^n(T_m^{(n)}) + \sigma\left(X^n(T_m^{(n)}), T_m^{(n)}\right) u_m^{(n)}, \qquad 1 \leq n \leq N, \;\; 1 \leq m \leq M_n, \tag{2.10}$$

where $\sigma(\cdot,\cdot)$ is some unknown function and $u_m^{(n)}$ are independent copies of a centered variable u with unit variance.

Our approach also applies to the model (2.10) under some additional mild conditions. Indeed, assuming the expectations exist, we have

$$\theta_k = \mathbb{E}_{\mathcal{B}}\left[(Y_{(2k-1)} - Y_{(k)})^2\right] = \mathbb{E}_{\mathcal{B}}\left[(X_{T_{(2k-1)}} - X_{T_{(k)}})^2\right] + \mathbb{E}_{\mathcal{B}}\left[\sigma^2\left(X_{T_{(2k-1)}}, T_{(2k-1)}\right)\right] + \mathbb{E}_{\mathcal{B}}\left[\sigma^2\left(X_{T_{(k)}}, T_{(k)}\right)\right].$$

From this identity it is clear that the arguments presented in Section 2.2.1 remain valid as long as the last two expectations on the right-hand side of the last display are equal and their value does not depend on k. Thus, in this case, even if the conditional variance of $\varepsilon_m^{(n)}$ is not given, we could consider the same estimator $\widehat{H}_{t_0}$. This remark leads us to the following additional assumption.

(E1) The variables $u_m^{(n)}$ from model (2.10) satisfy the Assumption (H5) with unit variance. Moreover, the function $\sigma(\cdot,\cdot)$ is bounded and the map $u \mapsto \mathbb{E}\left[\sigma^2(X_u, u)\right]$, $u \in I$, is constant in a fixed neighborhood of t_0.

Assumption (E1) allows the error term to be conditionally heteroscedastic, but imposes marginal (unconditional) homoscedasticity in a neighborhood of t_0. Under Assumption (E1), for any k we have

$$\mathbb{E}_{\mathcal{B}}\left[\sigma^2(X_{T_{(k)}}, T_{(k)})\right] = \mathbb{E}\left[\mathbb{E}\left(\sigma^2(X_{T_{(k)}}, T_{(k)}) \mid M, T_1, T_2, \ldots, T_M\right)\mathbf{1}_{\mathcal{B}}\right] = \mathbb{E}\left[\sigma^2(X_u, u)\right]\mathbb{P}(\mathcal{B}),$$

and thus the terms like $\mathbb{E}_{\mathcal{B}}[\sigma^2(X_{T_{(k)}}, T_{(k)})]$ cancel when considering the differences $\theta_{4k-3} - \theta_{2k-1}$ and $\theta_{2k-1} - \theta_k$.

Corollary 3. *Assume the observations consist of the pairs $(Y_m^{(n)}, T_m^{(n)}) \in \mathbb{R} \times I$ where $Y_m^{(n)}$ defined as in (2.10) and the realizations of X, M and T are mutually independent. Assume that Assumptions (H2)–(H4), (H6), (E1) hold. Then Corollaries 1 and 2 remain valid with the same local regularity estimator $\widehat{H}_{t_0}$.*

The proof of Corollary 3 could be obtained from the proof of Theorem 1 after obvious modifications, and hence will be omitted. It is worthwhile noting that, even if the regularity H_{t_0} is the same at any point t_0, one may not be able to estimate the regularity H_{t_0}

using only one observed noisy trajectory with conditionally heteroscedastic noise. This because, intuitively, it might be impossible to identify the oscillations of the signal of interest, that is to separate the increments of the trajectory of X from the differences of the error terms with variable variance. With our approach based on local observed increments averaged over several curves, the effect of the noise vanishes, provided the expectation of the conditional variance is constant. Hence, eventually the identification of the oscillations of X is recovered and there is no difference with respect to the case of homoscedastic errors.

2.3 Adaptive optimal smoothing

With at hand an estimate of the local regularity $\varsigma_{t_0} = \mathbf{d} + H_{t_0}$ obtained from a learning set of N_0 curves, we aim at recovering N_1 new noisy trajectories of X from what we call the online dataset. One of the most popular smoother is the local polynomial estimator, see [45]. This type of estimator crucially depends on a tuning parameter, the bandwidth, which should ideally be chosen according to the unknown regularity of the target function.

One has to connect a definition of local regularity that is meaningful from the theory of stochastic processes to the usual definition of function regularity used in nonparametric curve estimation. Fortunately, in our framework, the parameter ς_{t_0}, which is understood as the local regularity of the process $(X_t : t \in J_\mu(t_0))$ in *quadratic mean*, see (2.2), is intrinsically linked with the regularity of the sample paths of the process. Indeed, in many important situations, which are covered by our assumptions, the regularity of the sample paths of a process does not depend on the realization of this process. For example, the regularity of any Brownian path is $1/2$, in the sense that for any $\epsilon > 0$, almost surely the sample path belongs to the Hölder space $\mathcal{C}^{1/2-\epsilon}(I)$ and does not belong to $\mathcal{C}^{1/2+\epsilon}(J)$ whatever $J \subset I$. Here, for any $a > 0$, $\mathcal{C}^a(I)$ denotes the space of uniformly $a-$Hölder continuous functions defined on I, see Theorem 2.2 and Corollary 2.6 of [116] for precise definition. More generally the regularity of the sample paths of a process is linked to integrated regularities through the Kolmogorov's Continuity Theorem [116, Theorem 2.1]. In particular, Assumption (H3) ensures that, with probability 1, the trajectories of the process $(X_t^{(\mathbf{d})} : t \in J_\mu(t_0))$ are Hölder continuous with any exponent parameter $0 < a < H_{t_0}$.

Below, we define the local polynomial estimator and derive its theoretical properties. Since our focus of interest is the *simultaneous* denoising of the additional N_1 curves, we

consider the following pointwise risk: for a generic estimator $\widehat{X}_{t_0}^{[n_1]}$ of $X_{t_0}^{[n_1]}$, let

$$\mathcal{R}(\widehat{X}; t_0) = \mathbb{E}\left[\max_{1 \le n_1 \le N_1} \left|\widehat{X}_{t_0}^{[n_1]} - X_{t_0}^{[n_1]}\right|^2\right]. \tag{2.11}$$

First, we provide a sharp bound for this risk with $N_1 = 1$, in the case where a suitable estimator of $\varsigma_{t_0} = \mathbf{d} + H_{t_0}$, computed from another independent sample, is given. Such a result, of interest in itself in nonparametric curve estimation, seems to be new. In this case, the expectation defining the risk $\mathcal{R}(\widehat{X}; t_0)$ should be understood as the conditional expectation given the estimator of ς_{t_0}. Next, we provide a sharp bound for $\mathcal{R}(\widehat{X}; t_0)$ in the case where $N_1 \ge 1$ and the estimator of ς_{t_0} is obtained using the approach introduced in Section 2.2.

2.3.1 Local polynomial estimation

We assume that $\mathbf{d} \ge 0$ is an integer and $H_{t_0} \in (0, 1)$. Let $\hat{\mathbf{d}}$ and $\widehat{H}_{t_0}$ be some generic estimators of $\mathbf{d}$ and H_{t_0}, respectively, and let $\widehat{\varsigma}_{t_0} = \hat{\mathbf{d}} + \widehat{H}_{t_0}$ be the corresponding estimator of $\varsigma_{t_0} = \mathbf{d} + H_{t_0}$. We assume that $\hat{\mathbf{d}}$ and $\widehat{H}_{t_0}$ are independent of the N_1 from the online dataset, generated according to (3.1).

The estimator of ς_{t_0} could be used to smooth any curve $Y^{[n_1]}$ $(n_1 = 1, \ldots, N_1)$ from the online dataset. For the sake of readability, we omit the superscript $[n_1]$ and we consider a generic curve from the online dataset:

$$Y_m = X(T_m) + \varepsilon_m, \qquad 1 \le m \le M.$$

For any $u \in \mathbb{R}$, we consider the vector $U(u) = (1, u, \ldots, u^{\hat{\mathbf{d}}}/\hat{\mathbf{d}}!)$. Let $K : \mathbb{R} \to \mathbb{R}$ be a positive kernel and define:

$$\vartheta_{M,h} = \operatorname*{arg\,min}_{\vartheta \in \mathbb{R}^{\mathbf{d}+1}} \sum_{m=1}^{M} \left\{Y_m - \vartheta^\top U\left(\frac{T_m - t_0}{h}\right)\right\}^2 K\left(\frac{T_m - t_0}{h}\right),$$

where h is the bandwidth. The vector $\vartheta_{M,h}$ satisfies the normal equations $A\vartheta_{M,h} = a$ with

$$A = A_{M,h} = \frac{1}{Mh} \sum_{m=1}^{M} U\left(\frac{T_m - t_0}{h}\right) U^\top\left(\frac{T_m - t_0}{h}\right) K\left(\frac{T_m - t_0}{h}\right) \tag{2.12}$$

$$a = a_{M,h} = \frac{1}{Mh} \sum_{m=1}^{M} Y_m U\left(\frac{T_m - t_0}{h}\right) K\left(\frac{T_m - t_0}{h}\right). \tag{2.13}$$

Let λ be the smallest eigenvalue of the matrix A and remark that, whenever $\lambda > 0$, we have $\vartheta_{M,h} = A^{-1}a$.

Taking into account the expression of the bandwidth minimizing the pointwise mean squared risk for a regression function defined on I, with derivative of order $\mathbf{d}$ which is Hölder continuous in a neighborhood of t_0, with exact exponent H_{t_0}, we consider the bandwidth

$$\widehat{h} = \left(\frac{1}{M}\right)^{1/(2\widehat{\varsigma}_{t_0}+1)}.$$

Our focus of interest is on determining a nearly optimal rate of the bandwidth to be used to recover the trajectories of X. For the applications, one could also be interested in a nearly optimal constant, which in general needs to be estimated. In Section 2.4.1 we propose a simple way to estimate a suitable constant for the applications.

With at hand the bandwidth $\widehat{h}$, we propose the following definition of the local polynomial estimator of X_{t_0} of order $\mathbf{d}$:

$$\hat{X}_{t_0} = \begin{cases} U^\top(0)\widehat{\vartheta} & \text{if } \lambda > \log^{-1}(M) \text{ and } |U^\top(0)\widehat{\vartheta}| \leq \widehat{\tau}^{5/12}(M) \\ \widehat{\tau}^{5/12}(M) & \text{if } \lambda > \log^{-1}(M) \text{ and } |U^\top(0)\widehat{\vartheta}| > \widehat{\tau}^{5/12}(M) \\ 0 & \text{otherwise,} \end{cases},$$

where $\widehat{\vartheta} = \vartheta_{M,\widehat{h}}$ and, for any $y > 1$,

$$\widehat{\tau}(y) = \frac{1}{\log^2(y)}\left(\frac{y}{\log(y)}\right)^{2\widehat{\varsigma}_{t_0}/(2\widehat{\varsigma}_{t_0}+1)}.$$

The upper trimming with $\widehat{\tau}^{5/12}(M)$ is a technical device used to control the tails of $\hat{X}_{t_0}$. It has practically no influence in applications. For deriving our results on $\hat{X}_{t_0}$, we impose the following mild assumptions.

(LP1) There exist two positive constants, $\mathfrak{a}$ and $\mathfrak{A}$, such that for any $p \geq 1$:

$$\mathbb{E}\left[|X_t|^{2p}\right] \leq \frac{p!}{2}\mathfrak{a}\mathfrak{A}^{p-2}, \quad \forall t \in [0,1].$$

(LP2) We assume that, almost surely, $\mu/\log(\mu) \leq M \leq \mu\log(\mu)$.

(LP3) The estimator $\widehat{H}_{t_0}$ satisfies the property

$$\mathbb{P}\left(\left|\widehat{H}_{t_0} - H_{t_0}\right| > \log^{-2}(\mu)\right) \leq \mathfrak{K}_1 \exp(-\mu), \qquad \forall \mu > 0,$$

where $\mathfrak{K}_1$ is some positive constant.

(LP4) The estimator $\hat{\mathbf{d}}$ satisfies the property $\mathbb{P}(\hat{\mathbf{d}} \neq \mathbf{d}) \leq \mathfrak{K}_1 \exp(-\mu)$, $\forall \mu > 0$.

Assumption (LP1) provides a suitably tight control on the moments of X_t, but still allows for unbounded trajectories. Assumption (LP2) is a convenient, but mild, technical condition. It could be relaxed at the price of controlling the probability of the complement of the event $\{\mu/\log(\mu) \leq M \leq \mu \log(\mu)\}$, for instance using (H6). Assumptions (LP3) and (LP4) are very mild conditions that the generic estimators of the regularity should satisfy. Since $\mu^{1/\log^2(\mu)} = e^{1/\log(\mu)}$ for any $\mu > 1$, the concentration of $\widehat{H}_{t_0}$ at a suitable negative power of $\log(\mu)$ will suffice for the smoothing purposes. For simplicity, and without loss of generality we consider the same constant $\mathfrak{K}_1$ in Assumptions (LP3) and (LP4).

Theorem 2. *Assume that Assumptions (H1), (H2), (H4)–(H6) and Assumptions (LP1)–(LP4) hold true and let $K(\cdot)$ be a kernel such that, for any $t \in \mathbb{R}$:*

$$\kappa^{-1}\mathbf{1}_{[-\delta,\delta]}(t) \leq K(t) \leq \kappa \mathbf{1}_{[-1,1]}(t), \quad \textit{for some } 0 < \delta < 1 \textit{ and } \kappa \geq 1. \tag{2.14}$$

There then exists a constant Γ_0 such that for any $\mu \geq 1$,

$$\mathbb{E}\left[\exp\left\{\left(\tau(\mu)\left|\hat{X}_{t_0} - X_{t_0}\right|^2\right)^{1/4}\right\}\right] \leq \Gamma_0 \quad \textit{where} \quad \tau(\mu) = \frac{1}{\log^2(\mu)}\left(\frac{\mu}{\log(\mu)}\right)^{\frac{2\varsigma_{t_0}}{2\varsigma_{t_0}+1}}.$$

The bound on the $\exp(\sqrt{x})-$moment of the $|\hat{X}_{t_0} - X_{t_0}|$ seems a new result for local polynomial estimators. For our purposes, it will entail a sharp bound for $\mathcal{R}(\widehat{X}; t_0)$. More precisely, the price for considering a risk measure uniformly over the whole online dataset is very low, that is a multiplying factor as large as a power of $\log(N_1)$ in the risk bound we derive below. [50] derived sharp bounds for all the moments of $|\hat{X}_{t_0} - X_{t_0}|$. However, his bounds on the moments would induce a power of N_1 as multiplying factor for our risk bound, instead of the power of $\log(N_1)$.

Theorem 3. *Assume that assumptions of Theorem 2 hold true, and let K be a kernel*

which satisfies (2.14)*. There then exists a positive constant* Γ_1 *such that*

$$\mathcal{R}(\widehat{X};t_0)=\mathbb{E}\Big[\max_{1\leq n_1\leq N_1}\big|\widehat{X}_{t_0}^{[n_1]}-X_{t_0}^{[n_1]}\big|^2\Big]\leq\Gamma_1\log^2(\mu)\log^4(1+N_1)\{\log(\mu)\}^{\frac{2\varsigma_{t_0}}{2\varsigma_{t_0}+1}}\mu^{-\frac{2\varsigma_{t_0}}{2\varsigma_{t_0}+1}}.$$

If all the trajectories X were in $\mathcal{C}^{\varsigma_{t_0}}(J_\mu(t_0))$, ς_{t_0} were known and $N_1=1$, the risk bound for $\mathcal{R}(\widehat{X};t_0)$ would be of the usual nonparametric rate $\mu^{-2\varsigma_{t_0}/(2\varsigma_{t_0}+1)}$. Let us note that the fact that H_{t_0} is not known does not have any consequence on the risk bound in Theorem 3. Indeed, since $\mu^{1/\log^2\mu}=e^{1/\log\mu}$ for any μ, the order of the risk bound does not change as soon as the probability of the event $\{\hat{\mathbf{d}}=\mathbf{d}\}\cap\{|\widehat{H}_{t_0}-H_{t_0}|\leq 1/\log^2\mu\}$ tends to 1. The $\log(1+N_1)$ factor is given by the maximum over the N_1 curves in the online dataset. The factor $\{\log(\mu)\}^{2\varsigma_{t_0}/(2\varsigma_{t_0}+1)}$ is due to the concentration properties of M around its mean μ. This factor would not appear if M/μ is almost surely bounded and bounded away from zero. The factor $\log^2(\mu)$ comes from probability theory. The trajectories of a stochastic process X with local regularity H_{t_0} does not necessarily belong to $\mathcal{C}^{\varsigma_{t_0}}(J_\mu(t_0))$ but they are almost surely in any $\mathcal{C}^{\varsigma_{t_0}-\epsilon}(J_\mu(t_0))$ for any $0<\epsilon<\varsigma_{t_0}$.

Finally, let us notice that Corollary 1 states that the estimator defined by (2.7) satisfies (H3) for $\mathbf{d}=0$ and any $0<H_{t_0}<1$. This leads us to the following result.

Corollary 4. *Assume* $\mathbf{d}=0$ *and let* $\widehat{H}_{t_0}$ *be the estimator of* $0<H_{t_0}<1$ *defined in Corollary 1. Moreover, Assumptions (H1)–(H6) and Assumptions (LP1)–(LP2) hold true. If* $N_0\geq\mu^{1+b}$ *and* $N_1\leq\mu^B$ *for some* $b,B>0$*, then*

$$\mathcal{R}(\widehat{X};t_0)\leq\Gamma_1 B^4\log^7(\mu)\mu^{-\frac{2H_{t_0}}{2H_{t_0}+1}}.$$

2.4 Empirical analysis

In the usual local polynomial (LP) smoothing framework, given a sample of size M, the optimal bandwidth minimizing the pointwise mean squared error risk for a regression function defined on I, with derivative of order $\mathbf{d}$ which is Hölder continuous in a neighborhood of t_0, with exact exponent H_{t_0} and local Hölder constant L_{t_0}, is

$$h_{opt}=\left(\frac{C}{M}\right)^{1/(2\varsigma_{t_0}+1)}\quad\text{with}\quad C=C_{t_0}=\frac{\sigma_{t_0}^2\|K\|^2\lfloor\varsigma_{t_0}\rfloor!}{\varsigma_{t_0}L_{t_0}\int|K(v)||v|^{\varsigma_{t_0}}dv},\tag{2.15}$$

where $\|K\|^2=\int K^2(v)dv$ and $\sigma_{t_0}^2$ is the variance of the noise that could depend on t_0. See for instance [135]. Based on the properties of the trajectories of X discussed above,

this is our target bandwidth. It depends on two more unknown quantities, L_{t_0} and $\sigma^2_{t_0}$, for which we now propose estimation procedures.

The estimation of L_{t_0} could be based on similar ideas as used for H_{t_0}. For simplicity, we assume $\mathbf{d} = 0$. The extension to the case $\mathbf{d} \geq 1$ could follow the same pattern as for the estimation of the local regularity, using the trajectories of the derivatives. Using twice the relationship (2.3) with k and $2k-1$, respectively, we deduce

$$L^2_{t_0} \approx \frac{\theta_{2k-1} - \theta_k}{4^{H_{t_0}} - 1} \left(\frac{f(t_0)(\mu+1)}{k-1} \right)^{2H_{t_0}}.$$

On the other hand, using the approximation of the moments of the spacings, as given in Lemma 2, we have

$$\begin{aligned} \eta_{2k-1} - \eta_k &:= \mathbb{E}_{\mathcal{B}} \left[\left| T_{(4k-3)} - T_{(2k-1)} \right|^{2H_{t_0}} \right] - \mathbb{E}_{\mathcal{B}} \left[\left| T_{(2k-1)} - T_{(k)} \right|^{2H_{t_0}} \right] \\ &\approx (4^{H_{t_0}} - 1) \left(\frac{k-1}{f(t_0)(\mu+1)} \right)^{2H_{t_0}}. \end{aligned}$$

Given an estimator of H_{t_0}, the empirical counterparts of η_k obtained from the learning set of N_0 independent trajectories of X is

$$\widehat{\eta}_k = \frac{1}{N_0} \sum_{n=1}^{N_0} \left| T^{(n)}_{(2k-1)} - T^{(n)}_{(k)} \right|^{2\widehat{H}_{t_0}} \mathbf{1}_{\mathcal{B}_n},$$

where $\mathcal{B}_n$ is the sequence of events defined in (2.5). An estimate of η_{2k-1} could be obtained similarly. These facts lead us to the following estimator of the local Hölder constant L_{t_0} :

$$\widehat{L}^2_{t_0} = \widehat{L}^2_{t_0}(\widehat{H}_{t_0}) = \begin{cases} \dfrac{\widehat{\theta}_{2k-1} - \widehat{\theta}_k}{\widehat{\eta}_{2k-1} - \widehat{\eta}_k} & \text{if } \widehat{\eta}_{2k-1} > \widehat{\eta}_k \text{ and } \widehat{\theta}_{2k-1} > \widehat{\theta}_k, \\ 1 & \text{otherwise.} \end{cases} \tag{2.16}$$

For the implementation we propose $k = \lfloor (\widehat{K}_0 + 7)/8 \rfloor$ with $\widehat{K}_0 = \lfloor \widehat{\mu} \exp\left(-(\log\log\widehat{\mu})^2\right) \rfloor$ and $\widehat{\mu} = {N_0}^{-1} \sum_{n=1}^{N_0} M_n$.

To estimate the variance, we propose

$$\widehat{\sigma}^2 = \widehat{\sigma}^2_{t_0} = \frac{1}{N_0} \sum_{n=1}^{N_0} \frac{1}{2|\mathcal{S}_n|} \sum_{m \in \mathcal{S}_n} \left[Y^{(n)}_{(m)} - Y^{(n)}_{(m-1)} \right]^2, \tag{2.17}$$

where $\mathcal{S}_n \subset \{2, 3, \ldots, M_n\}$ is a set of indices for the $n-$th trajectory and $|\mathcal{S}_n|$ is the cardinal of $\mathcal{S}_n$. When the variance of the error ε is considered constant, one could take $\mathcal{S}_n = \{2, 3, \ldots, M_n\}$. When it depends on t_0, one could take

$$\mathcal{S}_n = \left\{ m : T_{(1)}^{(n)} \leq T_{(m)}^{(n)} \leq T_{(\widehat{K}_0)}^{(n)} \right\},$$

with $\widehat{K}_0$ defined above. This is the choice we used in our empirical investigation. When the variance of the errors also depends on the realizations X_u, as described in Section 2.2.4, in general it is no longer possible to consistently estimate $\sigma^2(X_{t_0}, t_0)$. Our simulation experiments indicate that the estimate (3.30) remains a reasonable choice.

Finally, the constant involved in the definition of the bandwidth could be estimated by $\widehat{C}$ obtained by plugging the estimates of the unknown quantities into the definition of C in (2.15). Concerning the kernel, we use $K(t) = (3/4)\,(1-t^2)\,\mathbf{1}_{[-1,1]}(t)$, that is the Epanechnikov kernel for which $\|K\|^2 = 3/5$ and $\int |K(v)||v|^{\varsigma_{t_0}} dv = 3\{(\varsigma_{t_0}+1)(\varsigma_{t_0}+3)\}^{-1}$.

An implementation of the method is available as a **R** package on Github at the URL adress: https://github.com/StevenGolovkine/denoisr.

2.4.1 Simulation experiments

In this section, we illustrate the behavior of our local regularity estimator $\widehat{\varsigma}_{t_0} = \hat{\mathbf{d}} + \widehat{H}_{t_0}$ computed using the *learning* set of noisy curves, and the performance of kernel smoother it induces for estimating the noisy curves from the *online* set. The procedure for calculating $\widehat{\varsigma}_{t_0}$ is summarized in the following algorithm where $LP(d)$ means local polynomial smoother with degree $d \geq 0$. The Nadaraya-Watson smoother corresponds to $LP(0)$.

For the curve estimation, we use the observations $(Y_1^{[n_1]}, T_1^{[n_1]}), \ldots, (Y_{M_{n_1}}^{[n_1]}, T_{M_{n_1}}^{[n_1]})$, $1 \leq n_1 \leq N_1$, and $LP(\hat{\mathbf{d}})$ with $\hat{\mathbf{d}}$ delivered by Algorithm 2. The bandwidth is calculated as $\widehat{h}_{n_1} = (\widehat{C}/M_{n_1})^{1/(2\widehat{\varsigma}_{t_0}+1)}$, $1 \leq n_1 \leq N_1$, with $\widehat{\varsigma}_{t_0}$ obtained from Algorithm 2. The constant estimate $\widehat{C}$ is the same for all curves in the *online* set, that is that obtained with $\widehat{\varsigma}_{t_0}$, $\widehat{L}_{t_0}(\widehat{H}_{t_0})$ and $\widehat{\sigma}^2_{t_0}$. We compare our approach with the classical *Cross-Validation (CV)* (least-squares leave-one-out) method applied for each curve $X^{[n_1]}$ separately. For CV, we use the **R** package `np` [69], after rescaling the CV bandwidth to account for their different definition of the Epanechnikov kernel. At this stage, we want to point out that our smoothing method is much faster than any standard, trajectory-by-trajectory approach, such as CV. We report a time comparison in the Apendix 2.E, and as expected, the ratio between the times needed for CV and for our approach is at least of the same order as

Algorithm 2: Estimation of the local regularity $\varsigma_{t_0} = \mathbf{d} + H_{t_0}$

Result: Estimation of ς_{t_0} from the learning set of N_0 noisy curves
Calculate $\widehat{\mu} = {N_0}^{-1} \sum_{n=1}^{N_0} M_n$ and $\widehat{K}_0 = \lfloor \widehat{\mu} \exp(-(\log\log(\widehat{\mu}))^2) \rfloor$;
Calculate $\widehat{H}_{t_0}$ and set $\hat{\mathbf{d}} = 0$;
while $\widehat{H}_{t_0} > 1 - \log^{-2}(\widehat{\mu})$ **do**
  Calculate $\widehat{L}_{t_0}(1)$, as in (2.16), and $\widehat{\sigma}^2_{t_0}$;
  Calculate $\widehat{C}$ with $\widehat{\varsigma}_{t_0} = \hat{\mathbf{d}} + \widehat{H}_{t_0}$, $\widehat{L}_{t_0}(1)$ and $\widehat{\sigma}^2_{t_0}$;
  Calculate the bandwidth $\widehat{h}_n = (\widehat{C}/M_n)^{1/(2\widehat{\varsigma}_{t_0}+1)}$, $1 \leq n \leq N_0$;
  Estimate the $(\hat{\mathbf{d}}+1)-$th derivative of the trajectories of X with $LP(\hat{\mathbf{d}}+1)$;
  Calculate $\widehat{H}_{t_0}$ using the estimated trajectories of the $(\hat{\mathbf{d}}+1)-$th derivative;
  Set $\hat{\mathbf{d}} = \hat{\mathbf{d}} + 1$;
end

N_1. It is worth noting that one cannot follow an *ad-hoc* approach and transfer one CV bandwidth from a curve $X^{[n_1]}$ to another because H_{t_0} is not known, and could even vary with t_0.

The data are generated from the model (3.1) using different settings for X, the distribution of T and the variance of the noise, as well as for N_0 and N_1. For X, we consider three types of Gaussian processes: fractional Brownian motion (fBm) with constant Hurst parameter $H \in (0,1)$, fBm with piecewise constant Hurst parameter, and integrated fBm. In the later case, $X_t = \int_0^t W_H(s)ds$, where W_H denotes a fBm with constant Hurst parameter H. The local regularity is constant for the first and the third type, and variable for the second. The third type is an example of X with smooth trajectories. We identify the setting for X by $s \in \{1,2,3\}$. A more detailed description of these processes, as well as plots of their trajectories, are provided in the Appendix 2.E. The number M of measuring times of a curve is a Poisson random variable with expectation μ, while for the measuring times T, we considered either a uniform distribution (identified by `unif`), or a deterministic equispaced grid (`equi`) on the range $[0,1]$. For the noise, we considered the Gaussian distribution with both constant and variable variance. The cases are identified by σ^2 which could be a number or a list, respectively. The values of σ^2 are chosen in such a way that the variance ratio signal-to-noise remains almost unchanged. Thus, one simulation setting is defined by the 7-tuple $(s, N_0, N_1, \mu, f, H, \sigma^2)$, with $f \in \{\texttt{unif}, \texttt{equi}\}$ and H, the Hurst parameter, is a list in the case of fBm with piecewise constant local regularity. Below, we present the results for a few settings, complementary results are reported in the Supplement. For each type of experiment, the reported results are obtained

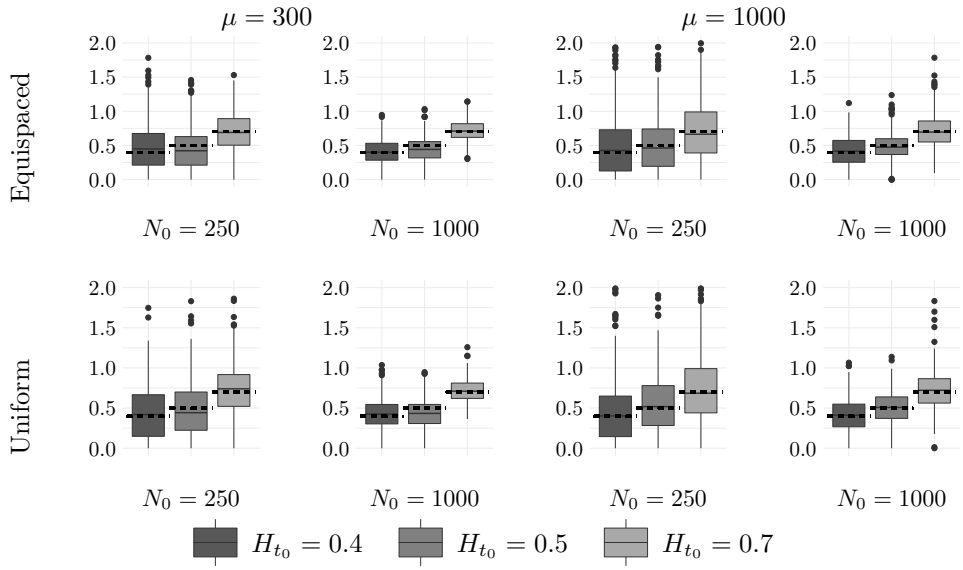


Figure 2.1: Estimation of the local regularity for piecewise fBm, with constant noise variance $\sigma^2 = 0.05$, at $t_0 = 1/6, 1/2$ and $5/6$. True values: $\varsigma_{t_0} = H_{t_0}$ equal to $0.4, 0.5$ and 0.7, respectively.

from 500 replications of the experiment.

Figure 2.1 presents the results for the local regularity estimation for piecewise fBm with homoscedastic noise. The local estimations of H_{t_0} are performed at $t_0 = 1/6, 1/2$ and $5/6$ which correspond to the middle of the interval for each regularity. The true values of H_{t_0} are 0.4, 0.5 and 0.7, respectively. The results show a quite accurate estimator $\widehat{H}_{t_0}$ and confirm the theoretical result on its concentration. Increasing either μ or N_0 improves the concentration. The results for `unif` and `equi` are quite similar. Figure 2.2 presents the estimation of ς_{t_0} for different settings $(3, N_0, 500, \mu, \texttt{equi}, 1.7, 0.005)$. As expected, our local regularity estimation approach also performs well for smooth trajectories.

Next, we present the results on the risk $\mathcal{R}(\widehat{X}; t_0)$. Figure 2.3 presents the boxplots of the risk $\mathcal{R}(\widehat{X}; t_0)$ defined in (2.11) in the case of piecewise constant local regularity, with three values of t_0, each one in the middle of the interval of the changes of regularity are defined. The results are quite good. Part of the curves with lower regularity are harder to estimate and thus results in higher risks than the more regular parts. It appears that N_0 and μ do not have the same influence on the risk as the estimation of the local regularity, and this is in line with the risk bound in Theorem 3. Thus, going from 300 to 1000 sampling points leads to large improvement in terms of risk whereas going from 250 to 1000 curves in the *learning* dataset only results in little or no improvement. Finally, it seems that the method achieves better results for equispaced sampling points.

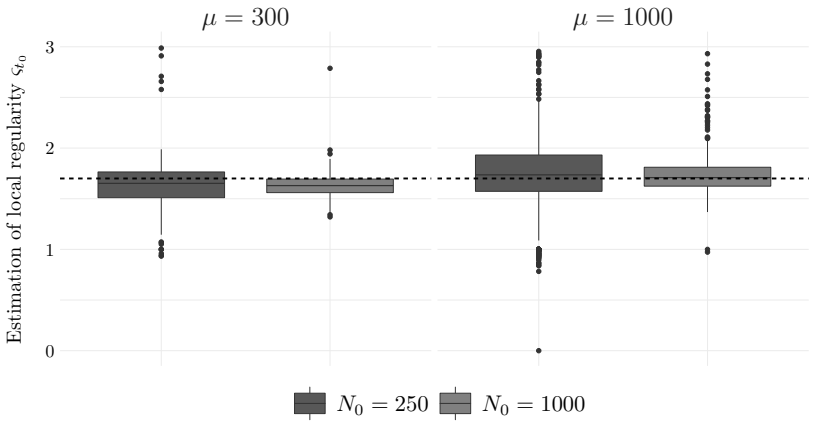


Figure 2.2: Estimation of the local regularity for integrated fBm, with constant noise variance $\sigma^2 = 0.005$, at $t_0 = 0.5$. True value: $\varsigma_{t_0} = 1.7$.

The same conclusions could be drawn from the results presented in Figure 2.4, obtained for the experiment defined by the 7-tuple $(3, 1000, 500, 1000, \texttt{equi}, 1.7, 0.005)$.

Finally, we present a comparison with the CV. Because of the large amount of computing resources required by CV, we only considered a few cases. Figure 2.5 presents the results in terms of the risk calculated at $t_0 = 0.5$ for the simulation setting defined by the tuple $(1, 1000, 500, 300, \texttt{equi}, 0.5, 0.05)$. We make the remark that our method and CV perform similarly despite the fact that CV uses a specifically tailored bandwidth for each curve in the *online* set. The homoscedastic setting is favorable to CV which, for a given curve, uses a global bandwidth at any t_0. Figure 2.6 presents the the heteroscedastic setting $(2, 1000, 500, 1000, \text{equi}, (0.4, 0.5, 0.7), 0.05)$. CV preserves good performances when the local regularity varies moderately. Our method shows close performance in this case, slightly better when $H_{t_0} = 0.7$. Finally, Figure 2.7 presents the results in the setting $(3, 1000, 500, 1000, \texttt{equi}, 1.7, 0.005)$. Again, CV and our method perform quite similarly.

2.4.2 Real data analysis: the NGSIM Study

In this section, our method is applied to data from the NGSIM study, which aims to "describe the interactions of multimodal travelers, vehicles and highway systems", see [62]. This study is known to be one of the largest publicly available source of naturalistic driving data. This dataset is widely used in traffic flow studies from the interpretation of traffic phenomena such as congestion to the validation of models for trajectories prediction (see *e.g.* [43, 76, 145, 95, 71] for some recent references). However, such data have been

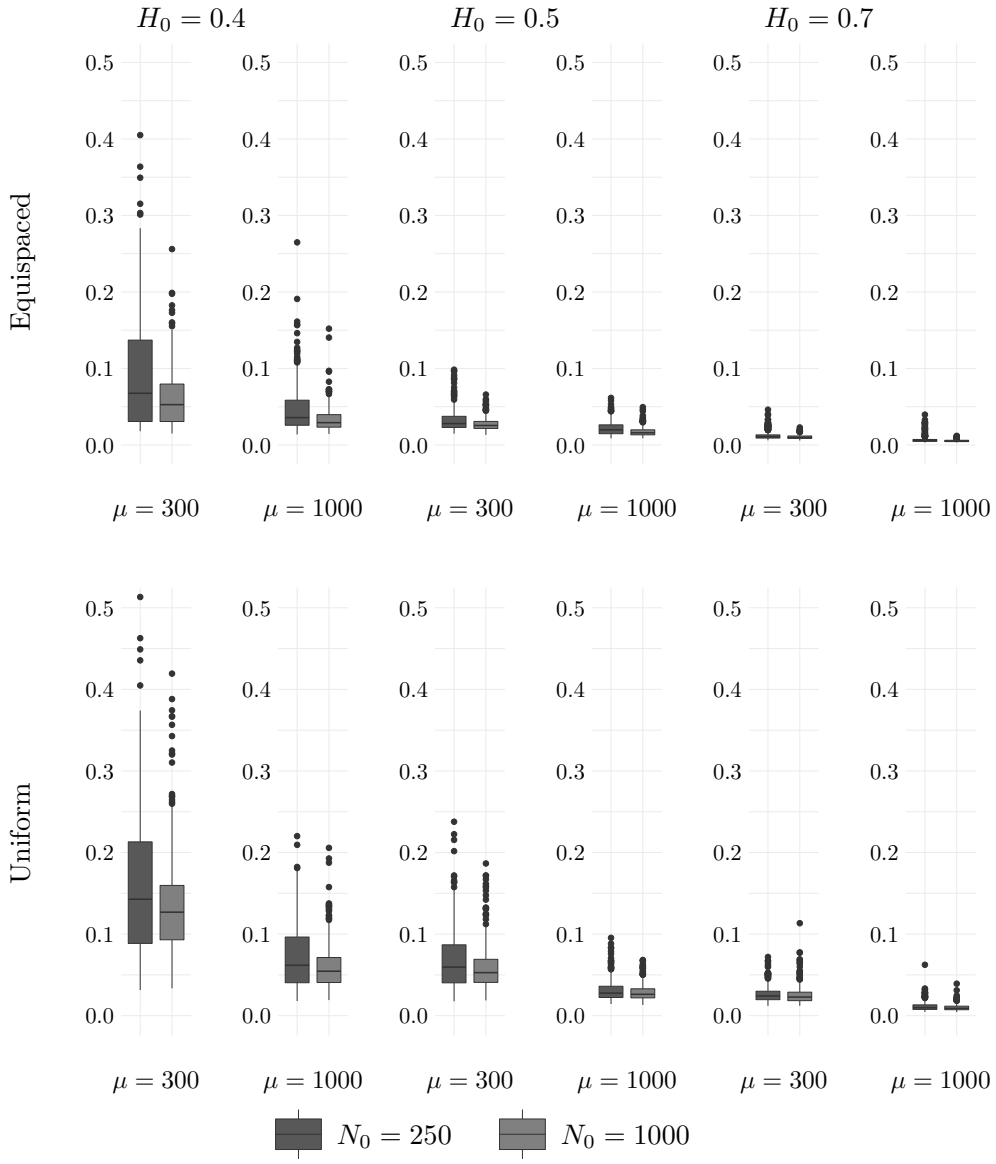


Figure 2.3: Estimation of the risks $\mathcal{R}(\widehat{X}; 1/6)$, $\mathcal{R}(\widehat{X}; 0.5)$ and $\mathcal{R}(\widehat{X}; 5/6)$ for piecewise fBm, with constant noise variance $\sigma^2 = 0.05$.

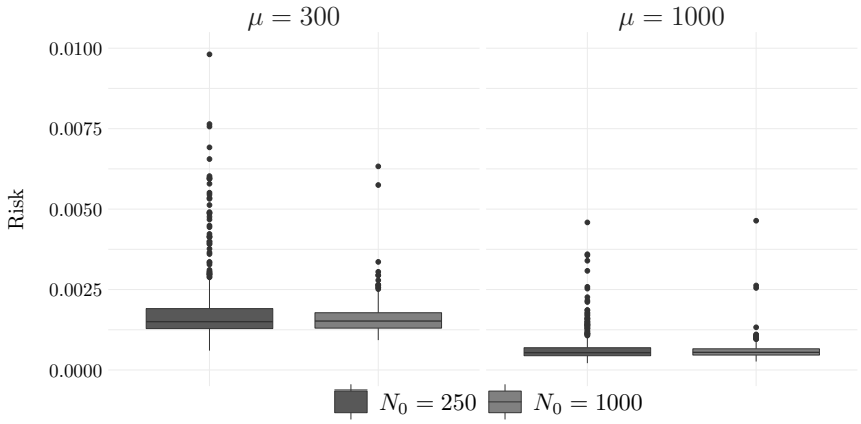


Figure 2.4: Estimation of the risk $\mathcal{R}(\widehat{X}; 0.5)$ for smoothing the noisy trajectories of an integrated fBm, with constant noise variance $\sigma^2 = 0.005$.

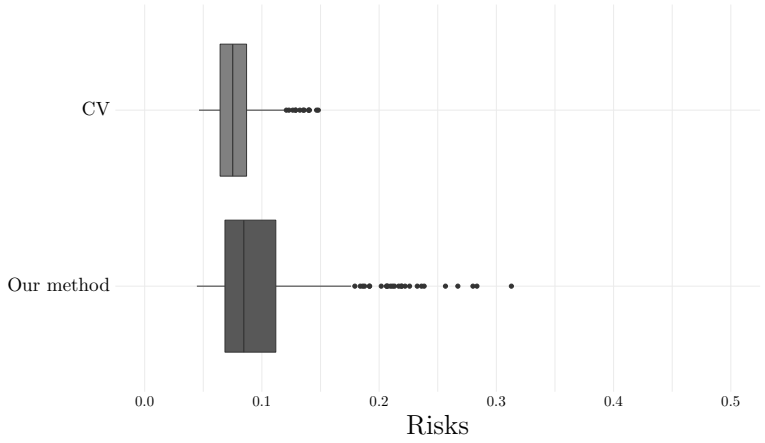


Figure 2.5: CV versus our method: comparing the pointwise risk $\mathcal{R}(\widehat{X}; 0.5)$ for smoothing the noisy trajectories of a fBm; simulation $(1, 1000, 500, 300, \texttt{equi}, 0.5, 0.05)$.

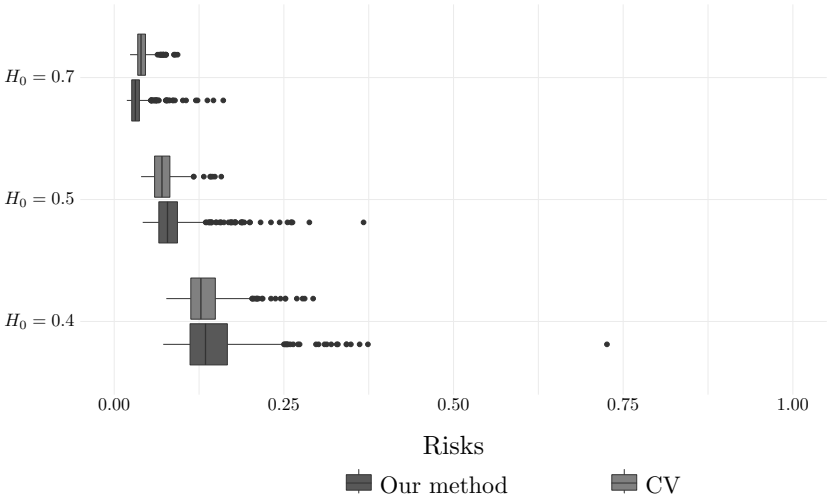


Figure 2.6: CV versus our method: comparing $\mathcal{R}(\widehat{X}; 1/6)$, $\mathcal{R}(\widehat{X}; 0.5)$ and $\mathcal{R}(\widehat{X}; 5/6)$ for smoothing the noisy trajectories of a piecewise fBm; simulation $(2, 1000, 500, 1000, \text{equi}, (0.4, 0.5, 0.7), 0.05)$.

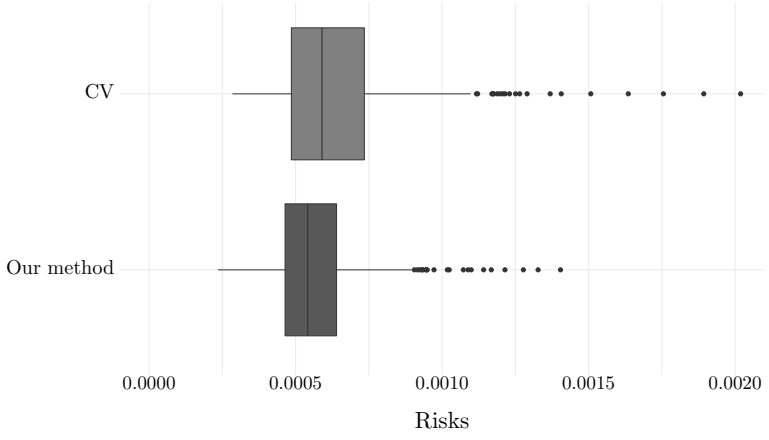


Figure 2.7: CV versus our method: comparing the pointwise risk $\mathcal{R}(\widehat{X}; 0.5)$ for smoothing the noisy trajectories of an integrated fBm; simulation $(3, 1000, 500, 1000, \text{equi}, 1.7, 0.005)$.

proved to be subject to measurement errors revealed by physical inconsistency between the space traveled, velocity and acceleration of the vehicles, *cf.* [105]. Montanino and Punzo [99] developed a trajectory-by-trajectory four-steps method to recover the signals from the noisy curves, and their methodology is now considered as a benchmark in the traffic flow engineering community for analyzing NGSIM data. The steps, finely tuned for the NGSIM data, are : 1. removing the outliers; 2. cutting off the high- and medium-frequency responses in the speed profile; 3. removing the residual nonphysical acceleration values, preserving the consistency requirements; 4. cutting off the high- and medium-frequency responses generated from step 3. The detailed description of these steps is provided in the Appendix 2.F.

To compare our smoothed curves to those of [99], we consider the following ratio:

$$r(\widehat{X},\widetilde{X}) = \frac{\sum_{m=1}^{M_n}\left[Y_m^{(n)} - \widehat{X}(T_m^{(n)})\right]^2}{\sum_{m=1}^{M_n}\left[Y_m^{(n)} - \widetilde{X}(T_m^{(n)})\right]^2}, \quad 1 \leq n \leq 1714,$$

where $\widehat{X}$ denotes our curve estimation while $\widetilde{X}$ is that obtained by [99]. A value of the ratio $r(\widehat{X},\widetilde{X})$ less than 1 indicates smoothed values closer to the observations.

For our illustration, we consider a subset of the NGSIM dataset, known as the I-80 dataset. It contains 45 minutes of trajectories for vehicles on the Interstate 80 Freeway in Emeryville, California, segmented into three 15-minute periods (from 4:00 p.m. to 4:15 p.m.; from 5:00 p.m. to 5:15 p.m. and from 5:15 p.m. to 5:30 p.m.) on April 13, 2005 and corresponds to different traffic conditions (congested, transition between uncongested and congested and fully congested). In total, the dataset contains trajectories, velocities and accelerations for $N_0 = 1714$ individual vehicles that passed through this highway during this period, recorded every 0.1s. The number M_n of measurements for each curve varies from 165 to 946. We focus on the velocity variable and rescale the measurement times for each of the 1714 velocity curves such that the first velocity measurement corresponds to $t = 0$ and the last one to $t = 1$. Figure 2.8 presents a sample of five curves from this data. It can easily be noticed that the velocities are quite erratic and their variation is not physically realistic, indicating the presence of a noise. Moreover, the data have been recorded at a moment of the day when traffic is evolving, it goes from fluid to dense traffic. Therefore, we consider that there are three groups in the data: a first group corresponding to a fluid (high-speed) traffic, a second one for in-between fluid and dense traffic, and a third groups corresponding to the dense (low-speed) traffic. To determine the

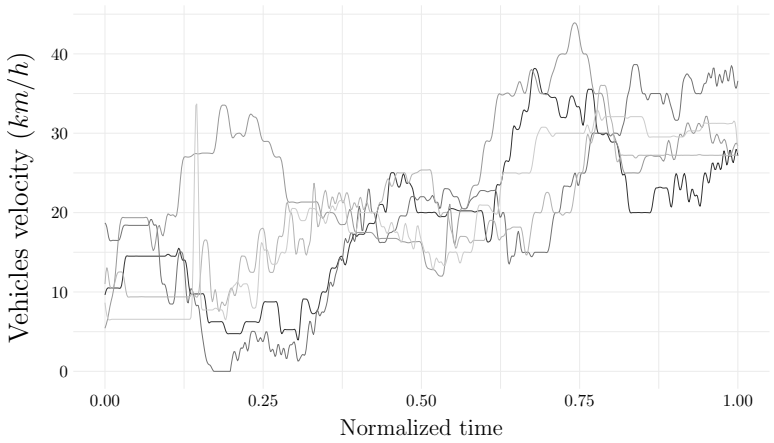


Figure 2.8: I-80 dataset illustration: a sample of five velocity curves.

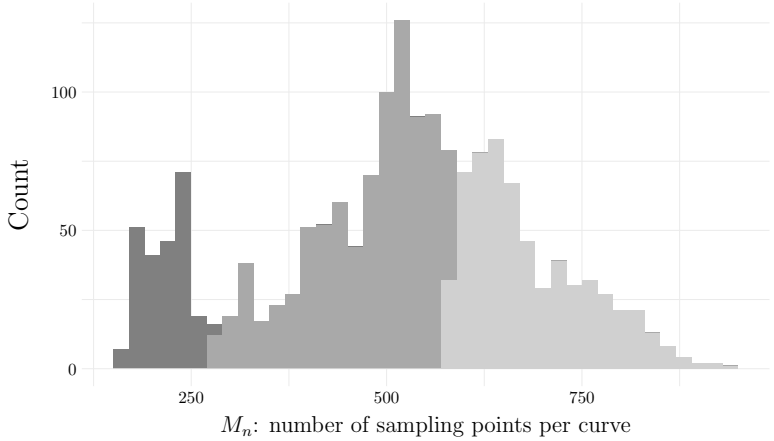
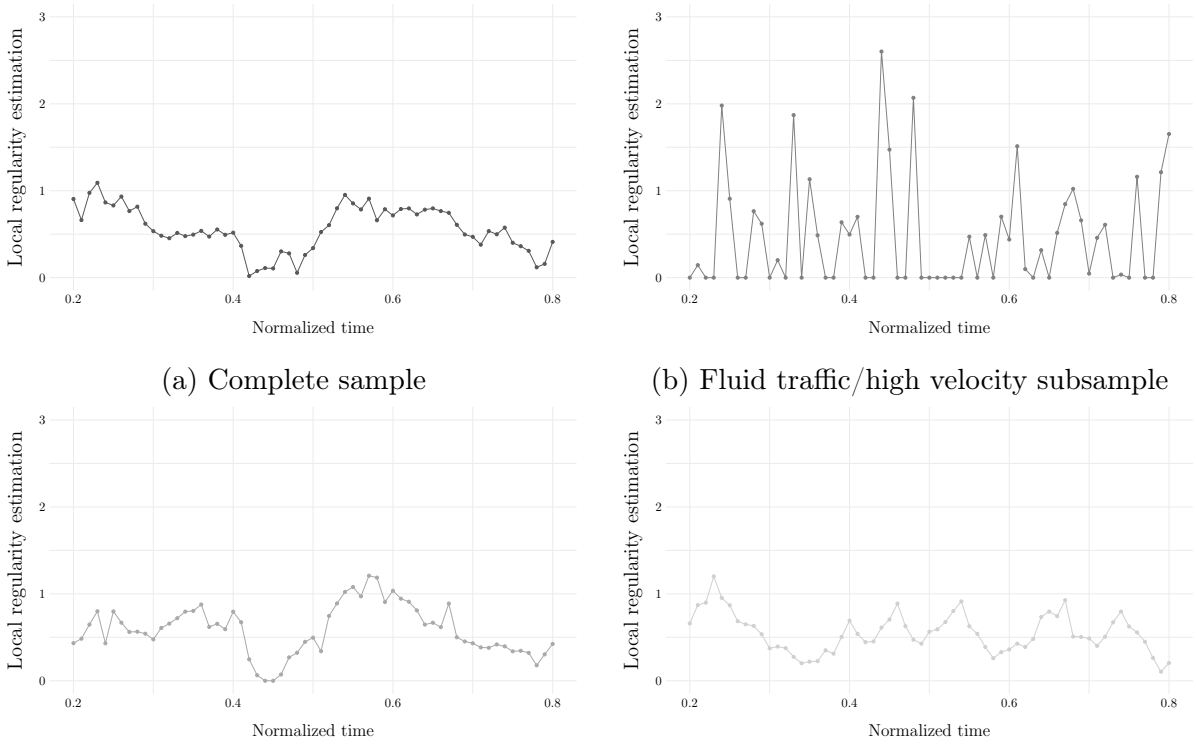


Figure 2.9: I-80 dataset clusters: density of sampling points for fluid (darkest gray), in-between fluid and dense, and dense traffic (lightest gray).

three clusters, we fit a finite Gaussian mixture model to the vector of number of sampling points. The model is estimated by an EM algorithm initialized by hierarchical model-based agglomerative clustering as proposed by Fraley and Raftery [49] and implemented in the **R** package `mclust` [122]. The optimal model is then selected according to BIC. The three resulting classes have 239, 869 and 606 velocity trajectories, respectively. Plots of randomly selected subsamples of trajectories from each groups are provided in the Appendix 2.F. The respective numbers of measures M_n are plotted in Figure 2.9. The mean estimates $\widehat{\mu}$ obtained in the three groups are $218, 474$ and 684, respectively, and the corresponding values $\widehat{K}_0$, as defined as in Corollary 1, are $13, 17$ and 20.

Figure 2.10 presents the results of the estimation of ς_{t_0} for values of t_0 from 0.2 to 0.8,



(a) Complete sample

(b) Fluid traffic/high velocity subsample

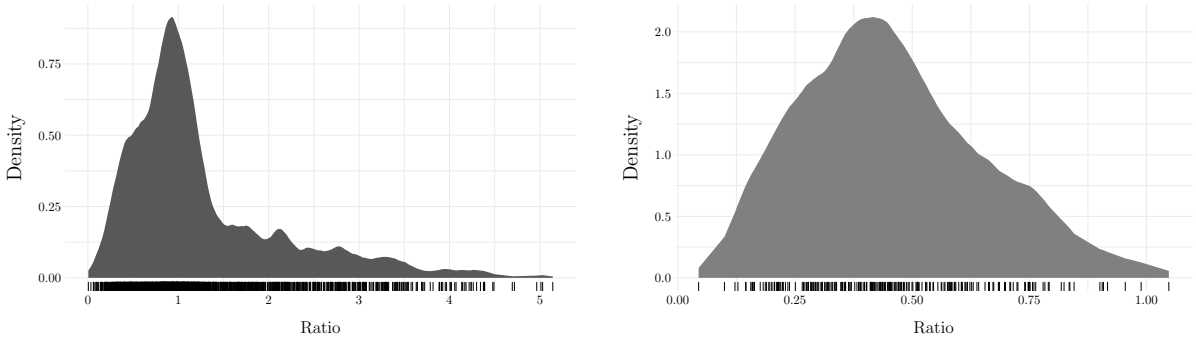
(c) In-between group subsample

(d) Dense traffic/low velocity subsample

Figure 2.10: Estimation of the local regularity of the velocity curves for different t_0.

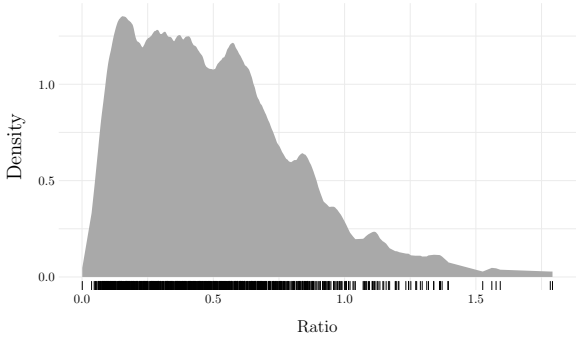
for each group. The evolution of ς_{t_0} is quite smooth, except for Group 1 (Figure 2.10b). A possible explanation could be the small number of curves and the average of M_n in this group, which correspond to low values of N_0 and $\widehat{\mu}$. We also provide the estimation of the regularity using the whole sample of size 1714. The differences we notice between the estimates of ς_{t_0} from different groups support our preliminary clustering step.

To compute the curve estimate we adopt a leave-one-curve-out procedure: each curve is smoothed using the local regularity estimates computed from the other curves in the group (or the other 1713 curves when the data is not split into groups). The densities of the resulting ratios $r(\widehat{X}, \widetilde{X})$ are plotted in Figure 2.11. When the traffic is fluid and the speed is high (group 1), our method perform much better than that of Montanino and Punzo. When the traffic is dense with low speed (group 3), the smoothed values obtained with the two methods are more similar, though our method still exhibits better performance for the majority of the curves.

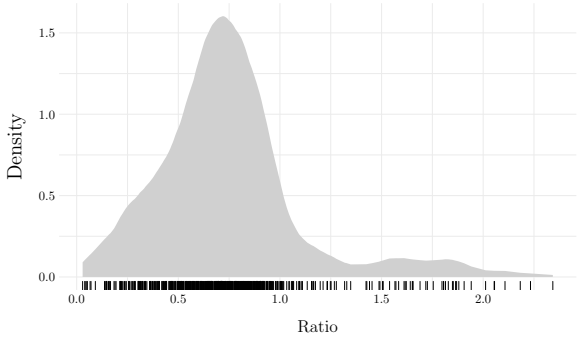


(a) Complete sample

(b) Fluid traffic/high velocity subsample



(c) In-between group subsample



(d) Dense traffic/low velocity subsample

Figure 2.11: Densities of the ratio $r(\widehat{X}, \widetilde{X})$ within different groups.

Appendices

2.A Proof of Theorem 1

The proof of Theorem 1 is based on several lemmas that we present in the following. For these lemmas, we implicitly assume that the conditions of Theorem 1 are satisfied.

Lemma 1. *Let r be an integer such that*

$$(\mu+1)^{\frac{\beta_f(2H_{t_0}+\beta_\phi)}{4+\beta_f(2H_{t_0}+\beta_\phi)}} \leq 8r \leq K_0.$$

Let $\mathfrak{s} \in \{1,2,4,8\}$ and let $1 \leq k, l \leq K_0$ be such that $l-k = \mathfrak{s}r$. Then, for sufficiently large μ, we have

$$\left| \mathbb{E}_{\mathcal{B}}\left[\left| X_{T_{(l)}} - X_{T_{(k)}} \right|^2 \right] - L_{t_0}^2 \left(\frac{l-k}{f(t_0)(\mu+1)} \right)^{2H_{t_0}} \right| \leq \mathfrak{c} \left(\frac{l-k}{f(t_0)(\mu+1)} \right)^{2H_{t_0}+\min(\beta_\phi, \beta_f H_{t_0}/2)},$$

where $\mathfrak{c} = \max(2L_\phi, \mathfrak{c}_1)$ and $\mathfrak{c}_1$ is a constant depending on H_{t_0}, β_ϕ, L_f, β_f and $f(t_0)$.

Proof of Lemma 1. Note that, by the definition of $\mathbb{E}_{\mathcal{B}}$, elementary properties of the conditional expectation, and Assumption (H3),

$$\begin{aligned}
\mathbb{E}_{\mathcal{B}}\left[\left| X_{T_{(l)}} - X_{T_{(k)}} \right|^2 \right] &= \mathbb{E}\left[\left| X_{T_{(l)}} - X_{T_{(k)}} \right|^2 \mathbf{1}_{\mathcal{B}} \right] \\
&= \mathbb{E}\left\{ \mathbb{E}\left[\left| X_{T_{(l)}} - X_{T_{(k)}} \right|^2 \mathbf{1}_{\mathcal{B}} \middle| M, T_1, \dots, T_M \right] \right\} \\
&= \mathbb{E}\left\{ \mathbb{E}\left[\left| X_{T_{(l)}} - X_{T_{(k)}} \right|^2 \middle| M, T_1, \dots, T_M \right] \mathbf{1}_{\mathcal{B}} \right\} \\
&= \mathbb{E}_{\mathcal{B}}\left\{ L_{t_0}^2 |T_{(l)} - T_{(k)}|^{2H_{t_0}} \left[1 + \phi_{t_0}(T_{(k)}, T_{(l)}) \right] \right\} \\
&=: (I) + (II),
\end{aligned}$$

where $(I) = L_{t_0}^2 \mathbb{E}_{\mathcal{B}}\left\{ |T_{(l)} - T_{(k)}|^{2H_{t_0}} \right\}$.

By Lemma 2 applied with $\alpha = 2H_{t_0} \leq 2$,

$$(I) = L_{t_0}^2 \left(\frac{l-k}{f(t_0)(\mu+1)} \right)^{2H_{t_0}} (1+R_1) \quad \text{with} \quad |R_1| \leq \mathfrak{c}_1(2H_{t_0}) \left(\frac{l-k}{f(t_0)(\mu+1)} \right)^{\beta_f H_{t_0}/2}. \tag{2.18}$$

On the other hand, Assumption (H3) implies that

$$|(II)| \leq L_{t_0}^2 L_\phi \mathbb{E}_{\mathcal{B}}\left(|T_{(l)} - T_{(k)}|^{2H_{t_0}+\beta_\phi}\right),$$

and using again Lemma 2 with $\alpha = 2H_{t_0} + \beta_\phi \leq 3$ we obtain

$$|(II)| \leq 2L_{t_0}^2 L_\phi \left(\frac{l-k}{f(t_0)(\mu+1)}\right)^{2H_{t_0}+\beta_\phi}, \tag{2.19}$$

for μ large enough such that $\mathfrak{c}_1(2H_{t_0} + \beta_\phi) \leq 1$. Then, from (2.18) and (2.19) we obtain

$$\mathbb{E}_{\mathcal{B}}\left[\left|X_{T_{(l)}} - X_{T_{(k)}}\right|^2\right] = L_{t_0}^2 \left(\frac{l-k}{f(t_0)(\mu+1)}\right)^{2H_{t_0}} (1+R),$$

where R is a remainder term such that, for sufficiently large μ,

$$\begin{aligned}|R| &\leq \max\left\{2L_\phi \left(\frac{l-k}{f(t_0)(\mu+1)}\right)^{\beta_\phi}, \mathfrak{c}_1(2H_{t_0})\left(\frac{l-k}{f(t_0)(\mu+1)}\right)^{\beta_f H_{t_0}/2}\right\} \\ &\leq \mathfrak{c}\left(\frac{l-k}{f(t_0)(\mu+1)}\right)^{\min(\beta_\phi, \beta_f H_{t_0}/2)},\end{aligned}$$

with $\mathfrak{c} = \max(2L_\phi, \mathfrak{c}_1(2H_{t_0}))$ with $\mathfrak{c}_1(\cdot)$ defined in Lemma 2. □

For the sake of readability, we state below a technical lemma on the moments of the spacings $T_{(l)} - T_{(k)}$, for which the proof is given in the Appendix 2.C. In Lemma 2, we consider that μ is sufficiently large to ensure $(\mu+1)^{\beta_f\alpha/(4+\beta_f\alpha)} + 1 \leq \mu/\{2\log(\mu)\}$.

Lemma 2. *Let $0 < \alpha \leq 3$ be a fixed parameter and let r be an integer such that*

$$(\mu+1)^{\frac{\beta_f\alpha}{4+\beta_f\alpha}} \leq 8r \leq K_0 \quad \textit{with} \quad K_0 \leq \frac{\mu}{2\log(\mu)}.$$

Let $\mathfrak{s} \in \{1, 2, 4, 8\}$ and let $1 \leq k, l \leq K_0$ be such that $l - k = \mathfrak{s}r$. Then, for sufficiently large μ,

$$\left|\mathbb{E}_{\mathcal{B}}\left[\left|T_{(l)} - T_{(k)}\right|^\alpha\right] - \left(\frac{l-k}{f(t_0)(\mu+1)}\right)^\alpha\right| \leq \mathfrak{c}_1\left(\frac{l-k}{f(t_0)(\mu+1)}\right)^{\alpha(1+\beta_f/4)},$$

with $\mathfrak{c}_1 = \mathfrak{c}_1(\alpha) = 8\mathfrak{c}_0\{2f(t_0)\}^{\beta_f\alpha/4}$ and $\mathfrak{c}_0$ a constant depending on α, L_f, β_f and $f(t_0)$.

Lemma 3. *Let k be a positive integer such that $2k-1 \leq K_0$. Then for any $\eta > 0$,*

$$q_k(\eta) := \max\left\{\mathbb{P}(\hat{\theta}_k - \theta_k \geq \eta),\ \ \mathbb{P}(\hat{\theta}_k - \theta_k \leq -\eta)\right\} \leq \exp\left(-\mathfrak{e} N_0 \eta^2\right),$$

where, using the notations introduced in Assumptions (H4) and (H5),

$$\mathfrak{e} = 1/(2\mathfrak{d} + 2\mathfrak{D}) \quad \textit{with} \quad \mathfrak{d} = 3\Big(\mathfrak{a}|J_\mu(t_0)| + 2\mathfrak{b}\Big) \quad \textit{and} \quad \mathfrak{D} = 3\max(\mathfrak{A}|J_\mu(t_0)|, \mathfrak{B}).$$

Proof of Lemma 3 . By the definition in (2.5) and (2.6),

$$\hat{\theta}_k = \frac{1}{N_0}\sum_{n=1}^{N_0} Z_n \qquad \text{where} \qquad Z_n = \left[Y^{(n)}_{(2k-1)} - Y^{(n)}_{(k)}\right]^2 \mathbf{1}_{\mathcal{B}_n},$$

and $\mathcal{B}_n = \left\{M_n \geq K_0, T^{(n)}_{(1)} \in J_\mu(t_0), \ldots, T^{(n)}_{(K_0)} \in J_\mu(t_0)\right\}$. Note that $\mathbb{E}(\hat{\theta}_k) = \theta_k$. Moreover, for any $p \geq 1$, using Assumptions (H4) and (H5), we have

$$\begin{aligned}
\mathbb{E}\Big(|Z_n|^p\Big) &= \mathbb{E}_{\mathcal{B}}\Big(|Y_{(2k-1)} - Y_{(k)}|^{2p}\Big) \\
&\leq 3^{2p-1}\mathbb{E}_{\mathcal{B}}\left(|X_{T_{(2k-1)}} - X_{T_{(k)}}|^{2p} + |\varepsilon_{(2k-1)}|^{2p} + |\varepsilon_{(k)}|^{2p}\right) \\
&\leq 3^{2p-1}\frac{p!}{2}\left(\mathfrak{a}\mathfrak{A}^{p-2}\sup_{u,v\in J_\mu(t_0)}|u-v|^{2pH_{t_0}} + 2\mathfrak{b}\mathfrak{B}^{p-2}\right) \\
&\leq \frac{p!}{2}\mathfrak{d}\mathfrak{D}^{p-2},
\end{aligned}$$

where $\mathfrak{d}$ and $\mathfrak{D}$ are defined in the statement of this lemma. Bernstein's inequality implies

$$\mathbb{P}(\hat{\theta}_k - \theta_k \geq \eta) \leq \exp\left(-\frac{N_0\eta^2}{2\mathfrak{d} + 2\mathfrak{D}\eta}\right) \leq \exp\left(-\mathfrak{e} N_0 \eta^2\right),$$

and the same bound is valid for $\mathbb{P}(\hat{\theta}_k - \theta_k \leq -\eta)$. The bound for $q_k(\eta)$ follows. □

Lemma 4. *Let $2 \leq k < l \leq (K_0+1)/2$ be two positive integers. For any $\eta > 0$, define*

$$p^+_{k,l}(\eta) = \mathbb{P}(\hat{\theta}_l - \hat{\theta}_k \geq (1+\eta)(\theta_l - \theta_k)) \quad \textit{and} \quad p^-_{k,l}(\eta) = \mathbb{P}(\hat{\theta}_l - \hat{\theta}_k \leq (1-\eta)(\theta_l - \theta_k)).$$

Then, for sufficiently large μ,

$$\max\left\{p^+_{k,l}(\eta), p^-_{k,l}(\eta)\right\} \leq 2\exp\left[-\frac{\mathfrak{e}}{16}N_0\eta^2\left(\frac{l-k}{\mu+1}\right)^{4H_{t_0}}\right],$$

with $\mathfrak{e}$ *defined in Lemma 3.*

Proof of Lemma 4. Assume that k and l satisfy the assumptions stated in the Lemma and assume moreover that μ is large enough so that $\eta(\theta_l - \theta_k)/2 < 1$. Then

$$\begin{aligned} p^+_{k,l}(\eta) &= \mathbb{P}\Big[(\hat{\theta}_l - \theta_l) - (\hat{\theta}_k - \theta_k) \geq \eta(\theta_l - \theta_k)\Big] \\ &\leq \mathbb{P}\Big[\hat{\theta}_l - \theta_l \geq \eta(\theta_l - \theta_k)/2\Big] + \mathbb{P}\Big[\hat{\theta}_k - \theta_k \leq -\eta(\theta_l - \theta_k)/2\Big] \\ &\leq q_l\left(\eta(\theta_l - \theta_k)/2\right) + q_k\left(\eta(\theta_l - \theta_k)/2\right), \end{aligned}$$

and the same bound is valid for $p^-_{k,l}(\eta)$. By (2.20) we have $\theta_l - \theta_k \geq \{(l-k)/(\mu+1)\}^{2H_{t_0}}/2$, provided μ is sufficiently large. Then we obtain the bound for $\max\left(p^+_{k,l}(\eta), p^-_{k,l}(\eta)\right)$ after applying Lemma 3. □

Proof of Theorem 1. Let $\epsilon > 0$. With the notation from (2.4) and (2.7), we can write

$$\mathbb{P}\left(\left|\widehat{H}_{t_0}(k) - H_{t_0}\right| > \varepsilon\right) \leq \mathbf{1}_{\left\{\left|H_{t_0}(k) - H_{t_0}\right| > \epsilon/2\right\}} + \mathbb{P}\left(\left|\widehat{H}(k) - H_{t_0}(k)\right| > \varepsilon/2\right) =: B + V,$$

and thus it suffices to bound the terms B and V.

The term B**.** The study of the set in the indicator function boils down to the study of the convergence of $H_{t_0}(k)$ to H_{t_0}. Using Lemma 1 with $l - k = k - 1 = r$ we have

$$\theta_k - 2\sigma^2 = \mathbb{E}_{\mathcal{B}}\left[\left(X_{T_{(2k-1)}} - X_{T_{(k)}}\right)^2\right] = L^2_{t_0}\left(\frac{k-1}{f(t_0)(\mu+1)}\right)^{2H_{t_0}}(1+\rho_k),$$

and

$$|\rho_k| \leq \mathfrak{c}\left(\frac{k-1}{f(t_0)(\mu+1)}\right)^{\min(\beta_\phi, \beta_f H_{t_0}/2)} =: \rho^*_k,$$

with $\mathfrak{c}$ a constant defined in Lemma 1. Using again Lemma 1 with $k = 2k-1$, $l = 4k-3$ and $\mathfrak{a} = 2$ and taking the difference, we deduce that there exists R_k such that

$$\begin{aligned} \theta_{2k-1} - \theta_k &= L^2_{t_0}\left(\frac{2(k-1)}{f(t_0)(\mu+1)}\right)^{2H_{t_0}}(1+\rho_{2k-1}) - L^2_{t_0}\left(\frac{k-1}{f(t_0)(\mu+1)}\right)^{2H_{t_0}}(1+\rho_k) \\ &= (4^{H_{t_0}} - 1)L^2_{t_0}\left(\frac{k-1}{f(t_0)(\mu+1)}\right)^{2H_{t_0}}(1+R_k), \end{aligned} \tag{2.20}$$

where

$$|R_k| = \left|\frac{4^{H_{t_0}}\rho_{2k-1} - \rho_k}{4^{H_{t_0}} - 1}\right| =\leq \frac{4^{H_{t_0}} + 1}{4^{H_{t_0}} - 1}\rho^*_{2k-1} \leq \frac{4^{H_{t_0}} + 1}{4^{H_{t_0}} - 1}\rho^*_{K_0}.$$

Similarly, we obtain:

$$\theta_{4k-3} - \theta_{2k-1} = (4^{H_{t_0}} - 1)L_{t_0}^2 \left(\frac{2(k-1)}{f(t_0)(\mu+1)} \right)^{2H_{t_0}} (1 + R_{2k-1}) \,. \tag{2.21}$$

Combining (2.20) and (2.21), we obtain

$$\log(\theta_{4k-3} - \theta_{2k-1}) - \log(\theta_{2k-1} - \theta_k) = H_{t_0} \log 4 + \log(1 + R_{2k-1}) - \log(1 + R_k),$$

which leads, using the definition of $H_{t_0}(k)$ given by (2.4), to:

$$H_{t_0}(k) = H_{t_0} + \eta_k \qquad \text{where} \qquad \eta_k = \frac{\log(1 + R_{2k-1}) - \log(1 + R_k)}{2 \log 2}.$$

Note that, for sufficiently large μ, both R_k and R_{2k-1} are greater that $-1/2$. This implies

$$|\eta_k| \leq \frac{|R_{2k-1} - R_k|}{\log 2} \leq \left(\frac{2}{\log 2} \frac{4^{H_{t_0}} + 1}{4^{H_{t_0}} - 1} \right) \rho^*_{4k-3}.$$

Thus, since $\rho^*_{4k-3} \leq \rho^*_{K_0}$, the condition $\left| H_{t_0}(k) - H_{t_0} \right| > \varepsilon/2$ fails and $B = 0$ as soon as

$$\epsilon > \left(\frac{4}{\log 2} \frac{4^{H_{t_0}} + 1}{4^{H_{t_0}} - 1} \right) \rho^*_{K_0},$$

that is as soon as condition (2.9) is satisfied, provided μ is sufficiently large.

The term V. Defining the event $\mathcal{D} = \{\hat{\theta}_{4k-3} > \hat{\theta}_{2k-1} > \hat{\theta}_k\}$, we can write

$$\mathbb{P}\left(\left| \widehat{H}(k) - H_{t_0}(k) \right| > \epsilon/2 \right) \leq \mathbb{P}\left(\left| \widehat{H}(k) - H_{t_0}(k) \right| > \epsilon/2, \mathcal{D} \right) + \mathbb{P}(\overline{\mathcal{D}}). \tag{2.22}$$

First note that using Lemma 4 we have, for sufficiently large μ :

$$\begin{aligned} \mathbb{P}(\overline{\mathcal{D}}) &\leq \mathbb{P}(\hat{\theta}_k \geq \hat{\theta}_{2k-1}) + \mathbb{P}(\hat{\theta}_{2k-1} \geq \hat{\theta}_{4k-3}) \\ &\leq p^-_{2k-1,k}(1) + p^-_{4k-3,2k-1}(1) \leq 4 \exp\left[-\frac{\mathfrak{e}}{16} N_0 \left(\frac{k-1}{\mu+1} \right)^{4H_{t_0}} \right]. \end{aligned} \tag{2.23}$$

Now, it remains to bound the quantity

$$\wp = \mathbb{P}\left(\left| \widehat{H}(k) - H_{t_0}(k) \right| > \epsilon/2, \mathcal{D} \right)$$

$$= \mathbb{P}\left[\left|\log\left(\frac{\hat{\theta}_{4k-3} - \hat{\theta}_{2k-1}}{\theta_{4k-3} - \theta_{2k-1}} \times \frac{\theta_{2k-1} - \theta_k}{\hat{\theta}_{2k-1} - \hat{\theta}_k}\right)\right| > \epsilon \log 2, \mathcal{D}\right].$$

Since both $\hat{\theta}_{4k-3} - \hat{\theta}_{2k-1}$ and $\hat{\theta}_{2k-1} - \hat{\theta}_k$ are positive under $\mathcal{D}$, we have

$$\begin{aligned}
\wp &\leq \mathbb{P}\left[\frac{\hat{\theta}_{4k-3} - \hat{\theta}_{2k-1}}{\theta_{4k-3} - \theta_{2k-1}} \times \frac{\theta_{2k-1} - \theta_k}{\hat{\theta}_{2k-1} - \hat{\theta}_k} > 2^{\epsilon}, \mathcal{D}\right] \\
&\quad + \mathbb{P}\left[\frac{\hat{\theta}_{4k-3} - \hat{\theta}_{2k-1}}{\theta_{4k-3} - \theta_{2k-1}} \times \frac{\theta_{2k-1} - \theta_k}{\hat{\theta}_{2k-1} - \hat{\theta}_k} < 2^{-\epsilon}, \mathcal{D}\right] \\
&\leq \mathbb{P}\left[\frac{\hat{\theta}_{4k-3} - \hat{\theta}_{2k-1}}{\theta_{4k-3} - \theta_{2k-1}} > 2^{\frac{\epsilon}{2}}\right] + \mathbb{P}\left[\frac{\hat{\theta}_{2k-1} - \hat{\theta}_k}{\theta_{2k-1} - \theta_k} < 2^{-\frac{\epsilon}{2}}\right] \\
&\quad + \mathbb{P}\left[\frac{\hat{\theta}_{4k-3} - \hat{\theta}_{2k-1}}{\theta_{4k-3} - \theta_{2k-1}} < 2^{-\frac{\epsilon}{2}}\right] + \mathbb{P}\left[\frac{\hat{\theta}_{2k-1} - \hat{\theta}_k}{\theta_{2k-1} - \theta_k} > 2^{\frac{\epsilon}{2}}\right].
\end{aligned}$$

Applying Lemma 4, we obtain:

$$\wp \leq p^{+}_{4k-3,2k-1}(2^{\frac{\epsilon}{2}} - 1) + p^{-}_{4k-3,2k-1}(1 - 2^{-\frac{\epsilon}{2}}) + p^{+}_{2k-1,k}(2^{\frac{\epsilon}{2}} - 1) + p^{-}_{2k-1,k}(1 - 2^{-\frac{\epsilon}{2}}).$$

Now remark that

$$\begin{aligned}
p^{+}_{2k-1,k}(2^{\frac{\epsilon}{2}} - 1) &\leq 2\exp\left[-\frac{\mathfrak{e}}{16} N_0 \left(2^{\frac{\epsilon}{2}} - 1\right)^2 \left(\frac{k-1}{\mu+1}\right)^{4H_{t_0}}\right] \\
&\leq 2\exp\left[-\frac{\mathfrak{e}\log^2(2)}{64} N_0 \epsilon^2 \left(\frac{k-1}{\mu+1}\right)^{4H_{t_0}}\right],
\end{aligned}$$

and, as soon as $\epsilon < 2/\log 2$, we have $1 - 2^{-\epsilon/2} \leq \epsilon/4$, which implies:

$$\begin{aligned}
p^{-}_{2k-1,k}(1 - 2^{-\frac{\epsilon}{2}}) &\leq 2\exp\left[-\frac{\mathfrak{e}}{16} N_0 \left(1 - 2^{-\frac{\epsilon}{2}}\right)^2 \left(\frac{k-1}{\mu+1}\right)^{4H_{t_0}}\right] \\
&\leq 2\exp\left[-\frac{\mathfrak{e}\log^2(2)}{256} N_0 \epsilon^2 \left(\frac{k-1}{\mu+1}\right)^{4H_{t_0}}\right].
\end{aligned}$$

Using similar derivations for the others terms, we obtain:

$$\wp \leq 8\exp\left[-\frac{\mathfrak{e}\log^2(2)}{256} N_0 \epsilon^2 \left(\frac{k-1}{\mu+1}\right)^{4H_{t_0}}\right]. \tag{2.24}$$

Combining (2.22) with (2.23) and (2.24), we obtain, for sufficiently large μ and $\epsilon < 2/\log 2$:

$$\mathbb{P}\left(\left|\widehat{H}(k) - H_{t_0}(k)\right| > \epsilon/2\right) \leq 12 \exp\left[-\mathfrak{f} N_0 \epsilon^2 \left(\frac{k-1}{\mu+1}\right)^{4H_{t_0}}\right],$$

where $\mathfrak{f} = \mathfrak{e}\log^2(2)/256$. □

2.B Proofs of Theorems 2 and 3

The proofs of Theorems 2 and 3 are based on the following lemmas for which the proofs are provided in the Appendix 2.C. For the first lemma we consider the matrix A defined in (5.2) with the bandwidth $\widehat{h} = M^{-1/(2\widehat{\varsigma}_{t_0}+1)}$. Let λ be the smallest eigenvalue of this matrix. Let

$$\mathbf{A} = f(t_0) \int_{\mathbb{R}} U(u)U^\top(u)K(u)du,$$

and let λ_0 denote its smallest eigenvalue. In the following, we assume that $K(\cdot)$ satisfies (2.14). Then $\mathbf{A}$ is positive definite [see 135, for details] and thus $\lambda_0 > 0$.

Lemma 5. *Under Assumptions (LP2)–(LP4), the matrix A is positive semidefinite. Moreover, there exists a positive constant $\mathfrak{g}$ that depends only on the kernel K, $\mathbf{d}$, $f(t_0)$, β_f, L_f and λ_0, such that, for sufficiently large μ,*

$$\sup_{0<\beta\leq\lambda_0/2} \mathbb{P}(\lambda \leq \beta) \leq \mathfrak{K}_2 \exp\left[-\frac{\mathfrak{g}}{2}\tau(\mu)\log^2(\mu)\right] \quad \textit{where} \quad \tau(\mu) = \frac{1}{\log^2(\mu)}\left(\frac{\mu}{\log(\mu)}\right)^{\frac{2\varsigma_{t_0}}{2\varsigma_{t_0}+1}},$$

$\varsigma_{t_0} = \mathbf{d} + H_{t_0}$, and $\mathfrak{K}_2$ is a universal constant.

Since the dimension of A and $\mathbf{A}$ are given by $\hat{\mathbf{d}}$, the probability $\mathbb{P}(\cdot)$ in Lemma 5 should be understood as the conditional probability given the estimator $\hat{\mathbf{d}}$.

Lemma 6. *Let ξ be a positive random variable such that $c_1 \coloneqq \mathbb{E}\left[\exp\left(\eta_0\xi^4\right)\right] < \infty$, for some positive constant η_0. Then, for any $\tau \geq 1$:*

$$\mathbb{E}\left[\exp\left(\tau\xi\right)\right] \leq c_1 \exp\left(c_2\tau^{4/3}\right) \qquad \textit{where} \qquad c_2 = \left(5/16\eta_0\right)^{1/3}.$$

Proof of Theorem 2. Without loss of generality, we could suppose that $f(t_0)/2 \leq f(t) \leq$

$2f(t_0)$, $\forall t \in J_\mu(t_0)$. We define the events

$$\mathcal{E} = \{\lambda > \lambda_0/2\}, \quad \mathcal{F} = \left\{|\widehat{H}_{t_0} - H_{t_0}| \leq \log^{-2}(\mu)\right\} \cap \left\{\hat{\mathbf{d}} = \mathbf{d}\right\},$$

and $\mathcal{G} = \{|X_{t_0}| \leq \tau^{\tilde{\alpha}}(\mu)\}$, with $1/3 < \tilde{\alpha} < \alpha = 5/12$. Next, let $Z = \left|\widehat{X}_{t_0} - X_{t_0}\right|$. Assume that μ is such that $\log^{-1}(\mu/\log(\mu)) < \lambda_0/2$, then using Assumption (H2), we have:

$$\mathbb{E}[\varphi(\tau(\mu)Z^2)] = (A) + (B) + (C) + (D),$$

where $\varphi(x) = \exp(x^{1/4})$ and

$$\begin{aligned}
(A) &= \mathbb{E}\left[\varphi\left(\tau(\mu)Z^2\right)\mathbf{1}_{\mathcal{E}}\mathbf{1}_{\mathcal{F}}\mathbf{1}_{\mathcal{G}}\right] \\
(B) &= \mathbb{E}\left[\varphi\left(\tau(\mu)Z^2\right)\mathbf{1}_{\overline{\mathcal{E}}}\right] \leq \mathbb{E}^{1/2}\left[\varphi^2\left(\tau(\mu)Z^2\right)\right]\mathbb{P}^{1/2}(\overline{\mathcal{E}}) \\
(C) &= \mathbb{E}\left[\varphi\left(\tau(\mu)Z^2\right)\mathbf{1}_{\overline{\mathcal{F}}}\right] \leq \mathbb{E}^{1/2}\left[\varphi^2\left(\tau(\mu)Z^2\right)\right]\mathbb{P}^{1/2}(\overline{\mathcal{F}}) \\
(D) &= \mathbb{E}\left[\varphi\left(\tau(\mu)Z^2\right)\mathbf{1}_{\overline{\mathcal{G}}}\right] \leq \mathbb{E}^{1/2}\left[\varphi^2\left(\tau(\mu)Z^2\right)\right]\mathbb{P}^{1/2}(\overline{\mathcal{G}}).
\end{aligned}$$

We show that (A) is the main term, and it is bounded by a constant.

By construction, $\left|\hat{X}_{t_0}\right| \leq \tau^{\alpha}(M)$ and, by (LP2),

$$\tau^{\alpha}(M) \geq \tau^{\alpha}(\mu/\log(\mu)) \geq \tau^{\tilde{\alpha}}(\mu),$$

provided that μ is sufficiently large. Thus, for sufficiently large μ,

$$\left|\hat{X}_{t_0} - X_{t_0}\right|\mathbf{1}_{\mathcal{G}} \leq \left|U^{\top}(0)\hat{\vartheta} - X_{t_0}\right|\mathbf{1}_{\mathcal{G}} \leq \left|U^{\top}(0)\hat{\vartheta} - X_{t_0}\right|,$$

with $\hat{\vartheta} = A^{-1}a$ and A and a defined in (5.2) and (5.3), respectively. Therefore,

$$\sqrt{\tau(\mu)}Z\mathbf{1}_{\mathcal{G}} \leq \sqrt{\tau(\mu)}\left|\sum_{m=1}^{M}(X_{T_m} - X_{t_0})W_m\right| + \sqrt{\tau(\mu)}\left|\sum_{m=1}^{M}\varepsilon_m W_m\right|,$$

where

$$W_m = \frac{1}{Mh}U^{\top}(0)A^{-1}U\left(\frac{T_m - t_0}{h}\right)K\left(\frac{T_m - t_0}{h}\right).$$

This leads to

$$\left(\sqrt{\tau(\mu)}Z\right)^{1/2}\mathbf{1}_{\mathcal{G}} \leq \left(\sqrt{\tau(\mu)}\left|\sum_{m=1}^{M}(X_{T_m} - X_{t_0})W_m\right|\right)^{1/2} + \left(\sqrt{\tau(\mu)}\left|\sum_{m=1}^{M}\varepsilon_m W_m\right|\right)^{1/2}.$$

Then, to show that (A) is finite, it suffices to show that

$$A_1 = \mathbb{E}\left[\exp\left\{\left(4\sqrt{\tau(\mu)}\left|\sum_{m=1}^{M}(X_{T_m} - X_{t_0})W_m\right|\right)^{1/2}\right\}\mathbf{1}_{\mathcal{E}}\mathbf{1}_{\mathcal{F}}\right],$$

and

$$A_2 = \mathbb{E}\left[\exp\left\{\left(4\sqrt{\tau(\mu)}\left|\sum_{m=1}^{M}\varepsilon_m W_m\right|\right)^{1/2}\right\}\mathbf{1}_{\mathcal{E}}\mathbf{1}_{\mathcal{F}}\right],$$

are finite and to apply Cauchy-Schwarz inequality. To control the stochastic term A_2, remark that:

$$A_2 = 1 + \sum_{p\geq 1}\frac{2^p[\tau(\mu)]^{p/4}}{p!}B_p, \tag{2.25}$$

where

$$B_p = \mathbb{E}\left[\left|\sum_{m=1}^{M}\varepsilon_m W_m\right|^{p/2}\mathbf{1}_{\mathcal{E}}\mathbf{1}_{\mathcal{F}}\right].$$

By Jensen's inequality, $B_1 \leq B_2^{1/2} \leq B_3^{1/3} \leq B_4^{1/4}$. Thus, it remains to control B_p for any $p \geq 4$. For such values of p, we use Marcinkiewicz-Zygmund's inequality and obtain:

$$B_p \leq \left(\frac{p}{2} - 1\right)^{p/2}\mathbb{E}\left[\left(\sum_{m=1}^{M}\varepsilon_m^2 W_m^2\right)^{p/4}\mathbf{1}_{\mathcal{E}}\mathbf{1}_{\mathcal{F}}\right].$$

By a version of Lemma 1.3 of [135], for κ defined in (2.14),

$$\sup_{1\leq m\leq M}|W_m|\mathbf{1}_{\mathcal{E}_{\lambda_0/2}} \leq \frac{4\kappa}{\lambda_0 M h},$$

and

$$\sum_{m=1}^{M}|W_m|\mathbf{1}_{\mathcal{E}_{\lambda_0/2}} \leq \frac{4\kappa}{\lambda_0}\frac{1}{Mh}\sum_{j=1}^{M}\mathbf{1}_{\{t_0-h\leq T_m\leq t_0+h\}}. \tag{2.26}$$

Let $\chi_m = h^{-1}\mathbf{1}_{\{t_0-h\leq T_m\leq t_0+h\}}$. By the Rosenthal inequality, there exists a universal constant C, such that, for any $q \geq 1$,

$$\tilde{\mathbb{E}}\left[\left(\sum_{m=1}^{M}\chi_m\right)^q\right] \leq \frac{C^q q^q}{\log^q q}\max\left\{\sum_{m=1}^{M}\tilde{\mathbb{E}}\chi_m^q, \left(\sum_{m=1}^{M}\tilde{\mathbb{E}}\chi_m\right)^q\right\} \leq \left(\frac{4qCf(t_0)M}{\log q}\right)^q. \tag{2.27}$$

See [85]. Since $W_m^2 \leq |W_m| \sup_{1\leq j\leq M} |W_j|$, we deduce

$$\tilde{\mathbb{E}}\left[\left(\sum_{m=1}^{M} W_m^2\right)^q \mathbf{1}_{\mathcal{E}}\right] \leq \left(\frac{4\kappa}{\lambda_0 Mh}\right)^q \left(\frac{16q\kappa C f(T)}{\lambda_0 \log q}\right)^q =: \left(\frac{1}{Mh}\right)^q \left(\frac{c_0 q}{\log q}\right)^q.$$

Using Assumption (LP2), we deduce:

$$\mathbb{E}\left[\left(\sum_{m=1}^{M} W_m^2\right)^q \mathbf{1}_{\mathcal{E}}\mathbf{1}_{\mathcal{F}}\right] \leq \left(\frac{c_0 q}{\log q}\right)^q \mathbb{E}\left[\left(\frac{\log \mu}{\mu}\right)^{q\frac{2\widehat{H}_{t_0}}{2\widehat{H}_{t_0}+1}} \mathbf{1}_{\mathcal{F}}\right] = 2\left(\frac{c_0 q}{\log q}\right)^q \frac{1}{\left\{\tau(\mu)\log^2 \mu\right\}^q}.$$

The last line can be deduced using similar arguments to those used to obtain (2.42) in the Appendix 2.C. Next, let $\widetilde{W}_m = W_m^2 / \sum_{j=1}^{M} W_j^2$. Since the error terms are independent on the T_m's, and using (H5), by Jensen's inequality

$$\begin{aligned}
B_p \left(\frac{p}{2}-1\right)^{-p/2} &\leq \mathbb{E}\left[\left(\sum_{m=1}^{M} |\varepsilon_m|^{p/2}\widetilde{W}_m\right)\left(\sum_{j=1}^{M} W_j^2\right)^{p/4} \mathbf{1}_{\mathcal{E}}\mathbf{1}_{\mathcal{F}}\right] \\
&= \mathbb{E}\left\{\mathbb{E}\left[\left(\sum_{m=1}^{M} |\varepsilon_m|^{p/2}\widetilde{W}_m\right) \mid M, W_1, \ldots, W_M\right]\left(\sum_{j=1}^{M} W_j^2\right)^{p/4} \mathbf{1}_{\mathcal{E}}\mathbf{1}_{\mathcal{F}}\right\} \\
&\leq \left(\mathbb{E}|\varepsilon|^{2p}\right)^{1/4} \mathbb{E}\left[\left(\sum_{m=1}^{M} W_m^2\right)^{p/4} \mathbf{1}_{\mathcal{E}}\mathbf{1}_{\mathcal{F}}\right] \\
&\leq \left(\frac{p!}{2}\mathfrak{b}\mathfrak{B}^{p-2}\right)^{1/4} \left(\frac{c_0 p}{4\log(p/4)}\right)^{p/4} \left(\frac{1}{\tau(\mu)\log^2 \mu}\right)^{p/4}.
\end{aligned}$$

Thus we have

$$\begin{aligned}
B_p &\leq \left(\frac{p}{2}-1\right)^{p/2} \left(\frac{p!}{2}\mathfrak{b}\mathfrak{B}^{p-2}\right)^{1/4} \left(\frac{c_0 p}{4\log(p/4)}\right)^{p/4} \left(\frac{1}{\tau(\mu)\log^2 \mu}\right)^{p/4} \\
&= \left(\frac{\mathfrak{b}}{2\mathfrak{B}^2}\right)^{1/4} \left(\frac{1}{\tau(\mu)\log^2 \mu}\right)^{p/4} D_p,
\end{aligned}$$

where

$$D_p = \left(\frac{p}{2}-1\right)^{p/2} (p!)^{1/4} \left(\frac{c_0 p\mathfrak{B}}{4\log(p/4)}\right)^{p/4} \leq p! \left(\frac{c_1}{\log p}\right)^{p/4},$$

for some constant c_1. For the last inequality, we use Stirling's formula. This implies that

there exists a universal constant c_2 such that

$$\frac{B_p}{p!} \leq \left(\frac{c_2}{\log p}\right)^{p/4} \left(\frac{1}{\tau(\mu)\log^2 \mu}\right)^{p/4}. \tag{2.28}$$

Combining (2.25) with (2.28) we obtain:

$$\begin{aligned} A_2 &= 1 + \left\{2B_1\tau^{1/4}(\mu) + 2B_2\tau^{1/2}(\mu) + \frac{4B_3}{3}\tau^{3/4}(\mu)\right\} + \sum_{p\geq 4} \frac{2^p \tau(\mu)^{p/4}}{p!} B_p \\ &\leq 1 + \left\{2\Big(B_4\tau(\mu)\Big)^{1/4} + 2\Big(B_4\tau(\mu)\Big)^{1/2} + \frac{4\Big(B_4\tau(\mu)\Big)^{3/4}}{3}\right\} + \sum_{p\geq 4} \left(\frac{16c_2}{\log p}\right)^{p/4} \left(\frac{1}{\log^2 \mu}\right)^{p/4} \\ &< \infty. \end{aligned}$$

The inequality on the last line comes from the fact that $B_4\tau(\mu)\log^2(\mu)$ is bounded.

To control the bias term A_1, let us define, for any $0 < \beta < H_{t_0}$:

$$\Lambda_\beta = \sup_{\substack{u,v\in J_\mu(t_0)\\ u\neq v}} \frac{|X_u^{(\mathbf{d})} - X_v^{(\mathbf{d})}|}{|u-v|^\beta},$$

where here $X_u^{(\mathbf{d})}$ denotes the $\mathbf{d}$-th derivative of the trajectory X_u. Applying Taylor's formula and using the basic properties satisfied by the weights W_m, we obtain:

$$\begin{aligned} \left|\sum_{m=1}^M X(T_m)W_m - X_{t_0}\right| &\leq \left|\sum_{m=1}^M \sum_{k=1}^{\mathbf{d}} \frac{X^{(k)}(t_0)}{k!}(T_m - t_0)^k W_m\right| \\ &\quad + \sum_{m=1}^M \frac{\left|X^{(\mathbf{d})}(t_0) - X^{(\mathbf{d})}(\zeta_m)\right|}{\mathbf{d}!}|T_m - t_0|^{\mathbf{d}}|W_m| \\ &\leq \frac{\Lambda_\beta}{\mathbf{d}!}\sum_{m=1}^M |T_m - t_0|^{\mathbf{d}+\beta}|W_m|, \end{aligned}$$

where $|\zeta_m - t_0| \leq |T_m - t_0|$. Note that this result is obtained using :

$$\sum_{m=1}^M (T_m - t_0)^k W_m = 0.$$

Since, under $\mathcal{E}$ we have, $W_m = 0$ as soon as $|T_m - t_0| > h$, :

$$\left|\sum_{m=1}^{M} X(T_m)W_m - X_{t_0}\right| \mathbf{1}_{\mathcal{E}} \leq \frac{\Lambda_\beta h^{\mathbf{d}+\beta}}{\mathbf{d}!} \sum_{m=1}^{M} |W_m| \mathbf{1}_{\mathcal{E}}$$
$$\leq \frac{\Lambda_\beta}{\mathbf{d}!} \frac{4\kappa}{\lambda_0} \frac{h^{\mathbf{d}+\beta}}{Mh} \sum_{m=1}^{M} \mathbf{1}_{\{t_0-h\leq T_m\leq t_0+h\}}.$$

The last line follows from (2.26). Moreover, combining the result obtained by [116, p. 27], with (H4), for any $0 < H_{t_0} - \beta < \beta_0$ where β_0 is some

$$\mathbb{E}\Lambda_\beta^{p/2} \leq 2^{\frac{1}{4}+\frac{p}{2}(H_{t_0}+1)} \left(\frac{1}{1-2^{\beta-H_{t_0}}}\right)^{p/2} \left(\frac{p!}{2}\mathfrak{a}\mathfrak{A}^{p-2}\right)^{1/4} \leq \frac{\mathfrak{a}^{1/4}}{\mathfrak{A}^{1/2}}(p!)^{1/4} \left(\frac{8\log 2\sqrt{\mathfrak{A}}}{H_{t_0}-\beta}\right)^{p/2}.$$

Since, by definition, the random variable Λ_β is independent of $\widehat{H}_{t_0}$, M and the T_m's, by the last inequality above and inequality (2.27), we have:

$$\mathbb{E}\left(\left|\sum_{m=1}^{M} X(T_m)W_m - X_{t_0}\right|^{p/2} \mathbf{1}_{\mathcal{E}}\mathbf{1}_{\mathcal{F}}\right) \leq \left(\frac{2pCf(t_0)}{\log(p/2)}\right)^{p/2} \mathbb{E}\left[\left(h^{\mathbf{d}+\beta}\right)^{p/2}\right] \mathbb{E}\left[(\Lambda_\beta)^{p/2}\right].$$

We thus obtain:

$$A_1 \leq \sum_{p\geq 0} \frac{\left(16\tau(\mu)\right)^{p/4}}{p!} \mathbb{E}\left(\left|\sum_{m=1}^{M} X(T_m)W_m - X_{t_0}\right|^{p/2} \mathbf{1}_{\mathcal{E}}\mathbf{1}_{\mathcal{F}}\right)$$
$$\leq \sum_{p\geq 0} \frac{\left(16\tau(\mu)\right)^{p/4}}{p!} \left(\frac{2pCf(t_0)}{\log(p/2)}\right)^{p/2} \mathbb{E}\left[\left(h^{\mathbf{d}+\beta}\right)^{p/2} \mathbf{1}_{\mathcal{F}}\right] \mathbb{E}\left[(\Lambda_\beta)^{p/2}\right].$$

Note that, on the event $\mathcal{F}$,

$$\left(h^{\mathbf{d}+\beta}\right)^{p/2} \leq C^{p/2} \left(\frac{\log\mu}{\mu}\right)^{\frac{p}{2}\frac{2(\mathbf{d}+\beta)}{2(\mathbf{d}+H_{t_0})+1}}.$$

Taking $\beta = H_{t_0} - \log^{-1}\mu$, since $(\mu/\log\mu)^{1/\log\mu}$ is bounded, we deduce that, for some constant $C > 0$,

$$A_1 \leq \sum_{p\geq 0} \frac{C^{p/2}}{\log^{p/2}(p)} < \infty.$$

It remains to control (B), (C) and (D). For this purpose, let us first note that, by the

Assumption (LP1), $c_1 := \mathbb{E}[\exp(\eta_0 X_{t_0}^2)] < \infty$, for $\eta_0 = 1/(2\mathfrak{A})$. We deduce that

$$\begin{aligned}\mathbb{E}\left[\varphi^2\Big(\tau(\mu)Z^2\Big)\right] &\leq \mathbb{E}\left[\exp\left(2\tau^{1/4}(\mu)\left|\hat{X}_{t_0} - X_{t_0}\right|^{1/2}\right)\right] \\ &\leq \mathbb{E}\left[\exp\left(2\tau^{1/4}(\mu)\left\{|\hat{X}_{t_0}|^{1/2} + |X_{t_0}|^{1/2}\right\}\right)\right] \\ &\leq \exp\left[2\tau^{1/4}(\mu)\tau^{\alpha/1}(\mu\log(\mu))\right]\mathbb{E}\left[\exp\left(2\tau^{1/4}(\mu)\,|X_{t_0}|^{1/2}\right)\right] \\ &\leq c_1\exp\left[2\tau^{1/4}(\mu)\tau^{\alpha/2}(\mu\log(\mu)) + 2^{4/3}c_2\tau^{1/3}(\mu)\right],\end{aligned}$$

where for the last inequality, we apply Lemma 6 with and $\xi = |X_{t_0}|^{1/2}$, and thus $c_2 = (5\mathfrak{A}/8)^{1/3}$. Now notice that, using Markov's inequality

$$\begin{aligned}\mathbb{P}(\overline{\mathcal{G}}) &= \mathbb{P}(|X_{t_0}| > \tau^{\tilde{\alpha}}(\mu)) \\ &\leq c_1\exp\left(-\eta_0\tau^{2\tilde{\alpha}}(\mu)\right).\end{aligned}$$

Since $\tilde{\alpha} < 1/2$, Assumptions (LP3) and (LP4) imply that for sufficiently large μ:

$$\mathbb{P}(\overline{\mathcal{F}}) \leq 2\mathfrak{K}_1\exp(-\mu) \leq \exp\left(-\eta_0\tau^{2\tilde{\alpha}}(\mu)\right).$$

Moreover, Lemma 5 also implies that, for sufficiently large μ:

$$\mathbb{P}(\overline{\mathcal{E}}) \leq \mathfrak{K}_2\exp\left(-\frac{\mathfrak{g}}{2}\tau(\mu)\log^2(\mu)\right) \leq \exp\left(-\eta_0\tau^{2\tilde{\alpha}}(\mu)\right).$$

Finally, if $\mathcal{H}$ denotes either $\mathcal{E}$, $\mathcal{F}$ or $\mathcal{G}$, we have

$$\mathbb{E}\left[\varphi^2\Big(\tau(\mu)Z^2\Big)\right]\mathbb{P}(\overline{\mathcal{H}}) \leq C\exp\left[2\tau^{1/4}(\mu)\tau^{\alpha/2}(\mu\log(\mu)) + 2^{4/3}c_2\tau^{1/3}(\mu) - \eta_0\tau^{2\tilde{\alpha}}(\mu)\right],$$

where C denotes a positive constant. The choice $\alpha = 5/12$ and $\tilde{\alpha} = 9/24$ allows us to deduce that $\mathbb{E}\left[\varphi^2\Big(\tau(\mu)Z^2\Big)\right]\mathbb{P}(\overline{\mathcal{H}})$ is bounded. This concludes the proof of Theorem 2. □

Proof of Theorem 3. By Theorem 2,

$$\max_{1\leq n_1\leq N_1}\mathbb{E}\left[\varphi\left\{\tau(\mu)\left|\widehat{X}_{t_0}^{[n_1]} - X_{t_0}^{[n_1]}\right|^2\right\}\right] \leq \Gamma_0 \quad \text{where} \quad \tau(\mu) = \frac{1}{\log^2(\mu)}\left(\frac{\mu}{\log(\mu)}\right)^{\frac{2\varsigma_{t_0}}{2\varsigma_{t_0}+1}},$$

and $\varphi(x) = \exp(x^{1/4})$. Now, let $x_0 = 256$ and consider $\widetilde{\varphi} \leq \varphi$ defined by

$$\widetilde{\varphi}(x) = \begin{cases} \varphi'(x_0)(x - x_0) + \varphi(x_0) & \text{if } x \leq x_0 \\ \varphi(x) & \text{if } x \geq x_0 \end{cases},$$

and note that $\widetilde{\varphi}$ is nondecreasing and convex. Then, by Lemma 1.6 in [135],

$$\mathbb{E}\left(\max_{1 \leq n_1 \leq N_1} \left| \widehat{X}_{t_0}^{[n_1]} - X_{t_0}^{[n_1]} \right|^2 \right) \leq \tau^{-1}(\mu) \widetilde{\varphi}^{\leftarrow}(\Gamma_0 N_1),$$

where $\widetilde{\varphi}^{\leftarrow}$ denotes the inverse function of $\widetilde{\varphi}$. Moreover, for N_1 sufficiently large, we have $\widetilde{\varphi}^{\leftarrow}(\Gamma_0 N_1) = \log^4(\Gamma_0 N_1)$. □

2.C Technical lemmas

Lemma 2. *Let $0 < \alpha \leq 3$ be a fixed parameter and let r be an integer such that*

$$(\mu + 1)^{\frac{\beta_f \alpha}{4 + \beta_f \alpha}} \leq 8r \leq K_0 \quad \textit{with} \quad K_0 \leq \frac{\mu}{2 \log(\mu)}.$$

Let $\mathfrak{s} \in \{1, 2, 4, 8\}$ and let $1 \leq k, l \leq K_0$ be such that $l - k = \mathfrak{s} r$. Then, for sufficiently large μ, we have

$$\left| \mathbb{E}_{\mathcal{B}} \left[\left| T_{(l)} - T_{(k)} \right|^{\alpha} \right] - \left(\frac{l - k}{f(t_0)(\mu + 1)} \right)^{\alpha} \right| \leq \mathfrak{c}_1(\alpha) \left(\frac{l - k}{f(t_0)(\mu + 1)} \right)^{\alpha(1 + \beta_f/4)},$$

where $\mathfrak{c}_1(\alpha) = 8\mathfrak{c}_0 (2 f(t_0))^{\beta_f \alpha / 4}$ and $\mathfrak{c}_0$ is defined by (2.45).

Proof of Lemma 2. Let $\mathcal{C} = \{M \geq K_0\} \setminus \mathcal{B}$. We have

$$\mathbb{E}_{\mathcal{B}} \left[\left| T_{(l)} - T_{(k)} \right|^{\alpha} \right] = \mathbb{E} \left[\left| T_{(l)} - T_{(k)} \right|^{\alpha} \mathbf{1}_{\mathcal{B}} \right] = (I) - (II)$$

where

$$(I) = \mathbb{E} \left[\left| T_{(l)} - T_{(k)} \right|^{\alpha} \mathbf{1}_{M \geq K_0} \right] \quad \text{and} \quad (II) = \mathbb{E} \left[\left| T_{(l)} - T_{(k)} \right|^{\alpha} \mathbf{1}_{\mathcal{C}} \right].$$

We study separately the two terms of the right hand side of the above equation.

Study of (II). Note that

$$\mathbb{E} \left[\left| T_{(l)} - T_{(k)} \right|^{\alpha} \mathbf{1}_{\mathcal{C}} \right] \leq |I|^{\alpha} \mathbb{P}(\mathcal{C}).$$

The event $\mathcal{C}$ happens if, less than K_0 random times among $T_1, \ldots, T_M$ fall into the interval $J_\mu(t_0)$. This implies that

$$\mathbb{P}(\mathcal{C}) \leq \mathbb{E}\left[\mathbb{P}(B_M < K_0 \mid M)\mathbf{1}_{\{M \geq K_0\}}\right]$$

where, for any integer $m \geq 1$, B_m denotes a Binomial random variable $\mathcal{B}(m, |J_\mu(t_0)|/|I|)$, independent of M. Using the Bernstein inequality, we obtain

$$\mathbb{P}(B_M < K_0 \mid M) \leq \exp\left(-\frac{2|J_\mu(t_0)|}{|I|}M + 2K_0\right).$$

Since $|J_\mu(t_0)|/|I| \leq (\log(\mu))^{-1}$ and $K_0 \leq (2\log(\mu))^{-1}\mu$, we obtain

$$\begin{aligned}\mathbb{P}(\mathcal{C}) &\leq \exp\left(-\frac{2|J_\mu(t_0)|}{|I|}\mu + 2K_0\right)\mathbb{E}\left[\exp\left(-\frac{2|J_\mu(t_0)|}{|I|}(M-\mu)\right)\mathbf{1}_{\{M\geq K_0\}}\right] \\ &\leq \exp\left(-\frac{\mu}{\log(\mu)}\right)\mathbb{E}\left[\exp\left(-\frac{2|J_\mu(t_0)|}{|I|}(M-\mu)\right)\mathbf{1}_{\{M\geq K_0\}}\right].\end{aligned}$$

To bound the last expectation, let $0 < \epsilon < 1$ be some real number. Then

$$\begin{aligned}&\exp\left(-\frac{\mu}{\log(\mu)}\right)\mathbb{E}\left[\exp\left(-\frac{2|J_\mu(t_0)|}{|I|}(M-\mu)\right)\mathbf{1}_{\{M\geq K_0\}}\right] \\ &\leq \exp\left(-\frac{\mu}{\log(\mu)}\right)\left\{\exp\left(\frac{2|J_\mu(t_0)|}{|I|}\mu\epsilon\right) + \mathbb{E}\left[\exp\left(-\frac{2|J_\mu(t_0)|}{|I|}(M-\mu)\right)\mathbf{1}_{\{K_0\leq M\leq \mu-\mu\epsilon\}}\right]\right\} \\ &\leq \exp\left(-\frac{\mu}{\log(\mu)}\right)\left\{\exp\left(\frac{4\mu\epsilon}{\log(\mu)}\right) + \exp\left(\frac{4\mu}{\log(\mu)}\right)\mathbb{P}\left[|M-\mu| > \mu\epsilon\right]\right\} \\ &\leq \exp\left(-\frac{\mu}{\log(\mu)}\right)\left\{\exp\left(\frac{4\mu\epsilon}{\log(\mu)}\right) + \exp\left(\frac{2\mu}{\log(\mu)}\right)\exp(-\gamma_0\mu\epsilon)\right\}.\end{aligned}$$

This implies that:

$$\mathbb{P}(\mathcal{C}) \leq \exp\left[-\frac{\mu}{\log(\mu)}(1-4\epsilon)\right] + \exp\left[-\mu\epsilon\left(\gamma_0 - \frac{1}{\epsilon\log(\mu)}\right)\right].$$

Taking $\epsilon = 1/8$, we obtain, for sufficiently large μ :

$$\mathbb{P}(\mathcal{C}) \leq 2\exp\left[-\frac{\mu}{2\log(\mu)}\right].$$

We finally obtain, for sufficiently large μ,

$$(II) = \mathbb{E}\left[\left|T_{(l)} - T_{(k)}\right|^{\alpha} \mathbf{1}_{\mathcal{C}}\right] \leq 2|I|^{\alpha} \exp\left[-\frac{\mu}{2\log(\mu)}\right]. \tag{2.29}$$

Study of (I)**.** We define the random variable ρ by the equation:

$$\mathbb{E}\left[\left|T_{(l)} - T_{(k)}\right|^{\alpha} \mid M\right] = \left(\frac{l-k}{f(t_0)(M+1)}\right)^{\alpha}\big(1+\rho\big)$$

Using Lemma SM 2, we have, almost surely:

$$|\rho| \leq \mathfrak{c}_0 \left\{\frac{1}{M} + \frac{1}{\mathfrak{s}r} + \frac{1}{M\mathfrak{s}r} + \left(\frac{\mathfrak{s}r}{M+1}\right)^{\beta_f \alpha/4}\right\}.$$

Whenever $8r \geq (\mu+1)^{\beta_f\alpha/(4+\beta_f\alpha)}$, by bounding smaller terms by the dominant ones and balancing the dominant terms on the right hand side of the last inequality, we have for μ large enough:

$$|\rho| \leq 3\mathfrak{c}_0 \left(\frac{\mathfrak{s}r}{M+1}\right)^{\beta_f\alpha/4} + \mathfrak{c}_0 \left(\frac{\mathfrak{s}r}{\mu+1}\right)^{\beta_f\alpha/4}. \tag{2.30}$$

On the other hand we have

$$\begin{aligned}
\mathbb{E}\left[\left|T_{(l)} - T_{(k)}\right|^{\alpha} \mathbf{1}_{M\geq K_0}\right] &= \mathbb{E}\left(\mathbb{E}\left[\left|T_{(l)} - T_{(k)}\right|^{\alpha} \mid M\right] \mathbf{1}_{M\geq K_0}\right) \\
&= \mathbb{E}\left[\left(\frac{l-k}{f(t_0)(M+1)}\right)^{\alpha}\big(1+\rho\big)\mathbf{1}_{M\geq K_0}\right] \\
&= \left(\frac{l-k}{f(t_0)(\mu+1)}\right)^{\alpha} \mathbb{E}\left[\left(\frac{\mu+1}{M+1}\right)^{\alpha}\big(1+\rho\big)\mathbf{1}_{M\geq K_0}\right]. \qquad (2.31)
\end{aligned}$$

Now, define:

$$t = \frac{(\log(\mu+1))^2}{2} \leq (\mu+1)/2,$$

and consider the following decomposition:

$$\begin{aligned}
\mathbb{E}\left[\left(\frac{\mu+1}{M+1}\right)^{\alpha}\big(1+\rho\big)\mathbf{1}_{M\geq K_0}\right] &= \mathbb{E}\left[\left(\frac{\mu+1}{M+1}\right)^{\alpha}\big(1+\rho\big)\mathbf{1}_{M\geq K_0}\mathbf{1}_{|M-\mu|\leq t}\right] \\
&\quad + \mathbb{E}\left[\left(\frac{\mu+1}{M+1}\right)^{\alpha}\big(1+\rho\big)\mathbf{1}_{M\geq K_0}\mathbf{1}_{|M-\mu|> t}\right].
\end{aligned}$$

Using Assumption (H6), combined with the fact that $r \leq \mu$, the term of the right hand

side can be roughly bounded as follows:

$$\mathbb{E}\left[\left(\frac{\mu+1}{M+1}\right)^{\alpha}\big(1+\rho\big)\mathbf{1}_{M\geq K_0}\mathbf{1}_{|M-\mu|>t}\right] \leq 4\mathfrak{c}_0(\mu+1)^{\alpha(1+\alpha\beta_f/4)}\mathbb{P}\left(|M-\mu|>t\right)$$
$$\leq 4\mathfrak{c}_0(\mu+1)^{\alpha(1+\alpha\beta_f/4)}\exp\left(-\frac{\gamma_0}{2}\left(\log(\mu+1)\right)^2\right).$$

Thus, for sufficiently large μ, we have:

$$\mathbb{E}\left[\left(\frac{\mu+1}{M+1}\right)^{\alpha}\big(1+\rho\big)\mathbf{1}_{M\geq K_0}\mathbf{1}_{|M-\mu|>t}\right] \leq \frac{1}{\mu+1}. \tag{2.32}$$

It remains to study the term

$$\mathbb{E}\left[\left(\frac{\mu+1}{M+1}\right)^{\alpha}\big(1+\rho\big)\mathbf{1}_{M\geq K_0}\mathbf{1}_{|M-\mu|\leq t}\right].$$

To do so, let us define

$$\tilde{\rho}_\alpha = \left(\frac{\mu+1}{M+1}\right)^{\alpha} - 1.$$

Since $K_0 < (\mu+1)/2$, we have

$$\left(\frac{\mu+1}{M+1}\right)^{\alpha}\big(1+\rho\big)\mathbf{1}_{M\geq K_0}\mathbf{1}_{|M-\mu|\leq t} = \big(1+\tilde{\rho}_\alpha\big)\big(1+\rho\big)\mathbf{1}_{|M-\mu|\leq t}$$
$$= 1 + \big(\tilde{\rho}_\alpha + \rho + \tilde{\rho}_\alpha\rho\big)\mathbf{1}_{|M-\mu|\leq t} - \mathbf{1}_{|M-\mu|>t}. \tag{2.33}$$

Under the event $\{|M-\mu \leq t|\}$, since $t < (\mu+1)/2$, we have:

$$1 - \frac{\alpha t}{\mu+1} \leq \left(\frac{\mu+1}{M+1}\right)^{\alpha} \leq 1 + \frac{2(2^\alpha-1)t}{\mu+1},$$

which leads to

$$|\tilde{\rho}_\alpha|\,\mathbf{1}_{\{|M-\mu\leq t|\}} \leq \frac{2(2^\alpha-1)t}{\mu+1} = (2^\alpha-1)\frac{\log^2(\mu+1)}{\mu+1}. \tag{2.34}$$

Note also that by (2.30):

$$|\rho|\mathbf{1}_{|M-\mu|\leq t} \leq 4\mathfrak{c}_0\left(\frac{2\mathfrak{s}r}{\mu+1}\right)^{\beta_f\alpha/4}. \tag{2.35}$$

Gathering (2.33), (2.34) and (2.35) we obtain, for sufficiently large μ :

$$\left|\mathbb{E}\left[\left(\frac{\mu+1}{M+1}\right)^{\alpha}\big(1+\rho\big)\mathbf{1}_{M\geq K_0}\mathbf{1}_{|M-\mu|\leq t}\right]-1\right| \leq 5\mathfrak{c}_0\left(\frac{2\mathfrak{s}r}{\mu+1}\right)^{\beta_f\alpha/4}+\mathbb{P}(|M-\mu|>t)$$
$$\leq 6\mathfrak{c}_0\left(\frac{2\mathfrak{s}r}{\mu+1}\right)^{\beta_f\alpha/4}. \tag{2.36}$$

Combining (2.31) with (2.32) and (2.36), we obtain, for sufficiently large μ :

$$\mathbb{E}\left[\left|T_{(l)}-T_{(k)}\right|^{\alpha}\mathbf{1}_{M\geq K_0}\right]=\left(\frac{l-k}{f(t_0)(\mu+1)}\right)^{\alpha}\left(1+\tilde{R}\right), \tag{2.37}$$

where

$$|\tilde{R}|\leq 7\mathfrak{c}_0\left(\frac{2\mathfrak{s}r}{\mu+1}\right)^{\beta_f\alpha/4}.$$

From (2.29) and (2.37), we obtain, for μ large enough:

$$\mathbb{E}_{\mathcal{B}}\left[\left|T_{(l)}-T_{(k)}\right|^{\alpha}\right]=\left(\frac{l-k}{f(t_0)(\mu+1)}\right)^{\alpha}(1+R),$$

where

$$|R|\leq 8\mathfrak{c}_0\left(\frac{2\mathfrak{s}r}{\mu+1}\right)^{\beta_f\alpha/4}=8\mathfrak{c}_0(2f(t_0))^{\beta_f\alpha/4}\left(\frac{l-k}{f(t_0)(\mu+1)}\right)^{\beta_f\alpha/4}.$$

This ends the proof. □

Let us recall the definitions

$$A=A_{M,h}=\frac{1}{Mh}\sum_{m=1}^{M}U\left(\frac{T_m-t_0}{h}\right)U^{\top}\left(\frac{T_m-t_0}{h}\right)K\left(\frac{T_m-t_0}{h}\right), \tag{2.38}$$

and

$$\mathbf{A}=f(t_0)\int_{\mathbb{R}}U(u)U^{\top}(u)K(u)du,$$

with $U(u)=(1,u,\ldots,u^{\hat{\mathbf{d}}}/\hat{\mathbf{d}}!)$. Moreover, λ and λ_0 are the smallest eigenvalues of A and $\mathbf{A}$, respectively. The matrix $\mathbf{A}$ is positive definite and thus $\lambda_0>0$. See [135]. The following result shows that, with high probability, λ stays away from zero. Let us recall that in our context, $\hat{\mathbf{d}}$ is a generic estimator of $\mathbf{d}$, independent of the online set of curves. Since dimension of the matrices A and $\mathbf{A}$ are given by this estimator, the probability $\mathbb{P}(\cdot)$ in Lemma 5 should be understood as the conditional probability given the estimator $\hat{\mathbf{d}}$.

Finally, recall that

$$\widehat{h} = \left(\frac{1}{M}\right)^{1/(2\widehat{\varsigma}_{t_0}+1)}.$$

Lemma 5. *Let $K(\cdot)$ be a kernel such that, for any $t \in \mathbb{R}$:*

$$\kappa^{-1}\mathbf{1}_{[-\delta,\delta]}(t) \leq K(t) \leq \kappa \mathbf{1}_{[-1,1]}(t), \quad \text{for some } 0 < \delta < 1 \text{ and } \kappa \geq 1. \tag{2.39}$$

Under Assumptions (LP2), (LP3) and (LP4), the matrix A defined as in (2.38), with $h = \widehat{h}$, is positive semidefinite. Moreover, there exists a positive constant $\mathfrak{g}$ that depends only on K, $\mathbf{d}$, $f(t_0)$ and λ_0 such that, for M sufficiently large,

$$\mathbb{P}(\lambda \leq \beta | M) \leq 2\exp(-\mathfrak{g} M \widehat{h}), \quad \forall 0 < \beta \leq \lambda_0/2, \tag{2.40}$$

and, for sufficiently large μ,

$$\sup_{0<\beta\leq\lambda_0/2} \mathbb{P}(\lambda \leq \beta) \leq \mathfrak{K}_2 \exp\left[-\frac{\mathfrak{g}}{2}\tau(\mu)\log^2(\mu)\right] \quad \text{where} \quad \tau(\mu) = \frac{1}{\log^2(\mu)}\left(\frac{\mu}{\log(\mu)}\right)^{\frac{2\varsigma_{t_0}}{2\varsigma_{t_0}+1}}, \tag{2.41}$$

with $\varsigma_{t_0} = \mathbf{d} + H_{t_0}$. Here, $\mathfrak{K}_2$ is a universal constant.

Proof of Lemma 5. Without loss of generality, we could work on the set $\{\hat{\mathbf{d}} = \mathbf{d}\}$. Moreover, for simplicity, we write h instead of $\widehat{h}$ below in this proof.

Note that, using Assumption (LP2), for any $1 \leq i \leq j \leq \mathbf{d}$, the element $A_{i,j}$ tends almost surely to the element $\mathbf{A}_{i,j}$ as μ goes to infinity. This implies that the matrix A tends to the matrix $\mathbf{A}$. This also implies that, for sufficiently large μ, we have $\lambda > 0$. More precisely, we have:

$$|\lambda - \lambda_0| \leq \|A - \mathbf{A}\|_2 \leq (\mathbf{d}+1)\|A - \mathbf{A}\|_\infty,$$

where $\|\cdot\|_2$ denotes the norm induced by the Euclidean norm whereas $\|\cdot\|_\infty$ denotes the entrywise sup-norm. Let $\tilde{\mathbb{P}}(\cdot)$ and $\tilde{\mathbb{E}}(\cdot)$ denote the conditional probability $\mathbb{P}(\cdot|M)$ and conditional expectation $\mathbb{E}(\cdot|M)$, respectively. Then:

$$\tilde{\mathbb{P}}(\lambda \leq \beta) \leq \tilde{\mathbb{P}}(|\lambda - \lambda_0| \geq \lambda_0/2) \leq \sum_{0\leq i,j\leq \mathbf{d}} \tilde{\mathbb{P}}\left(|(A_n)_{i,j} - \mathbf{A}_{i,j}| \geq \lambda_0/\{2(\mathbf{d}+1)\}\right).$$

Next, we decompose

$$A_{i,j} - \mathbf{A}_{i,j} = A_{i,j} - \tilde{\mathbb{E}}(A_{i,j}) + \tilde{\mathbb{E}}(A_{i,j}) - \mathbf{A}_{i,j}.$$

Using Assumption (H2) and the fact that $K(\cdot)$ has the support $[-1, 1]$, we have:

$$\begin{aligned}
\left|\tilde{\mathbb{E}}(A_{i,j}) - \mathbf{A}_{i,j}\right| &\leq \left| \int_{\mathbb{R}} \left[U(u)U^\top(u)\right]_{i,j} K(u)\Big\{f(t_0 + hu) - f(t_0)\Big\} du \right| \\
&= L_f h^{\beta_f} \int_{\mathbb{R}} \left| \left[U(u)U^\top(u)\right]_{i,j} uK(u) \right| du \\
&\leq L_f h^{\beta_f} \int_{\mathbb{R}} |u|\, K(u) du \\
&=: L_f \|K\|_1 h^{\beta_f}.
\end{aligned}$$

This implies that, for h sufficiently small, that is for M sufficiently large,

$$\begin{aligned}
\tilde{\mathbb{P}}(\lambda \leq \beta) &\leq \sum_{0 \leq i,j \leq \mathbf{d}} \tilde{\mathbb{P}}(|A_{i,j} - \tilde{\mathbb{E}}(A_{i,j})| \geq \lambda_0/\{2(\mathbf{d}+1)\} - L_f \|K\|_1 h^{\beta_f}) \\
&\leq \sum_{0 \leq i,j \leq \mathbf{d}} \tilde{\mathbb{P}}(|A_{i,j} - \tilde{\mathbb{E}}(A_{i,j})| > \lambda_0/\{4(\mathbf{d}+1)\}).
\end{aligned}$$

Let us define

$$\begin{aligned}
\xi_{m,i,j} &= \left[U\left(\frac{T_m - t_0}{h}\right) U^\top \left(\frac{T_m - t_0}{h}\right)\right]_{i,j} K\left(\frac{T_m - t_0}{h}\right) \\
&= \frac{1}{i!j!} \left(\frac{T_m - t_0}{h}\right)^{i+j} K\left(\frac{T_m - t_0}{h}\right).
\end{aligned}$$

By property (2.39), we have

$$\left|\xi_{m,i,j} - \tilde{\mathbb{E}}(\xi_{m,i,j})\right| \leq 2\kappa.$$

Moreover, for h sufficiently small, that is for M sufficiently large, $f(t) \leq 2f(t_0)$, $\forall |t - t_0| \leq h$, and thus

$$\begin{aligned}
\sum_{m=1}^{M} \widetilde{\mathrm{Var}}(\xi_{i,j}^{(m)}) &\leq \sum_{m=1}^{M} \tilde{\mathbb{E}}\left[\{\xi_{i,j}^{(m)}\}^2\right] \\
&\leq 2f(t_0) Mh \int_{\mathbb{R}} \left| \left[U(u)U^\top(u)\right]_{i,j} \right| K^2(u) du \\
&\leq 2f(t_0) \|K\|_2^2\, Mh.
\end{aligned}$$

Applying the Bernstein inequality [see 123, p. 95], we obtain, for any $x > 0$:

$$\tilde{\mathbb{P}}\left(\frac{1}{Mh}\sum_{m=1}^{M}\left|\xi_{m,i,j} - \tilde{\mathbb{E}}(\xi_{m,i,j})\right| > x\right) \leq 2\exp\left(-\frac{M^2x^2}{\frac{2\|f\|_\infty\|K\|_2^2M}{h} + \frac{4\kappa xM}{3h}}\right).$$

Then equation (2.40) follows if we define:

$$\mathfrak{g} = \psi\left(\frac{\lambda_0}{4(\mathbf{d}+1)}\right) \quad \text{with} \quad \psi(x) = \frac{x^2}{2\|f\|_\infty\|K\|_2^2 + \frac{4\kappa x}{3}}.$$

It remains to prove (2.41). Let us define the events

$$\mathcal{E}_\beta = \{\lambda > \beta\} \quad \text{and} \quad \mathcal{F} = \left\{|\widehat{H}_{t_0} - H_{t_0}| \leq \log^{-2}(\mu)\right\} \cap \left\{\hat{\mathbf{d}} = \mathbf{d}\right\}.$$

Using (2.40), we have

$$\begin{aligned}\mathbb{P}(\overline{\mathcal{E}}_\beta) &= 2\mathbb{E}[\exp(-\mathfrak{g}Mh)\mathbf{1}_{\mathcal{F}}] + \mathbb{P}(\overline{\mathcal{F}}) \\ &\leq 2\mathbb{E}[\exp(-\mathfrak{g}Mh)\mathbf{1}_{\mathcal{F}}] + 2\mathfrak{K}_1\exp(-\mu).\end{aligned}$$

The last line comes from Assumption (LP3). Note that under $\mathcal{F}$

$$Mh = M^{\frac{2\{\hat{\mathbf{d}}+\widehat{H}_{t_0}\}}{2\{\hat{\mathbf{d}}+\widehat{H}_{t_0}\}+1}} = M^{\frac{2\{\mathbf{d}+\widehat{H}_{t_0}\}}{2\{\mathbf{d}+\widehat{H}_{t_0}\}+1}} = M^{\frac{2\{\mathbf{d}+H_{t_0}\}}{2\{\mathbf{d}+H_{t_0}\}+1}+\eta},$$

with

$$|\eta| = \left|\frac{2(\widehat{H}_{t_0} - H_{t_0})}{(2\{\hat{\mathbf{d}}+\widehat{H}_{t_0}\}+1)(2\{\mathbf{d}+H_{t_0}\}+1)}\right| \leq 2|\widehat{H}_{t_0} - H_{t_0}|.$$

Assumption (LP2) implies that, under $\mathcal{F}$ and, for sufficiently large μ,

$$Mh \geq \left(\frac{\mu}{\log(\mu)}\right)^{\frac{2\{\mathbf{d}+H_{t_0}\}}{2\{\mathbf{d}+H_{t_0}\}+1}-\frac{2}{\log^2(\mu)}} \geq \frac{1}{2}\left(\frac{\mu}{\log(\mu)}\right)^{\frac{2\{\mathbf{d}+H_{t_0}\}}{2\{\mathbf{d}+H_{t_0}\}+1}}. \tag{2.42}$$

Thus, we have

$$\begin{aligned}\mathbb{P}(\overline{\mathcal{E}}_\beta) &\leq 2\exp\left(-\frac{\mathfrak{g}}{2}\tau(\mu)\log^2(\mu)\right) + 2\mathfrak{K}_1\exp(-\mu) \\ &\leq \mathfrak{K}_2\exp\left[-\frac{\mathfrak{g}}{2}\tau(\mu)\log^2(\mu)\right],\end{aligned}$$

for some positive constant $\mathfrak{K}_2$, that does not depend on $0 < \beta \leq \lambda_0/2$. □

Lemma 6. *Let ξ be a positive random variable such that*

$$c_1 := \mathbb{E}\left[\exp\left(\eta_0 \xi^4\right)\right] < \infty,$$

for some positive constant η_0. Then, for any $\tau \geq 1$:

$$\mathbb{E}\left[\exp\left(\tau\xi\right)\right] \leq c_1 \exp\left(c_2 \tau^{4/3}\right) \qquad \textit{where} \qquad c_2 = \left(\frac{5}{16\eta_0}\right)^{1/3}.$$

Proof of Lemma 6. Defining $\zeta = (16\eta_0/5)^{1/4}\xi$, we can assume, without loss of generality that $\eta_0 = 5/16$. Let $\gamma \geq \tau$. Remark that, since

$$1 - \frac{\tau\xi}{\gamma} = \left(1 - \frac{\tau\xi}{4\gamma}\right)^4 - 6\left(\frac{\tau\xi}{4\gamma}\right)^2 + 4\left(\frac{\tau\xi}{4\gamma}\right)^3 - \left(\frac{\tau\xi}{4\gamma}\right)^4,$$

we obtain:

$$\begin{aligned}
\mathbb{E}\left[\exp\left(\tau\xi\right)\right] &= \exp(\gamma)\mathbb{E}\left[\exp\left(-\gamma\left(1 - \frac{\tau\xi}{\gamma}\right)\right)\right] \\
&\leq \exp(\gamma)\mathbb{E}\left[\exp\left(\frac{3\gamma}{8}\left(\frac{\tau\xi}{\gamma}\right)^2\right)\exp\left(\frac{\gamma}{256}\left(\frac{\tau\xi}{\gamma}\right)^4\right)\right] \\
&\leq \exp(\gamma+\eta)\mathbb{E}\left[\exp\left(-\eta\left(1 - \frac{3\gamma}{8\eta}\left(\frac{\tau\xi}{\gamma}\right)^2\right)\right)\exp\left(\frac{\gamma}{256}\left(\frac{\tau\xi}{\gamma}\right)^4\right)\right].
\end{aligned}$$

Using the fact that

$$1 - \frac{3\gamma}{8\eta}\left(\frac{\tau\xi}{\gamma}\right)^2 = \left[1 - \frac{3\gamma}{16\eta}\left(\frac{\tau\xi}{\gamma}\right)^2\right]^2 - \left[\frac{3\gamma}{16\eta}\left(\frac{\tau\xi}{\gamma}\right)^2\right]^2,$$

we obtain:

$$\mathbb{E}\left[\exp\left(\tau\xi\right)\right] \leq \exp(\gamma+\eta)\mathbb{E}\left[\exp\left(\frac{9}{256}\frac{\tau^4\xi^4}{\eta\gamma^2} + \frac{1}{256}\frac{\tau^4\xi^4}{\gamma^3}\right)\right].$$

Taking $\gamma = \eta = \tau^{4/3}/2$, we obtain:

$$\mathbb{E}\left[\exp\left(\tau\xi\right)\right] \leq \exp(\tau^{4/3})\mathbb{E}\left[\exp\left(\frac{5}{16}\xi^4\right)\right].$$

This completes the proof. □

2.D Moment bounds for spacings

We need to find an accurate approximation for moments like

$$\mathbb{E}[(T_{(k)} - T_{(l)})^\alpha \mid M = m]$$

where $1 \leq l < k \leq K_0 \leq m$, $\alpha > 0$. Here, $T_{(1)} \leq \ldots \leq T_{(K_0)}$ are defined as in section 2.2, that is the subvector of the K_0 closest values to t_0. We assume that T admits a density f. Such moments will be considered with k and l such that, for some fixed value $t_0 \in [0, 1]$ such that $f(t_0) > 0$,

$$\frac{\max(|\lfloor t_0 m \rfloor - k|, |\lfloor t_0 m \rfloor - l|)}{m+1} \leq 8\frac{k-l}{m+1} \tag{2.43}$$

and

$$\frac{k-l}{m+1} \quad \text{is small,} \tag{2.44}$$

and converges to zero when $m \to \infty$. Herein, for any real number a, $\lfloor a \rfloor$ denotes the largest integer smaller than or equal to a. These conditions on k and l allows for $(k-l)$ increasing slower than m.

Let us point out that $T_{(1)} \leq \ldots \leq T_{(K_0)}$ defined in section 2.2 is not the order statistics from a random sample of T. In fact, $T_{(k)}$, with $1 \leq k \leq K_0$, is the $(G+k)-$th order statistics of the sample $T_1, \ldots, T_m$. Here G is a random variable and its value is determined by the way the subvector of K_0 closest values to t_0 is built. It is important to notice that G depends of the smallest and the largest values in this subvector, but is independent of the other components of the subvector. In particular, this means that in the case where T has a uniform distribution, the law of the spacings between $T_{(1)} \leq \ldots \leq T_{(K_0)}$ coincides with the law of the same type of spacings between the order statistics of a uniform sample of size m on $[0, 1]$. In particular, in the uniform case, the law of $T_{(k)} - T_{(l)}$ depends only on m and $k - l$. For this reason, first we consider the case of T with uniform law. In the general case, we use the transformation by the distribution function in order to get back to the uniform case.

2.D.1 The uniform case

Consider U a uniform random variable on $[0,1]$. Let $U_1, \ldots, U_m$ be an independent sample of U and let $U_{(1)}, \ldots, U_{(m)}$ be the order statistics. In the case of a uniform sample, $U_{(k)} - U_{(l)}$ and $U_{(k-l)}$ have the same distribution, that is a beta distribution $\mathrm{Beta}(k-l, m-(k-l)+1)$. Hence in this case, it is equivalent to study the moments of $U_{(r)}$ with $1 \leq r = k-l \leq m-1$. The variable $U_{(r)}$ has a $\mathrm{Beta}(r, m-r+1)$ distribution. It also worthwhile to notice that $U_{(k)} - U_{(l)}$ and $U_{(l)}$ are independent, and the same is true for $U_{(k)} - U_{(l)}$ and $U_{(k)}$.

By elementary calculations, we have

$$\mathbb{E}\left[U_{(r)}^{\alpha}\right] = \frac{B(\alpha+r, m-r+1)}{B(r, m-r+1)} = \frac{\Gamma(\alpha+r)}{\Gamma(r)} \frac{\Gamma(m+1)}{\Gamma(m+\alpha+1)},$$

where $B(\cdot,\cdot)$ denotes the beta function and $\Gamma(\cdot)$ the gamma function. To derive the bounds for the moments of interest, we use some existing results on the approximation of the gamma functions and the ratios of the gamma functions. The results are recalled in Section 2.D.3 below.

Moment bounds in the uniform case

Let M be a random variable taking positive integer values. In the following proposition we assume that, given the realization of $M \geq K_0$, $T_1, \ldots, T_M$ be an independent sample with uniform distribution on $[0,1]$.

Lemma SM 1. *Consider $0 < \alpha \leq 3$ and $1 \leq l < k \leq m$, and let $r = k - l$. Then, for any $m \geq K_0$ in the support of M,*

$$\left| \mathbb{E}\left[(T_{(k)} - T_{(l)})^{\alpha} \mid M = m\right] - \frac{\Gamma(\alpha+r)}{\Gamma(r)} \frac{1}{(m+1)^{\alpha}} \right| \leq \frac{3}{m} \frac{\Gamma(\alpha+r)}{\Gamma(r)} \frac{1}{(m+1)^{\alpha}}$$

and

$$\left| \mathbb{E}\left[(T_{(k)} - T_{(l)})^{\alpha} \mid M = m\right] - \left(\frac{r}{m+1}\right)^{\alpha} \right| \leq \left(\frac{r}{m+1}\right)^{\alpha} \left[\frac{3}{m} + \frac{4}{r} + \frac{12}{mr}\right].$$

Proof of Lemma SM 1. Given that $M = m$, $T_{(k)} - T_{(l)}$ is distributed as $U_{(r)}$, the $r-$th order statistic, with $1 \leq r = k - l \leq m - 1$, of an independent sample of size m from the

uniform law on $[0,1]$. Using inequality (2.50) with $x = m+1$ and $s = \alpha$, we can write

$$\begin{aligned}\left|\mathbb{E}\left[U_{(r)}^{\alpha} \mid M=m\right] - \frac{\Gamma(\alpha+r)}{\Gamma(r)}\frac{1}{(m+1)^{\alpha}}\right| & \\ = \frac{\Gamma(\alpha+r)}{\Gamma(r)}\frac{1}{(m+1)^{\alpha}}\left|\frac{(m+1)^{\alpha}\Gamma(m+1)}{\Gamma(m+\alpha+1)} - 1\right| & \\ \leq \frac{3}{m}\frac{\Gamma(\alpha+r)}{\Gamma(r)}\frac{1}{(m+1)^{\alpha}}. &\end{aligned}$$

Next, using inequality (2.49) twice, with $x = r$ and $s = \alpha$, and triangle inequality

$$\begin{aligned}\left|\mathbb{E}\left[U_{(r)}^{\alpha} \mid M=m\right] - \left(\frac{r}{m+1}\right)^{\alpha}\right| &\leq \left|\mathbb{E}\left[U_{(r)}^{\alpha} \mid M=m\right] - \frac{\Gamma(\alpha+r)}{\Gamma(r)}\frac{1}{(m+1)^{\alpha}}\right| \\ &\quad + \left(\frac{r}{m+1}\right)^{\alpha}\left|\frac{\Gamma(\alpha+r)}{r^{\alpha}\Gamma(r)} - 1\right| \\ &\leq \left(\frac{r}{m+1}\right)^{\alpha}\left[\frac{3}{m}\frac{\Gamma(\alpha+r)}{r^{\alpha}\Gamma(r)} + \frac{4}{r}\right] \\ &\leq \left(\frac{r}{m+1}\right)^{\alpha}\left[\frac{3}{m}\left(1+\frac{4}{r}\right) + \frac{4}{r}\right].\end{aligned}$$

□

2.D.2 The general case

Given the realization of M, let $T_1, T_2, \ldots$ be an independent sample from T, a random variable independent of M, with an absolute continuous distribution on $[0,1]$. Let f (resp. F) (resp. Q) denote the density (resp. distribution function) (resp. quantile function) of T. We assume that F is strictly increasing on $[0,1]$ and thus Q is the inverse function for F, and Q is differentiable with $Q' = 1/f$. Then, given $M = m$, for any $1 \leq l < k \leq m$, the joint distribution of the order statistics $(T_{(k)}, T_{(l)})$ is the same as the joint distribution of $(Q(U_{(k)}), Q(U_{(l)}))$, where $U_{(1)}, \ldots, U_{(m)}$ is the order statistics of an independent uniform sample on $[0,1]$.

Moment bounds in the general case

Assume $\inf_{t\in[0,1]} f(t) > 0$ and f is Hölder continuous around t_0, i.e. there exists $L_f > 0$, $0 < \beta_f \leq 1$, and a neighborhood of t_0 in $[0,1]$ such that for any u, v in this neighborhood, $|f(u) - f(v)| \leq L_f|u-v|^{\beta_f}$.

Lemma SM 2. *Let m be an integer value in the support of M. Let $t_0 \in [0,1]$, assume that k and l are satisfying the conditions (2.43)-(2.44), and let $r = k - l$. The for any $0 < \alpha \leq 3$,*

$$\left| \mathbb{E}\left[(T_{(k)} - T_{(l)})^\alpha \mid M = m \right] - \frac{\Gamma(\alpha + r)}{\Gamma(r)} \left(\frac{1}{f(t_0)(m+1)} \right)^\alpha \right| \leq \frac{\Gamma(\alpha + r)}{\Gamma(r)} \left(\frac{1}{f(t_0)(m+1)} \right)^\alpha \left[\frac{3}{m} + C \left(\frac{r}{m+1} \right)^{\alpha\beta_f/4} \right],$$

and

$$\begin{aligned} &\left| \mathbb{E}\left[(T_{(k)} - T_{(l)})^\alpha \mid M = m \right] - \left(\frac{r}{f(t_0)(m+1)} \right)^\alpha \right| \\ &\quad \leq \left(\frac{r}{f(t_0)(m+1)} \right)^\alpha \left[\frac{3}{m} + \frac{4}{r} + \frac{12}{mr} + C \left(\frac{r}{m+1} \right)^{\alpha\beta_f/4} \right] \\ &\quad \leq \mathfrak{c}_0 \left(\frac{r}{f(t_0)(m+1)} \right)^\alpha \left[\frac{1}{m} + \frac{1}{r} + \frac{1}{mr} + \left(\frac{r}{m+1} \right)^{\alpha\beta_f/4} \right], \end{aligned} \tag{2.45}$$

with C and $\mathfrak{c}_0$ are two constants depending only on α and L_f, β_f and $f(t_0)$.

Proof of Lemma SM 2. In the following, we use several times the following property: for any $a, b, \alpha \geq 0$,

$$(a+b)^\alpha \leq \max(1, 2^{\alpha-1}) \left(a^\alpha + b^\alpha \right).$$

Next, given $M = m$, $\mathbb{E}\left[(T_{(k)} - T_{(l)})^\alpha \mid M = m \right] = \mathbb{E}\left[\{Q(U_{(k)}) - Q(U_{(l)})\}^\alpha \mid M = m \right]$. By a first order Taylor expansion of $Q(U_{(k)})$ around the point $U_{(l)}$, we get

$$Q(U_{(k)}) - Q(U_{(l)}) = \frac{1}{f(t_0)} \left[U_{(k)} - U_{(l)} \right] \left[1 + r(m,k,l) \right], \tag{2.46}$$

with

$$r(m,k,l) = \int_0^1 \frac{f(t_0) - f(U_{(l)} + t[U_{(k)} - U_{(l)}])}{f(U_{(l)} + t[U_{(k)} - U_{(l)}])} dt.$$

Note that due to the fact the Q is increasing and almost surely$U_{(k)} > U_{(l)}$, the identity (2.46) implies that $1 + r(m,k,l) > 0$ almost surely. Using the triangle inequality and the properties of f,

$$|r(m,k,l)| \leq \frac{L_f}{f(t_0)/2} \left(|U_{(l)} - t_0|^{\beta_f} + |U_{(k)} - U_{(l)}|^{\beta_f} \right).$$

Let

$$t_m = \frac{\lfloor t_0(m-1) \rfloor + 1}{m+1}.$$

Note that $1/(m+1) \le t_m \le m/(m+1)$ and

$$t_m = \mathbb{E}[U_{(t_m(m+1))}].$$

Next, we can bound

$$\begin{aligned} |U_{(l)} - t_0| &\le |U_{(l)} - t_m| + |t_m - t_0| \le |U_{(l)} - \mathbb{E}[U_{(t_m(m+1))}]| + \frac{2}{m+1} \\ &\le |U_{(l)} - U_{(t_m(m+1))}| + |U_{(t_m(m+1))} - \mathbb{E}[U_{(t_m(m+1))}]| + \frac{2}{m+1}. \end{aligned}$$

Thus, with the convention $U_{(0)} = 0$,

$$\begin{aligned} \mathbb{E}\left[|U_{(l)} - t_0|^{\beta_f} \mid M = m\right] &\le \mathbb{E}\left[U^{\beta_f}_{(|l - t_m(m+1)|)} \mid M = m\right] \\ &\quad + \mathbb{E}\left[\left|U_{(t_m(m+1))} - \mathbb{E}[U_{(t_m(m+1))}]\right|^{\beta_f} \mid M = m\right] + \left(\frac{2}{m+1}\right)^{\beta_f}. \end{aligned}$$

By the facts presented in the uniform case, when $l \neq t_m(m+1)$,

$$\mathbb{E}\left[U^{\beta_f}_{(|l - t_m(m+1)|)} \mid M = m\right] = \frac{\Gamma(\beta_f + |l - t_m(m+1)|)}{\Gamma(|l - t_m(m+1)|)} \frac{\Gamma(m+1)}{\Gamma(m + \beta_f + 1)},$$

and using Wendel's double inequality (2.48) with $s = \beta_f$, and (2.43), the product of the ratios of the gamma functions is bounded from above by

$$\left(\frac{|l - t_m(m+1)|}{m + 1 + \beta_f}\right)^{\beta_f} \left(1 + \frac{\beta_f}{m+1}\right) \le 9 \left(\frac{r}{m+1}\right)^{\beta_f}.$$

On the other hand, using Jensen's inequality and the variance of a beta distribution with parameters $t_m(m+1)$ and $(1-t_m)(m+1)$,

$$\begin{aligned} &\mathbb{E}\left[\left|U_{(t_m(m+1))} - \mathbb{E}[U_{(t_m(m+1))}]\right|^{\beta_f} \mid M = m\right] \\ &\qquad \le \mathbb{E}^{\beta_f/2}\left[\left|U_{(t_m(m+1))} - \mathbb{E}[U_{(t_m(m+1))}]\right|^2 \mid M = m\right] = \left(\frac{t_m(1-t_m)}{m+2}\right)^{\beta_f/2}. \end{aligned}$$

Gathering facts and using Lemma SM 1, there exists a constant c such that

$$\mathbb{E}\left[|U_{(l)} - t_0|^{\beta_f} \mid M = m\right] \leq c\left(\frac{r}{m+1}\right)^{\beta_f/2}.$$

On the other hand, since $U_{(k)} - U_{(l)}$ is independent of $U_{(l)}$, from above and Lemma SM 1 we deduce that for any $0 < \alpha' \leq \alpha \leq 3$,

$$\mathbb{E}\left[\{U_{(k)} - U_{(l)}\}^{\alpha}|r(m,k,l)|^{\alpha'} \mid M = m\right] \leq C\left(\frac{r}{m+1}\right)^{\alpha+\alpha'\beta_f/2} \tag{2.47}$$

for some constant C depending on L_f, β_f and $f(t_0)$.

Coming back to relationship (2.46), taking power α on both sides of the identity, we can write

$$\mathbb{E}\left[\{Q(U_{(k)}) - Q(U_{(l)})\}^{\alpha} \mid M = m\right] = \frac{1}{f^{\alpha}(t_0)}\mathbb{E}\left[U_{(r)}^{\alpha} \mid M = m\right] + R(m,k,l)$$

with

$$R(m,k,l) = \mathbb{E}\left[\{U_{(k)} - U_{(l)}\}^{\alpha}\{[1 + r(m,k,l)]^{\alpha} - 1\} \mid M = m\right].$$

Since for any $a > -1$ and $0 < \alpha \leq 3$,

$$|(1+a)^{\alpha} - 1| = |(1+a)^{\alpha/2} - 1||(1+a)^{\alpha/2} + 1| \leq 2|a|^{\alpha/2}(|a|^{\alpha/2} + 2),$$

using the bound (2.47) with $\alpha' = \alpha$ and $\alpha' = \alpha/2$,

$$|R(m,k,l)| \leq c_R\left(\frac{r}{m+1}\right)^{\alpha(1+\beta_f/4)},$$

for some constant c_R depending on L_f, β_f and $f(t_0)$. It remains to apply Lemma SM 1 to complete the proof. □

2.D.3 Wendel's type inequalities for gamma function ratios

Since in our case, we only need to consider $\alpha \in (0, 3]$, we could use the sharp bounds for the ratio of two gamma functions, as deduced by [140]. For any $x > 0$ and $s \geq 0$, let

$$R(x,s) = \frac{\Gamma(x+s)}{\Gamma(x)}.$$

[140] proved that when $0 \leq s \leq 1$,

$$\left(\frac{1}{1+s/x}\right)^{1-s} \leq \frac{R(x,s)}{x^s} \leq 1. \tag{2.48}$$

Since

$$1 - \frac{s}{x} \leq \left(\frac{1}{1+s/x}\right)^{1-s}, \qquad \forall x \geq 1, 0 \leq s \leq 1,$$

we can deduce that, when $0 \leq s \leq 1$,

$$1 - \frac{1}{x} \leq 1 - \frac{s}{x} \leq \frac{R(x,s)}{x^s} \leq 1, \qquad \forall x \geq 1.$$

Next, using the recurrence formula for the gamma function, when $1 \leq s \leq 2$ we can write

$$\frac{R(x,s)}{x^s} = \left(1 + \frac{s-1}{x}\right) \frac{R(x,s-1)}{x^{s-1}}$$

and deduce

$$1 - \frac{1}{x} \leq \left(1 + \frac{s-1}{x}\right)\left(1 - \frac{s-1}{x}\right) \leq \frac{R(x,s)}{x^s} \leq 1 + \frac{s-1}{x} \leq 1 + \frac{1}{x}, \qquad \forall x \geq 1.$$

For our purpose, we could deduce the following bounds: for any $0 \leq s \leq 2$,

$$1 - \frac{1}{x} \leq \frac{R(x,s)}{x^s} \leq 1 + \frac{1}{x}, \qquad \forall x \geq 1,$$

and

$$1 - \frac{1}{x-1} \leq \frac{x^s}{R(x,s)} \leq 1 + \frac{1}{x-1}, \qquad \forall x \geq 2.$$

Finally, using again the recurrence formula for the gamma function, when $2 \leq s \leq 3$, we can write

$$\frac{R(x,s)}{x^s} = \left(1 + \frac{s-1}{x}\right)\left(1 + \frac{s-2}{x}\right) \frac{R(x,s-2)}{x^{s-2}}$$

and deduce, for $2 \leq s \leq 3$, and $x \geq 2$,

$$\frac{R(x,s)}{x^s} \leq \left(1 + \frac{s-1}{x}\right)\left(1 + \frac{s-2}{x}\right) = 1 + \frac{3}{x} + \frac{2}{x^2} \leq 1 + \frac{4}{x}, \quad \forall x \geq 2,$$

and

$$\frac{x^s}{R(x,s)} \geq 1 - \frac{3x+2}{(x+2)(x+1)} \geq 1 - \frac{3}{x+2} \geq 1 - \frac{3}{x}, \quad \forall x \geq 1.$$

On the other hand,

$$\frac{R(x,s)}{x^s} \geq \left(1+\frac{1}{x}\right)\frac{R(x,s-2)}{x^{s-2}} \geq \left(1+\frac{1}{x}\right)\left(1-\frac{s-2}{x}\right) \geq 1-\frac{1}{x}$$

and

$$\frac{x^s}{R(x,s)} \leq \frac{x}{x+1}\frac{x^{s-2}}{R(x,s-2)} \leq \frac{x}{x+1}\,\frac{x}{x-(s-2)} \leq \frac{x^2}{x^2-1} \leq 1+\frac{1}{x-1}.$$

Gathering facts, for $0 \leq s \leq 3$

$$\left|\frac{R(x,s)}{x^s}-1\right| \leq \frac{4}{x}, \qquad \forall x \geq 2, \tag{2.49}$$

and

$$\left|\frac{x^s}{R(x,s)}-1\right| \leq \frac{3}{x-1}, \qquad \forall x \geq 2. \tag{2.50}$$

2.E Additional simulation results

2.E.1 The settings for X

In our simulations, we use three types of stochastic processes to generate the trajectories of X that we recall in the following.

- **Setting 1**: *Fractional Brownian motion.* The curves are generated using a classical fractional Brownian motion with constant Hurst parameter $H \in (0,1)$. In this case, the local regularity of the process is the same at every point. Figure 2.12a illustrates one realization of this setting.

- **Setting 2**: *Piecewise fractional Brownian motion.* The curves are generated as a concatenation of multiple fractional Brownian motions with different regularities, that is with different Hurst parameters for different time periods. In this case, the local regularity is no longer constant. Figure 2.12b illustrates one realization of this setting.

- **Setting 3**: *Integrated fractional Brownian motion.* The curves X_t are obtained as integrals $\int_0^t W_H(s)ds$, $t \in [0,1]$, of the paths of a fractional Brownian motion process W_H with constant Hurst parameter H. Here, the local regularity of the process is

the same at each point but will be greater than 1, thus this setting corresponds to the case of smooth trajectories. Figure 2.12c illustrates one realization of this setting.

2.E.2 On the computation time

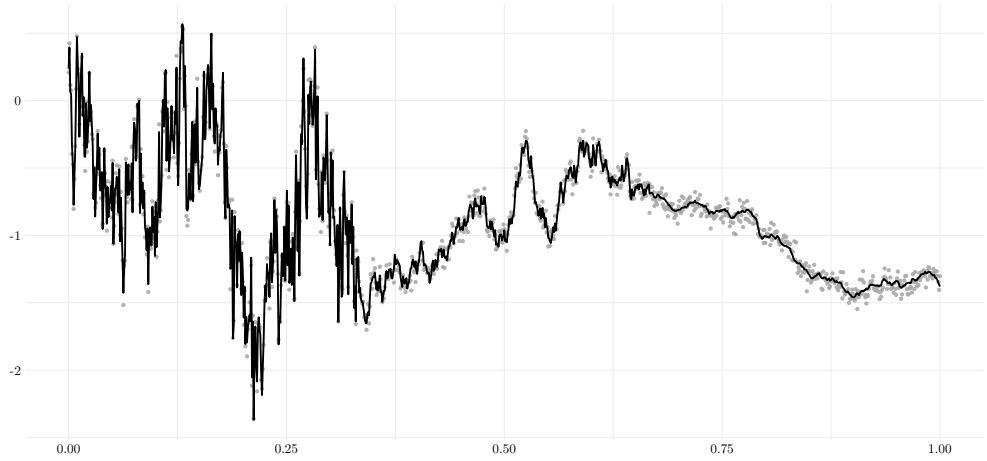
Figure 2.13a presents the violin plots of the needed time to smooth $N_1 = 1000$ curves. The results are obtained using the simulation $(1, 1000, 1000, 300, \texttt{equi}, 0.5, 0.05)$. They correspond to the total CPU time (system time and user time) to estimate the bandwidth h_n and then estimate the curves at their sampling points. We perform these computations on a personal computer equipped with a processor Intel Core i7-6600U, CPU: 2.60GHz, RAM: 24Go and rerun the estimation 10 times. We observe that our smoothing device outperforms cross-validation and *plug-in* in terms of computation time: about 1000 times faster than the cross-validation. Let $\mathcal{H}_n$ be a set of bandwiths. For the cross-validation, we may explain these differences because of the computation of the estimator for each bandwidth in $\mathcal{H}_n$ and each curve $X^{(n)}$ of the sample ($N_1 \times \text{Card}(\mathcal{H}_n)$ calls to the estimation function) while our estimator requires only one estimation of the regularity of the functions and one evaluation of the estimator per curve (N_1 calls to the estimation function). In a similar way, figure 2.13b presents the violin plots of the time necessary to smooth $N_1 = 1000$ curves with the parameters of the simulation $(3, 1000, 1000, 1000, \texttt{equi}, 1.7, 0.005)$. The same personal computer is used and the simulation is also run 10 times. For setting 3, our procedure is slower than for setting 1, which can be easily explained by the computation of the derivatives of each curve $X^{(n)}$. However, the computation time for the cross-validation is still not comparable with ours.

2.E.3 On the estimation of the local regularity

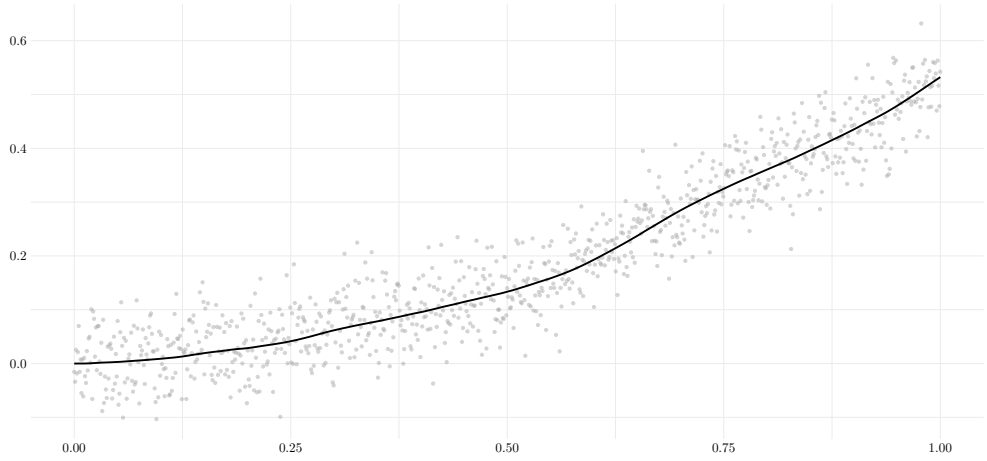
Figure 2.14 presents the results for the local regularity estimation for fBm with homoscedastic noise. The local estimation of H_{t_0} is performed at $t_0 = 1/2$ which correspond to the middle of the interval. The true value of H_{t_0} is 0.5. The results show an accurate estimator $\widehat{H}_{t_0}$, except, maybe, for the simulation $(1, 250, 500, 1000, \texttt{equi}, 0.5, 0.05)$ where there is not enough curves compared to the number of sampling points.

Figure 2.15 presents the results for the local regularity estimation for piecewise fBm with heteroscedastic noise. The local estimations of H_{t_0} are performed at $t_0 = 1/6, 1/2$ and $5/6$ which correspond to the middle of the interval for each regularity. The true values

(a) Brownian motion

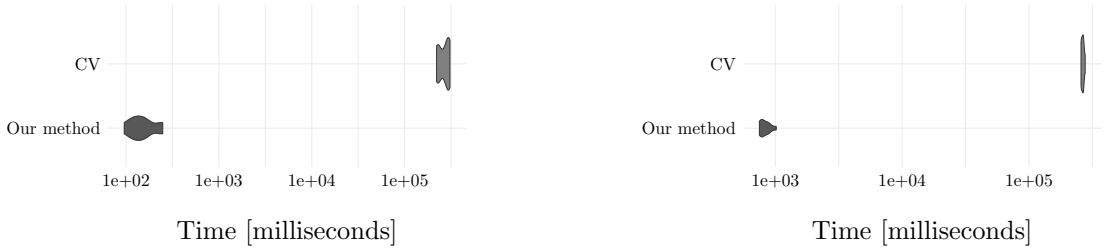


(b) Piecewise Brownian motion



(c) Integrated Brownian motion

Figure 2.12: Illustrations of simulated data generated according to the different settings. The curves correspond to the generated trajectories without noise that we aim to recover, and the grey points correspond to the noisy measurements.



(a) For setting 1

(b) For setting 3

Figure 2.13: Computational times (log scale)

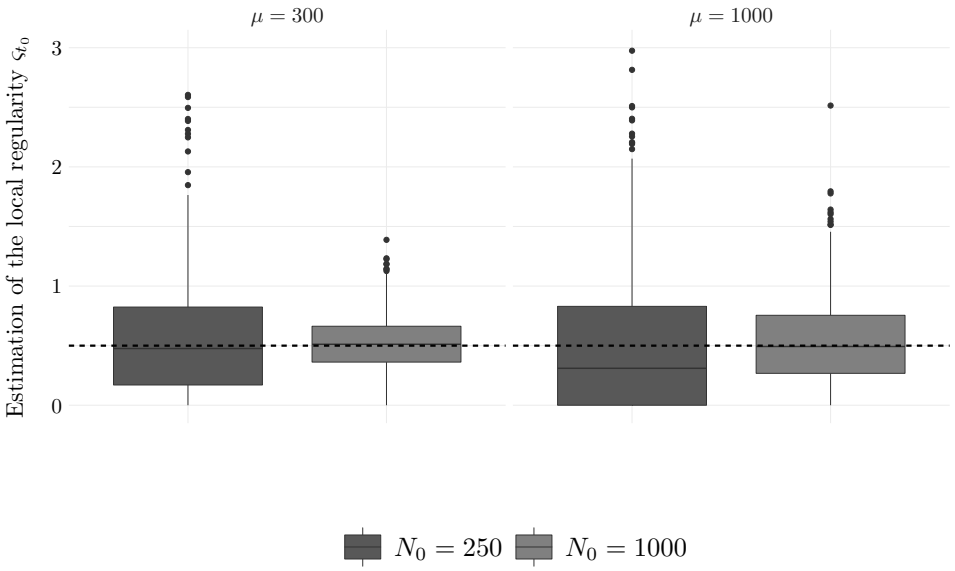


Figure 2.14: Estimation of the local regularity for fBm, with constant noise variance $\sigma^2 = 0.05$, at $t_0 = 1/2$. True value: $\varsigma_{t_0} = 0.5$.

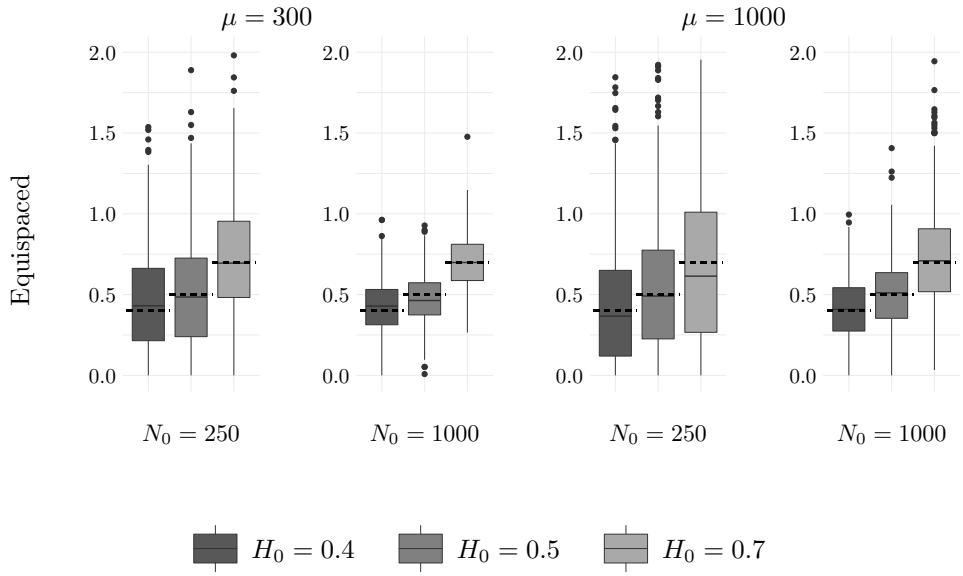


Figure 2.15: Estimation of the local regularity for piecewise fBm, with non-constant noise variance $\sigma^2 = 0.04, 0.05$ and 0.07, at $t_0 = 1/6, 1/2$ and $5/6$, respectively. True values: $\varsigma_{t_0} = H_{t_0}$ equal to $0.4, 0.5$ and 0.7, respectively.

of H_{t_0} are $0.4, 0.5$ and 0.7, respectively. The true values of σ^2 are $0.04, 0.05$ and 0.07, respectively. The results show an accurate estimator $\widehat{H}_{t_0}$.

2.E.4 On the pointwise risk

For technical convenience, in our theoretical study, we only considered the case where the regularity estimator ς_{t_0} is applied with an independent sample. If one wants to smooth the curves in the learning set, one can use a leave-one-out method. That is, for each curve, one can estimate the local regularity without that curve, and smooth the curve with the estimate obtained. Our method for calculating $\widehat{H}_{t_0}$ is very fast, and such a leave-one-curve-out procedure is feasible. This idea was used to analyze the NGSIM data. However, one could also simply smooth the *learning* set curves using the same local regularity estimates obtained from this dataset.

Figure 2.16 presents the estimation of the risks $\mathcal{R}(\widehat{X}; 1/6)$, $\mathcal{R}(\widehat{X}; 0.5)$ and $\mathcal{R}(\widehat{X}; 5/6)$ for piecewise fBm, with constant noise variance $\sigma^2 = 0.05$, when the training and the test set are the same. The simulation results indicate that our theoretical results could be extended to the case where the *online* set is taken equal to the *learning* set, though the concentration deteriorates. The theoretical investigation of this issue is left for future

work.

Figure 2.17 presents the estimation of the risks $\mathcal{R}(\widehat{X}; 0.5)$ for fBm, with constant noise variance $\sigma^2 = 0.05$.

Figure 2.18 presents the estimation of the risks $\mathcal{R}(\widehat{X}; 1/6)$, $\mathcal{R}(\widehat{X}; 0.5)$ and $\mathcal{R}(\widehat{X}; 5/6)$ for piecewise fBm, with heteroscedastic noise. The conclusion are the same than the homoscedastic case.

2.F Traffic flow: Montanino and Punzo [99] methodology

Montanino and Punzo [99] presents a four steps methodology to make the NGSIM data usable. For a complete description of the steps, we let the reader refer to their article [99]. We briefly summarize their method here. The four steps below are applied for each trajectory separately.

Step 1. *Removing the outliers*

They remove the measurements that lead to unreliable values of the acceleration by cutting all the records above a deterministic threshold of 30 m/s^2. The missing points are interpolated using a natural cubic spline with 10 reference points before and after the outliers.

Step 2. *Cutting off the high- and medium-frequency responses in the speed profile*

They remove the noise from the signal by linear smoothing of the signal with low-pass filter. The considered one is a first-order Butterworth filter [16] with cutoff frequency of 1.25 Hz.

Step 3. *Removing the residual unphysical acceleration values, keeping the consistency requirements*

They remove residual peaks that exceed defined thresholds (varying with speed levels). For that, they move the position of the vehicle when the peak in acceleration appears in order to fulfill the thresholds. In order to prevent inconsistency, a 5th-degree polynomial interpolation with constraint on the space traveled plus minor conditions was applied on a 1s window around the peak points.

Step 4. Cutting off the high- and medium-frequency reponses generated from step 3

This step is the same as the step 2 but using the results of the step 3.

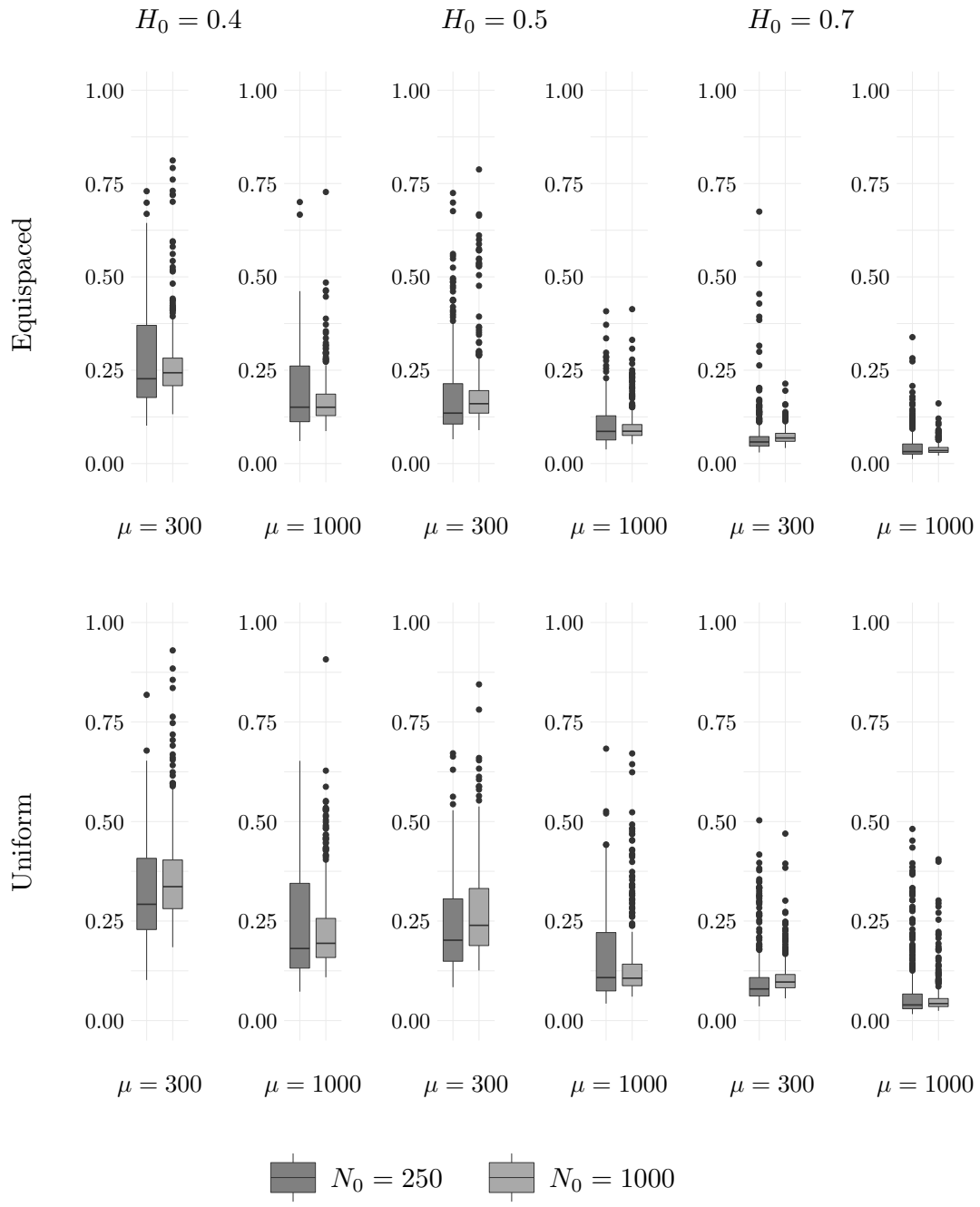


Figure 2.16: Estimation of the risks $\mathcal{R}(\widehat{X}; 1/6)$, $\mathcal{R}(\widehat{X}; 0.5)$ and $\mathcal{R}(\widehat{X}; 5/6)$ for piecewise fBm, with constant noise variance $\sigma^2 = 0.05$, when the training and the test set are the same.

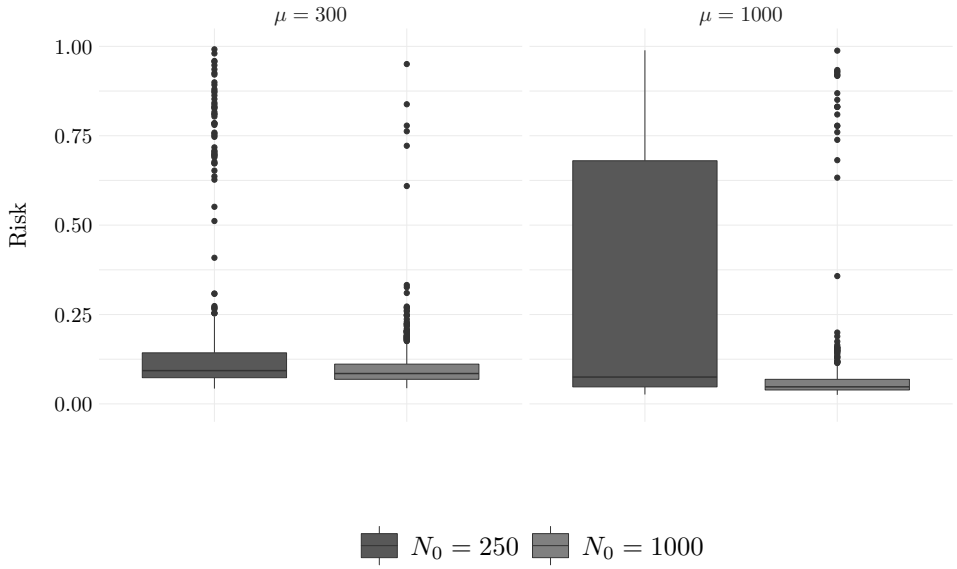


Figure 2.17: Estimation of the risk $\mathcal{R}(\widehat{X}; 0.5)$ for smoothing the noisy trajectories of a fBm, with constant noise variance $\sigma^2 = 0.05$.

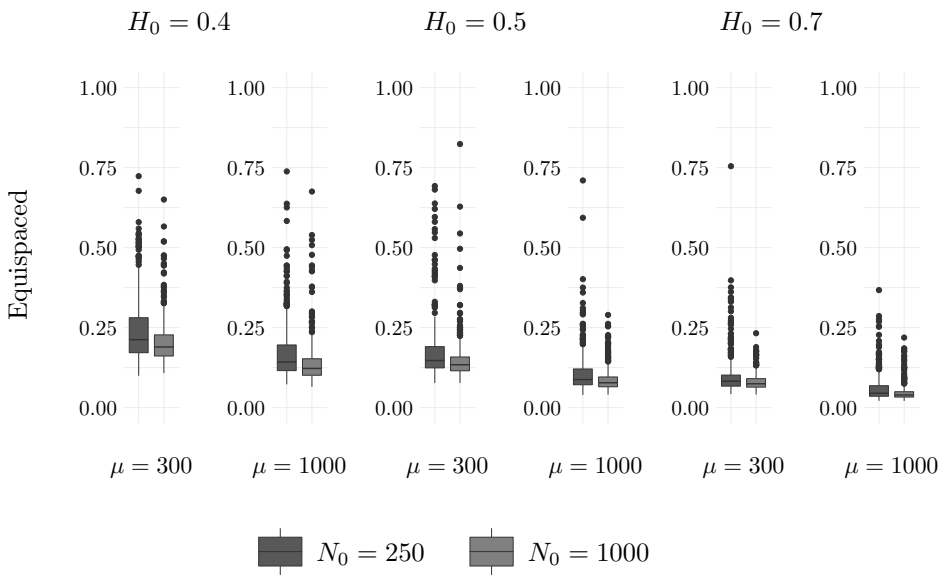


Figure 2.18: Estimation of the risks $\mathcal{R}(\widehat{X}; 1/6)$, $\mathcal{R}(\widehat{X}; 0.5)$ and $\mathcal{R}(\widehat{X}; 5/6)$ for piecewise fBm, with non-constant noise variance $\sigma^2 = 0.04, 0.05$ and 0.07.

The methodology of [99] seems very specific to the NGSIM dataset, or at least some trajectory dataset, and by extension can not be easily applied to others. For using the algorithm on other trajectory datasets, their method requires some fine-tuning of the parameters.

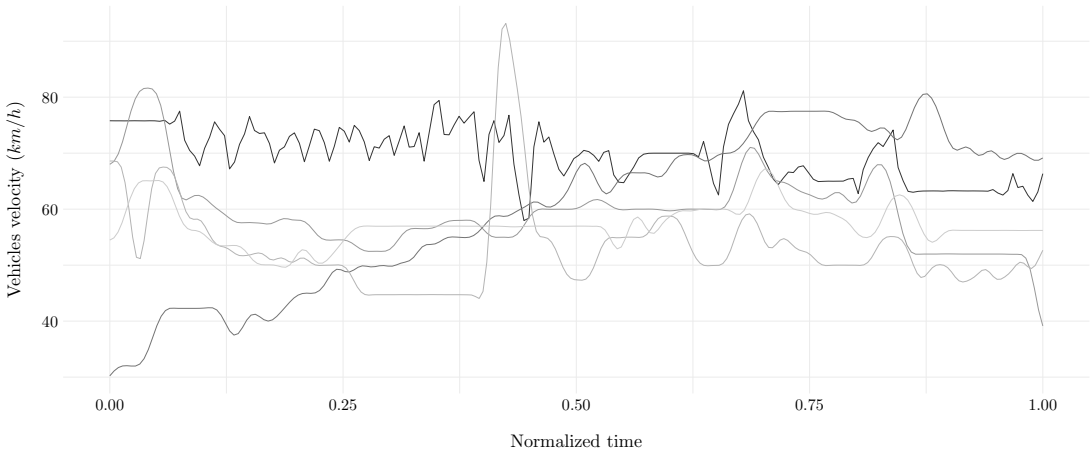
As explained in the main text, the 1714 observation units from the I-80 dataset, available in the NGSIM study, have been recorded at moments of the day when traffic is evolving, it goes from fluid to dense traffic. Therefore, we consider that there are three groups in the data: a first group corresponding to a fluid (high-speed) traffic, a second one for in-between fluid and dense traffic, and a third groups corresponding to the dense (low-speed) traffic. Our local regularity approach, and the kernel smoothing induced, are applied for each group separately. The three group clustering was performed using a Gaussian mixture model estimated by an EM algorithm initialized by hierarchical model-based agglomerative clustering as proposed by Fraley and Raftery [49] and implemented in the **R** package `mclust` [122]. The optimal model is then selected according to BIC. The three resulting classes have 239, 869 and 606 velocity trajectories, respectively. Plots of randomly selected subsamples of trajectories from each groups are provided in Figure 2.19.

2.G Complements on the real-data applications

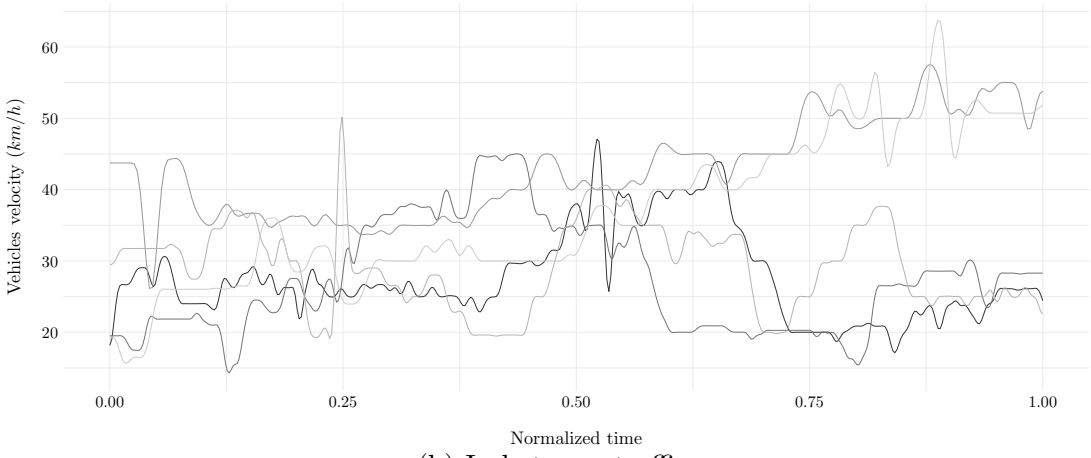
In this section, we point out the fact that our situation is not specific only to the traffic flow data, but can be applied to other real datasets.

2.G.1 Canadian weather

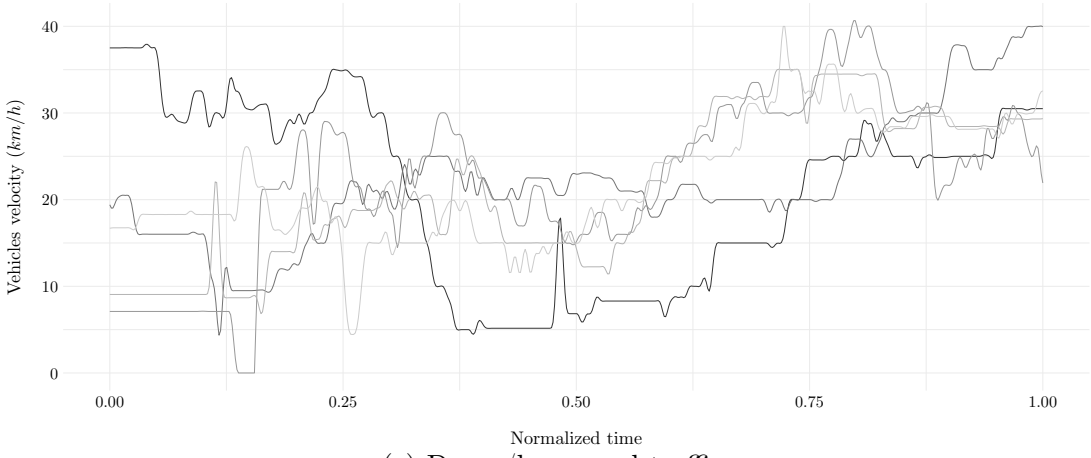
The Canadian Weather dataset [110, 108] records the daily temperature and precipitations in Canada averaged over the period from 1960 to 1994. Here, we are interested in the average daily temperature for each day of the year. It contains the measurements of 35 canadian stations. Here, we have $N_0 = 35$ and $\mu = 365$. A sample of five temperature curves has been plotted in the Figure 2.20a. Figure 2.20b presents the estimation of H_{t_0} for different t_0. We see that the estimation varies around 1 with $\widehat{K_0} = 25$.



(a) Fluid/high-speed traffic

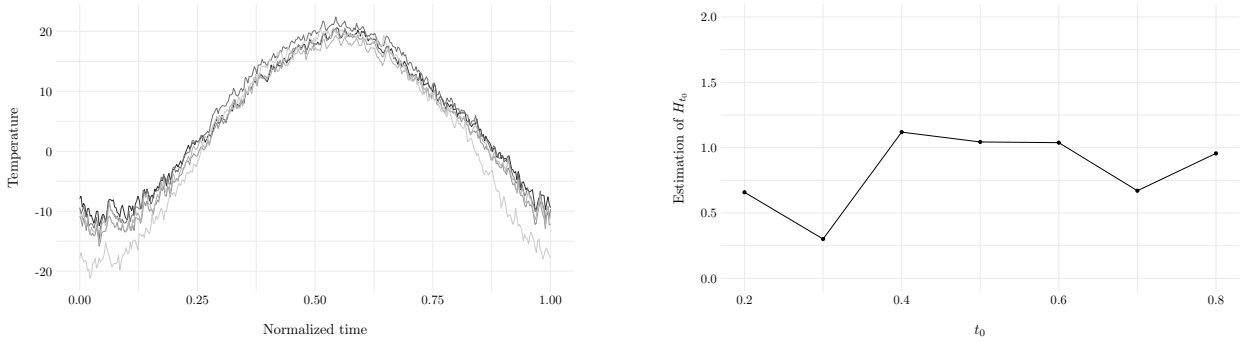


(b) In-between traffic



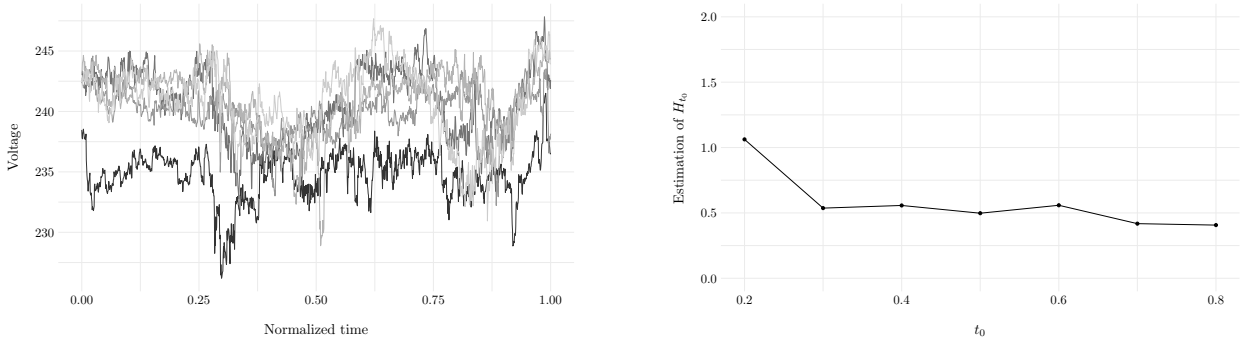
(c) Dense/low-speed traffic

Figure 2.19: I-80 dataset illustration of the clusters: a sample of five velocity curves from each of the three groups of curves



(a) A sample of five temperature curves. (b) Estimation of H_{t_0}

Figure 2.20: Canadian weather dataset illustration.



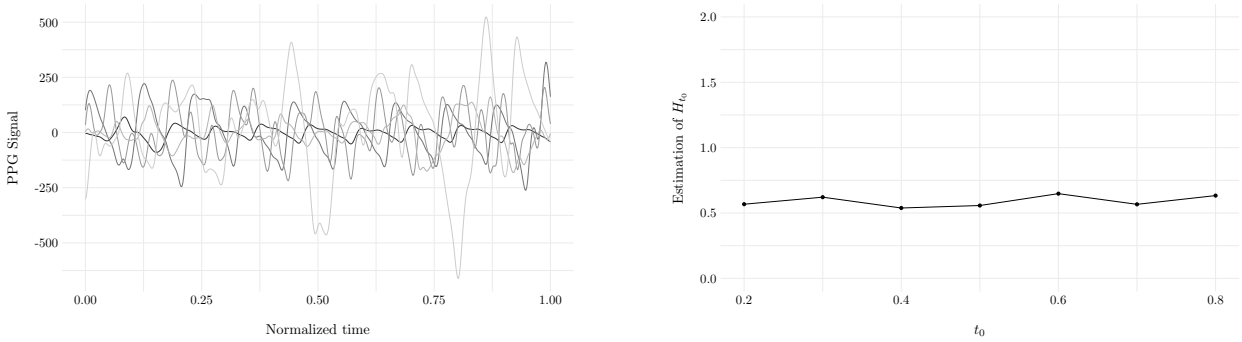
(a) A sample of five power curves. (b) Estimation of H_{t_0}

Figure 2.21: Household active power consumption dataset illustration.

2.G.2 Household Active Power Consumption

The Household Active Power Consumption dataset is part of the Monash University, UEA, UCR time series regression archive [128] and was sourced from the UCI repository[1]. The data measures diverse energy related features of a house located in Sceaux, near Paris every minute between December 2006 and November 2010. In total, its represents around 2 million data points. These data are used to predict the daily power consumption of a house. Here, we are only interested in the daily voltage. The dataset contains $N_0 = 746$ time series of $\mu = 1440$ measurements. Figure 2.21a presents a sample of five curves from this dataset. The estimation of the local regularity H_{t_0}, plotted in Figure 2.21b, is around 0.5 with $\widehat{K_0} = 73$.

1. http://bit.ly/electric_power



(a) A sample of five PPG curves.

(b) Estimation of H_{t_0}

Figure 2.22: PPG-Dalia dataset illustration.

2.G.3 PPG-Dalia

The PPG-Dalia dataset is also part of the Monash University, UEA, UCR time series regression archive [128] and was also sourced from the UCI repository[2]. PPG sensors are widely used in smart wearable devices to measure heart rate [115]. They contain a single channel PPG and 3D accelerometer motion data recorded from 15 subjects performing various real-life activities. Measurements from each subject are segmented into 8 second windows with 6 second overlaps, resulting in $N_0 = 65000$ time series of $\mu = 512$ features. Here, we are interested in the PPG channel. A sample of five curves is plotted in Figure 2.22a. The estimation of the local regularity H_{t_0} is also around 0.5 (see Figure 2.22b) with $\widehat{K}_0 = 25$.

2. http://bit.ly/ppg_dalia

Chapter 3

Adaptive estimation of irregular mean and covariance function

Abstract: *We propose straightforward nonparametric estimators for the mean and the covariance functions of functional data. Our setup covers a wide panel of practical situations. The random trajectories are not necessarily differentiable, have unknown regularity, and are measured with error at discrete design points. The measurement error could be heteroscedastic. The design points could be either randomly drawn or common for all curves. The definition of our nonparametric estimators depends on the local regularity of the stochastic process generating the functional data. We first propose a simple estimator of this local regularity which takes strength from the replication and regularization features of functional data. Next, we use the "smoooothing first, then estimate" approach for the mean and the covariance functions. The new nonparametric estimators do not involve numerical optimization, are easy to calculate and to update, and perform well in simulations. They have optimal rates of convergence in the minimax sense, with both sparsely or densely sampled curves.*

Contents

3.1 Introduction 106

3.2 From unfeasible to feasible optimal estimators 111

3.3 Local regularity estimator 116

3.3.1 Local regularity in quadratic mean 116

3.3.2 The local regularity estimation method 117

3.3.3 Concentration properties of the local regularity estimator . . . 119

3.3.4 From process regularity to trajectories regularity 120

3.4 Optimal mean and covariance estimators 121
3.4.1 Adaptive mean estimation 123
3.4.2 Adaptive covariance function estimates 126
3.4.3 Estimator on the diagonal band of the covariance function . . . 128
3.5 Empirical study 129
3.5.1 Implementation aspects 129
3.5.2 Simulation design 130
3.5.3 Mean estimation 131
3.5.4 Covariance estimation 132
3.6 Discussion and conclusions 134
Appendix 137
3.A Details on the definition (3.21) 137
3.B Proofs 139
3.C Additional simulation results 143
3.C.1 Description of the real data set used to build the simulations . 143
3.C.2 Construction of the simulation design 143

3.1 Introduction

Motivated by a large number of applications, there is a great interest in models for observation entities in the form of a sequence of measurements recorded intermittently at several discrete points in time. FDA considers such data as being values on the trajectories of a stochastic process, recorded with some error, at discrete random times. The mean and the covariances functions play a critical role in FDA models.

To formalize the framework, consider $\mathcal{T} \subset \mathbb{R}$ to be a non degenerate compact interval. Data consist of a random sample of sample paths from a second-order stochastic process $X = (X_t : t \in \mathcal{T})$ with continuous trajectories. The mean and covariance functions are $\mu(t) = \mathbb{E}(X_t)$ and

$$\Gamma(s,t) = \mathbb{E}\left\{[X_s - \mu(s)][X_t - \mu(t)]\right\} = \mathbb{E}\left(X_s X_t\right) - \mu(s)\mu(t), \qquad s,t \in \mathcal{T},$$

respectively. If the independent realizations $X^{(1)}, \ldots, X^{(n)}, \ldots, X^{(N)}$ of X were observed, the natural estimators would be

$$\widetilde{\mu}_N(t) = \frac{1}{N}\sum_{n=1}^{N} X_t^{(n)}, \qquad t \in \mathcal{T},$$

and

$$\widetilde{\Gamma}_N(s,t) = \frac{1}{N-1}\sum_{n=1}^{N}\{X_s^{(n)} - \widetilde{\mu}_N(s)\}\{X_t^{(n)} - \widetilde{\mu}_N(t)\}, \qquad s,t \in \mathcal{T}.$$

In real applications, the curve are rarely observed without error and never at each value $t \in \mathcal{T}$. That is why we consider the following common and more realistic setup. For each $1 \leq n \leq N$, and given a positive integer M_i, let $T_m^{(n)} \in \mathcal{T}$, $1 \leq m \leq M_n$, be the observation times for the curve $X^{(n)}$. The observations associated to a curve, or trajectory, $X^{(n)}$ consist of the pairs $(Y_m^{(n)}, T_m^{(n)}) \in \mathbb{R} \times \mathcal{T}$ where $Y_m^{(n)}$ is defined as

$$Y_m^{(n)} = X^{(n)}(T_m^{(n)}) + \varepsilon_m^{(n)}, \qquad 1 \leq n \leq N, \;\; 1 \leq m \leq M_n, \tag{3.1}$$

and $\varepsilon_m^{(n)}$ is an independent (centered) error variable. Here and in the following, we use the notation $X_t^{(n)}$ for the value at a generic point $t \in \mathcal{T}$ of the realization $X^{(n)}$ of X, while $X^{(n)}(T_m^{(n)})$ denotes the measurement at $T_m^{(n)}$ of this realization.

A commonly used idea is to build feasible versions of $\widetilde{\mu}_N(\cdot)$ and $\widetilde{\Gamma}_N(\cdot,\cdot)$ using non-parametric estimates of $X_t^{(n)}$ and $X_s^{(n)}X_t^{(n)}$, such as obtained by smoothing splines or local polynomials. This approach, usually called "smoothing first, then estimate" or "two-stage procedure", have been considered, among others, in [63, 151]. In general, the sample trajectories are required to admit at least second order derivatives on the whole domain $\mathcal{T}$. Li and Hsing [91] and Zhang and Wang [152, 153] propose an alternative local linear smoothing approach where the estimators are determined by suitably weighting schemes which involve the whole sample of curves. This idea exploits the so-called replication and regularization features of functional data (see [108, ch. 22]). In this alternative approach, the regularity assumptions are imposed on the mean and covariance functions, which are required to admit second, or higher, order derivatives on the whole domain. Since, in general, the mean and covariance functions are more regular than the sample trajectories, the approach based on weighting schemes using all the sample curves seems preferable. However, in some applications, for instance in signal processing, the mean and covariance functions could be quite irregular, and in general of unknown irregularity.

Cai and Yuan [19, 18] derive the optimal rates of convergence, in the minimax sense,

for the mean and covariance functions, respectively, and propose optimal estimators. The optimal estimator of the mean function proposed in [19] is a smoothing spline estimator which could be built only if the regularity of the sample paths is known. For the covariance function, Cai and Yuan [18] used the representation of the covariance function in a tensor product reproducing kernel Hilbert space (RKHS) space. Next, they derived estimators for $\Gamma(s,t)$ by a low dimension version of this representation obtained by a regularization procedure, provided the values M_n are not very different. This procedure does not require a given regularity, but involve some numerical optimization. See also [141]. The minimax optimal rates for the mean and covariance functions are defined by the sum of two type of terms. One corresponds to the rate of convergence of the $\widetilde{\mu}_N(\cdot)$ and $\widetilde{\Gamma}_N(\cdot,\cdot)$, which is the standard rate of convergence for empirical means and covariances. The other contribution to the optimal rates is given by the differences between $\widetilde{\mu}_N(\cdot)$ and $\widetilde{\Gamma}_N(\cdot,\cdot)$ and their feasible versions. The rates of the differences depend on the sample trajectories regularity. The reason is that the minimax lower bounds should also take into account the case where the trajectories have the same regularity as the function to be estimated.

The estimation of the mean and covariance functions presents another specific feature. The minimax optimal rates of convergence depend on the nature of the measurement times $T_m^{(n)}$. For now, two situations were investigated in the literature. On the one hand, the so-called *independent design* case where, given the M_n's, the $T_m^{(n)}$ are obtained as random sample of size $M_1 + \cdots + M_N$ from same continuous distribution. On the other hand, the so-called *common design* case where the M_n are all equal to some integer value $\mathfrak{m}$, and $T_m^{(n)}$, $1 \leq m \leq \mathfrak{m}$, are the same across the curves $X^{(n)}$. In both cases, the best rates for the nonparametric estimators depend on the regularity of the sample trajectories. These rates also depend on the number of different observation times $T_m^{(n)}$, that is equal to $M_1 + \cdots + M_N$ with independent design, and equal to $\mathfrak{m}$ with common design. In other words, the replication feature of functional data is less impactful with common design. See [19] for the case of the mean function, and [18, 20] for the covariance function case.

Here we propose a simple "smoothing first, then estimate" type method, based on $1-$dimensional smoothing, which achieves minimax optimal rates of convergence. The process is allowed to have a piecewise constant, unknown regularity. Our method does not involve any numerical optimization, such is required for the regularization based approaches. It can be applied in both common and independent design situations, and allows for general heteroscedastic measurement errors $\varepsilon_m^{(n)}$. Our approach is suitable with both sparsely or densely sampled curves. The definition of sparse and dense regimes is

recalled in Section 3.2.

Let $\widehat{X}^{(n)}$ a suitable nonparametric estimator of the curve $X^{(n)}$ applied with the M_n pairs $(Y_m^{(n)}, T_m^{(n)})$, for instance a local polynomial estimator. What will make this estimator suitable is that it takes into account the regularity of the process X and the final estimation purpose, that is the mean or the covariance function. Moreover, for each n, the smoothing parameter is allowed to depend on M_n. These features can be achieved in an easy, data-driven way, as it will be explained below. For now, with at hand the $\widehat{X}^{(n)}$'s tuned for the mean function estimation, we define

$$\widehat{\mu}_N(t) = \frac{1}{N}\sum_{n=1}^{N} \widehat{X}_t^{(n)}, \qquad t \in \mathcal{T}. \tag{3.2}$$

For the covariance function, we distinguish the diagonal from the non-diagonal points. For now, with at hand the $\widehat{X}^{(n)}$'s tuned for the covariance function estimation, and for some $\delta \geq 0$ determined using the data, let us define

$$\widehat{\Gamma}_N(s,t) = \frac{1}{N-1}\sum_{n=1}^{N}\{\widehat{X}_s^{(n)} - \widehat{\mu}_N(s)\}\{\widehat{X}_t^{(n)} - \widehat{\mu}_N(t)\}, \qquad s,t \in \mathcal{T}, \ \ |s-t| > \delta. \tag{3.3}$$

Moreover,

$$\widehat{\Gamma}_N(s,t) = \frac{1}{N}\sum_{i=1}^{N} \widehat{(X^{(n)})_u^2} - \widehat{\mu}_N^2(u), \quad \text{with} \ \ u = (s+t)/2, \ \ s,t \in \mathcal{T}, \ \ |s-t| \leq \delta, \tag{3.4}$$

where, for each $u \in \mathcal{T}$, $\widehat{(X^{(n)})_u^2}$ is a suitable nonparametric estimator of the squared of the value of the curve $X^{(i)}$ at the point u. It is well-known that the variance function $\Gamma(s,s)$ induces a singularity when estimating the covariance function $\Gamma(\cdot,\cdot)$. See, for instance, [152, Remark 4]. This singularity causes little problem when studying pointwise rates of convergence and confidence intervals. In that cases, one could set $\delta = 0$ and use a modified covariance function estimator only for the diagonal $\widehat{\Gamma}_N(s,s)$. However, the diagonal singularity could deteriorate the rate of converge of $\widehat{\Gamma}_N$ in integrated squared norm. In general, this rate deterioration is propagated and affects the rates of convergence for the estimators of the eigenvalues and eigenfunctions of the covariance operator defined by Γ. The idea underlying (3.4) is simple and the diagonal tuning parameter δ can decrease to zero according to a data-driven rule.

Although the methodology we propose in the following is quite general and can be

used with different types of smoothers, we will focus on the case where the $\widehat{X}_t^{(n)}$ and $\widehat{(X^{(n)})_u^2}$ are obtained by local polynomials with a compactly supported kernel. In this case tuning the $\widehat{X}^{(n)}$'s means to suitably choose the rate of decrease and the constant defining the bandwidth. Moreover, when the kernel is supported on $[-1, 1]$, we could take δ equal to the bandwidth. The nonparametric estimator of the squared values of the curve $X^{(n)}$ is built smoothing the squares of the $Y_m^{(n)}$, $1 \leq m \leq M_n$, with the same bandwidth as used for $\widehat{X}^{(n)}$, and is corrected for the upward bias induced by the measurement errors.

To the best of our knowledge, there is no contribution following the "smoothing first, then estimate" idea, which considers estimators of the curves $X^{(n)}$ adapted to their regularity and to the purpose of estimating mean and covariance functions. It is clear that trajectory-by-trajectory adaptive optimal smoothing, for instance using the Goldenshluger and Lepski [55] method, yields in general sub-optimal rates of convergence for $\widehat{\mu}_N(t)$ and $\widehat{\Gamma}_N(s,t)$. The reason is that trajectory-by-trajectory smoothing ignores the information contained in the other $N-1$ curves in the sample generated according to the same stochastic process X. See [19] for a discussion on the differences with the usual nonparametric rates. One can also use CV for choosing the bandwidth with the suitably weighting schemes, such as the one proposed in [91, 152]. However, this would require significant computational effort, and, to our best knowledge, the idea have not yet received any theoretical justification. Using the replication and regularization features of functional data, we propose a new, simple and effective estimator for the local regularity of the process X, a probabilistic concept which determines the analytic regularity of the trajectories of X. The replication feature of the functional data makes the concept of local regularity of the process a more meaningful parameter than the usual trajectory regularity, which is an analytic concept designed for a single function. Our local regularity estimator, inspired by but diferent from the estimator introduced in [59], combines information both across and within curves. Moreover, it allows for general heteroscedastic measurement errors, does not involve any optimization and is obtained after a fast, possibly parallel, computation. With at hand the local regularity estimator, we derive the optimal estimators $\widehat{X}_t^{(n)}$ and $\widehat{(X^{(n)})_u^2}$, in the sense that, for each $s, t \in \mathcal{T}$, they make the rates of the differences

$$\widehat{\mu}_N - \widetilde{\mu}_N \quad \text{and} \quad \widehat{\Gamma}_N - \widetilde{\Gamma}_N,$$

to be almost as small as possible in the minimax sense. For each curve $X^{(n)}$, the smoothing parameters used to build $\widehat{X}_t^{(n)}$ and $\widehat{(X^{(n)})_u^2}$, depend on M_i and N, but can explicitly

be computed given the estimate of the local regularity of X. Then, up to logarithmic multiplicative terms, our estimators $\widehat{\mu}_N$ and $\widehat{\Gamma}_N$ are rate optimal. It is worth noting that we achieve the optimal rates of converges without using a bivariate nonparametric estimator of $X_s^{(n)}X_t^{(n)}$. We explain this somehow unexpected result by the particular separable structure of the bivariate functions $(s,t) \mapsto X_s^{(n)}X_t^{(n)}$ which can be estimated at the faster rate of univariate functions.

In Section 3.2, we provide insight on why the local regularity of the process X is a natural feature to be taken into account. Moreover, we explain why the simple "smoothing first, then estimate" approach achieves optimal rates of convergence when the regularity of X is known. In Section 3.3, we formally define the local regularity of the process X and introduce a new estimator for this regularity. For simplicity, we propose to build it using a separate learning sample. We derive exponential bounds for the concentration of the regularity estimator under mild conditions. In particular, both independent and common designs are allowed, the process regularity is allowed to be piecewise constant, and the size of the learning sample can be much smaller than N. At the end of Section 3.3, we formally describe the relationship between the process regularity, a probabilistic concept, and the trajectories regularity, an analytic concept. In Section 3.4, we use the regularity estimate to build suitable local polynomial estimates of the trajectories, which are further used to adaptively estimate the mean function. The adaptive estimator of the covariance function is given in Section 3.4.2. The finite sample performance of the new estimators of the mean and covariance functions are illustrated in Section 3.5 using simulated and real data samples. The proofs are relegated to the Appendix.

3.2 From unfeasible to feasible optimal estimators

Let us first provide insight on the reason why the local regularity of the process generating the curves is a meaningful concept, and why our approach can achieve good performance. For this purpose, we analyze the difference $\widehat{\mu}_N(t) - \widetilde{\mu}_N(t)$, $s,t \in \mathcal{T}$, but the ideas can be easily extended to the covariance function estimation. The data $(Y_m^{(n)}, T_m^{(n)}) \in \mathbb{R} \times \mathcal{T}$ are generated according to the model (3.1) with

$$\varepsilon_m^{(n)} = \sigma(T_m^{(n)}, X^{(n)}(T_m^{(n)}))u_m^{(n)}, \quad 1 \leq m \leq M_i,\ 1 \leq i \leq N, \tag{3.5}$$

where $u_m^{(n)}$ are independent copies of a centered variable e with unit variance, and $\sigma(t,x)$ is some unknown bounded function which account for possibly heteroscedastic measurement errors. For each $1 \leq n \leq N$, the observations times $T_m^{(n)}$ are independent realizations of a random variable T taking values in $\mathcal{T}$. The integers $M_1, \ldots, M_N$ represent an independent sample of an integer-valued random variable M with expectation $\mathfrak{m}$ which increases with N. Thus $M_1, \ldots, M_N$ is the Nth line in a triangular array of integer numbers. We assume that the realizations of X, M and T are mutually independent. Let $\mathcal{T}_{obs}^{(n)}$ denote the set of observation times $T_m^{(n)}$, $1 \leq n \leq M_n$, on the trajectory $X^{(n)}$. In the common design case, $M \equiv \mathfrak{m}$, and the $\mathcal{T}_{obs}^{(n)}$ are the same for all n. Thus, if not stated differently, the issues discussed in this section apply to both independent design and common design cases.

Let

$$\mathbb{E}_n(\cdot) = \mathbb{E}(\cdot \mid M_n, \mathcal{T}_{obs}^{(n)}, X^{(n)}) \quad \text{and} \quad \mathbb{E}_{M,T}(\cdot) = \mathbb{E}\left(\cdot \mid M_n, \mathcal{T}_{obs}^{(n)}, 1 \leq n \leq N\right).$$

For any $t \in \mathcal{T}$, we consider a generic linear nonparametric estimator

$$\widehat{X}_t^{(n)} = \sum_{m=1}^{M_n} Y_m^{(n)} W_m^{(n)}(t), \quad 1 \leq n \leq N. \tag{3.6}$$

The weights $W_m^{(n)}(t)$ are defined as functions of the elements in $\mathcal{T}_{obs}^{(n)}$. An example is the local polynomial estimator which we investigate in Section 3.4. We consider the decomposition

$$\widehat{X}_t^{(n)} - X_t^{(n)} = B_t^{(n)} + V_t^{(n)}, \quad t \in \mathcal{T}, \tag{3.7}$$

where

$$B_t^{(n)} := \mathbb{E}_n\left[\widehat{X}_t^{(n)}\right] - X_t^{(n)} \quad \text{and} \quad V_t^{(n)} := \widehat{X}_t^{(n)} - \mathbb{E}_n\left[\widehat{X}_t^{(n)}\right] = \sum_{m=1}^{M_n} \varepsilon_m^{(n)} W_m^{(n)}(t).$$

Let us point out that, if the generic nonparametric estimator is constructed following exactly the same rule for each curve $X^{(n)}$, all the couples of random variables $(B_t^{(n)}, V_t^{(n)})$, $1 \leq n \leq N$, are independent and have the same distribution. We could reasonably assume that the $B_t^{(n)}$ and $V_t^{(n)}$ are squared integrable for all t.

For the mean function we can write

$$\widehat{\mu}_N(t) - \widetilde{\mu}_N(t) = \frac{1}{N}\sum_{n=1}^{N} B_t^{(n)} + \frac{1}{N}\sum_{n=1}^{N} V_t^{(n)}.$$

All the variables $\varepsilon_m^{(n)}$ are centered and conditionally independent, with bounded conditional variance, given $\mathcal{T}_{obs}^{(1)}, \ldots, \mathcal{T}_{obs}^{(N)}$. Thus, given $M_1, \ldots, M_N$ and $\mathcal{T}_{obs}^{(1)}, \ldots, \mathcal{T}_{obs}^{(N)}$, on the variance part we obtain

$$\mathbb{E}_{M,T}\left[\left\{N^{-1}\sum_{i=1}^{N} V_t^{(i)}\right\}^2\right] = N^{-1}\mathbb{E}_{M,T}\left[N^{-1}\sum_{i=1}^{N}\left\{V_t^{(i)}\right\}^2\right]$$
$$\leq N^{-1}\sup_x \sigma^2(t,x) \times N^{-1}\sum_{i=1}^{N}\left\{\max_m \left|W_m^{(i)}(t)\right| \times \sum_{m=1}^{M_i}\left|W_m^{(i)}(t)\right|\right\}. \quad (3.8)$$

For local polynomials with bandwidth h, under some mild conditions, the rate of decrease of the right-hand side in the last display, given the design, is $O_{\mathbb{P}}\left((N\mathfrak{m}h)^{-1}\right)$.

On the bias part, let us suppose for the moment that the trajectories are not differentiable. By Cauchy-Schwarz inequality, we have

$$\mathbb{E}_{M,T}\left[\left\{N^{-1}\sum_{i=1}^{N} B_t^{(i)}\right\}^2\right] \leq N^{-1}\sum_{i=1}^{N}\mathbb{E}_{M,T}\left[\left\{B_t^{(i)}\right\}^2\right]$$
$$\leq N^{-1}\sum_{i=1}^{N}\left\{\sum_{m=1}^{M_i}\left|W_m^{(i)}(t)\right| \times \sum_{m=1}^{M_i}\mathbb{E}_{M,T}\left(\left\{X^n(T_m^{(n)}) - X_t^{(i)}\right\}^2 \mid \mathcal{T}_{obs}^{(i)}\right)|W_m^{(i)}(t)|\right\}. \quad (3.9)$$

It now becomes clear that the rate of the square of the bias term in the difference $\widehat{\mu}_N(t) - \widetilde{\mu}_N(t)$ is determined by the second order moment of the increments $X_s^{(n)} - X_t^{(n)}$. If, for s and t which are close, we suppose

$$\mathbb{E}\left(\left\{X_s^{(n)} - X_t^{(n)}\right\}^2\right) \approx L_0^2|t-s|^{2H_0}, \quad (3.10)$$

with some constants $0 < H_0 \leq 1$ and L_0, then the rate of the right-hand side quantity in (3.9) can be bounded by

$$N^{-1}\sum_{i=1}^{N}\left\{\sum_{m=1}^{M_i}\left|W_m^{(i)}(t)\right| \times \sum_{m=1}^{M_i} L_0^2\left|T_m^{(n)} - t\right|^{2H_0}|W_m^{(i)}(t)|\right\}.$$

For local polynomials with bandwidth h, this leads to $O_{\mathbb{P}}\left(h^{2H_0}\right)$. We call H_0 the Holder exponent of $X^{(n)}$.

With $\delta-$times differentiable trajectories, $\delta \geq 1$ integer, the condition (3.10) should be considered with the difference of the $\delta-$th derivatives of $X^{(n)}$ values instead of the

difference of $X^{(n)}$ values itself. Then, following the lines of [135, Proposition 1.13], we use Taylor expansion and the property $\sum_{m=1}^{M_n}(T_m^{(n)}-t)^d W_m^{(n)}(t)=0$ which is satisfied by any $0\leq d\leq \delta$ and $1\leq n\leq N$. Then, the rate in (3.9) becomes $O_{\mathbb{P}}\left(h^{2(\delta+H_\delta)}\right)$, where $H_\delta\in(0,1]$ denotes the Holder exponent of the $\delta-$th derivatives of $X^{(n)}$.

Gathering facts, we deduce that, in the case of $\delta-$times differentiable trajectories, with $H_\delta\in(0,1]$, the Holder exponent of the derivatives of order δ of $X^{(n)}$, for h of order $(N\mathfrak{m})^{-1/(1+2\{\delta+H_\delta\})}$, whenever this order is possible, we obtain

$$\mathbb{E}_{M,T}\left[\{\widehat{\mu}_N(t)-\widetilde{\mu}_N(t)\}^2\right]=O_{\mathbb{P}}\left((N\mathfrak{m})^{-\frac{2(\delta+H_\delta)}{1+2(\delta+H_\delta)}}\right).$$

Thus, given the local regularity H_0, the estimator $\widehat{\mu}_N(t)$ can achieve the minimax optimal rate for the estimation of the mean function $\mu(t)$. See [19].

Let us notice that in some cases, with local polynomials, the estimator defined in (3.6) could be degenerate, that is, for few trajectories, all the weights $W_m^{(n)}(t)$ could be equal to zero. The trajectories for which this happens could change with t. Then, $\widehat{\mu}_N(t)$ is defined as an average over the trajectories for which the estimator (3.6) is not degenerate. This can more likely happen in the so-called *sparse* regime, where $\mathfrak{m}^{2(\delta+H_\delta)}\ll N$. A similar phenomenon occurs with estimators determined by suitably weighting schemes, see for instance [91, equation (2.1)], or [152, equation (2.3)]. However, in the independent case, one benefits from the full strength of the replication feature of functional data which makes that only at most a fraction of trajectories will yield degenerate estimators $\widehat{X}_t^{(n)}$, and thus the bounds in (3.8) and (3.9) remain meaningful.

The case of common design requires some special attention. For simplicity, let us assume the common design points are equidistant and consider the local polynomials are built with the kernel supported on $[-1,1]$. In this case, the bandwidth cannot have a rate smaller than $\mathfrak{m}^{-1}$, otherwise all the weights $W_m^{(n)}(t)$ could all be equal to zero. This means that in the case of common design, the optimal bandwidth is given by the minimization of $h^{2(\delta+H_\delta)}+(N\mathfrak{m}h)^{-1}$ under the constraint that $\mathfrak{m}h$ stays away from zero. Without loss of generality, we could set $h=k/\mathfrak{m}$ with k a positive integer and search k which minimizes $h^{2(\delta+H_\delta)}+(N\mathfrak{m}h)^{-1}$. Balancing the two terms, one expects the optimal $k/\mathfrak{m}$ to have the rate $(N\mathfrak{m})^{-1/\{1+2(\delta+H_\delta)\}}$. If $\mathfrak{m}^{2(\delta+H_\delta)}$ is larger than N, that is in the so-called *dense* regime, the optimal k is well defined and $k\approx(\mathfrak{m}^{2(\delta+H_\delta)}/N)^{1/\{1+2(\delta+H_\delta)\}}$ and, with this optimal choice, $\mathbb{E}_{M,T}\left[\{\widehat{\mu}_N(t)-\widetilde{\mu}_N(t)\}^2\right]=o_{\mathbb{P}}\left(N^{-1}\right)$. If $\mathfrak{m}^{2(\delta+H_\delta)}\ll N$, then the constraint that $k\geq 1$ becomes binding, and is no longer possible to balance the squared bias term and the

variance term. The rate of $h^{2(\delta+H_\delta)}$ dominates the rate $(N\mathfrak{m}h)^{-1}$. Then, the minimal rate for $\mathbb{E}_{M,T}\left[\{\widehat{\mu}_N(t)-\widetilde{\mu}_N(t)\}^2\right]$ corresponds to $k=1$, and is $O_{\mathbb{P}}\left(\mathfrak{m}^{-2(\delta+H_\delta)}\right)$. We obtain a similar conclusion with differentiable trajectories. Gathering facts, we recover the optimal rate for mean estimation with common design, that is $O_{\mathbb{P}}\left(\mathfrak{m}^{-2(\delta+H_\delta)}+N^{-1}\right)$, as found in [19]. Finally, let us recall the somehow surprising message from Cai and Yuan [19, p. 2332]: the interpolation is rate optimal when $\mathfrak{m}^{2(\delta+H_\delta)} \ll N$ in the case of common design, smoothing does not improve converges rates. Our contribution on this aspect is a data-driven rule for the practitioner which completes this theoretical fact about the interpolation. The adaptive bandwidth rule proposed in Section 3.4 automatically chooses between smoothing and interpolation.

We learn from above that the "smoothing first, then estimate" approach leads to optimal rates of convergence for estimating the mean function with independent and common design, as derived in [19], provided the regularity of the process X is known. We will show that achieving optimal rates using the local regularity is also possible for the covariance function. This probabilistic concept of regularity, summarized by (3.10), will be formally introduced in the next section. Next, we introduce a simple estimator of the local regularity of X. It will be shown that, under some mild conditions, this estimator concentrates around the true local regularity faster than a suitable negative power of $\log(\mathfrak{m})$. This suffices to show that our feasible estimators $\widehat{\mu}_N(t)$ and $\widehat{\Gamma}_N(s,t)$ achieve the same rates as the local regularity was known. Let us point out that the local regularity estimate also provides the practitioner with a procedure which automatically adapts to the so-called sparse and dense curve sampling regimes. Despite their importance, these situations cannot be distinguished in practice without the knowledge of the regularity of the trajectories.

Let us end this section with a discussion of the differences with the approach based on weighting scheme, as for instance considered in [91, 152]. If one knows the regularity of $\mu(\cdot)$, then one could define $B_t^{(n)}$ and $V_t^{(n)}$ in (3.7) centering by the mean function instead of the trajectory $X_t^{(n)}$. Then, one could derive the rate of $\mathbb{E}\left[\{\widehat{\mu}_N(t)-\mu(t)\}^2\right]$ and find the bandwidth which minimizes this rate, exactly as done in [91, 152] where $\mu(\cdot)$ is assumed to be twice continuously differentiable. However, the estimation of the regularity of $\mu(\cdot)$ remains an open problem.

3.3 Local regularity estimator

We will allow the sample paths of the real-valued random process $X = (X_t : t \in \mathcal{T})$ to have piecewise constant regularity. That means that there exists a finite, but unknown, partition of $\mathcal{T}$ in non degenerate, compact intervals, such that, in the interior of each interval of the partition, the regularity of the process is the same. To allow this generality, we first need to formally introduce the notion of *local regularity* of the process X. Given this type of regularity, the Kolmogorov Extension Theorem allows to determine the analytic regularity of the trajectories of X. The details are provided in Section 3.3.4.

The most convenient way to obtain the theoretical properties of our "smoothing first, then estimate" approach for the mean and covariance functions, will be to consider that the estimator of the local regularity of X is built from a separate, independent sample of trajectories. We will call the data obtained from this separate random sample of trajectories, generated by the same stochastic process X, the *learning sample*. We will show that, in theory, the size of the learning sample can be as small as any positive power of $\mathfrak{m}$, regardless of the type of design, independent or common. Our simulation experiences reveal that in practice one can confidently use the same sample for learning the local regularity of X and for estimating the mean and covariance functions.

3.3.1 Local regularity in quadratic mean

Let $\mathcal{O}_*$ be an open subinterval of $\mathcal{T}$, with length $\Delta_* = \operatorname{diam}(\mathcal{O}_*)$. Whenever it exists, let $\nabla^d X_t$ denote the $d-$th derivative, $d \geq 1$, of the generic curve X_t at the point $t \in \mathcal{T}$. By definition $\nabla^0 X_t \equiv X_t$. The following assumptions ensure that, locally on $\mathcal{O}_*$, the derivatives of the process X exist up to the integer order $\delta \geq 0$ and that the δ-th derivative is regular in quadratic mean.

Assumptions. Let $\delta \in \mathbb{N}$ and $0 < H_\delta \leq 1$.

(H1) With probability 1, for any $d \in \{0, \ldots, \delta\}$ the d-th order derivative $\nabla^d X_t$ of X_t exists for all $t \in \mathcal{O}_*$, and satisfies:

$$0 < \underline{a}_d = \inf_{u \in \mathcal{O}_*} \mathbb{E}\left[(\nabla^d X_u)^2\right] \leq \sup_{u \in \mathcal{O}_*} \mathbb{E}\left[(\nabla^d X_u)^2\right] = \overline{a}_d < \infty.$$

(H2) There exist two positive constants M_δ and β_δ such that:

$$\left|\mathbb{E}\left[(\nabla^\delta X_t - \nabla^\delta X_s)^2\right] - L_\delta^2 |t-s|^{2H_\delta}\right| \leq M_\delta^2 |t-s|^{2H_\delta} \Delta_*^{2\beta_\delta}, \qquad s, t \in \mathcal{O}_*.$$

(H3) There exists $\mathfrak{a} > 0$ and $\mathfrak{A} > 0$ such that, for any $d \in \{0, \dots, \delta\}$ and any $p \geq 1$:

$$\mathbb{E}\left[|\nabla^d X_t - \nabla^d X_s|^{2p}\right] \leq \frac{p!}{2}\mathfrak{a}\mathfrak{A}^{p-2}, \qquad s, t \in \mathcal{O}_*.$$

Definition 1. *For any $\delta \in \mathbb{N}$, $0 < H_\delta \leq 1$ and $L_\delta > 0$, the class $\mathcal{X}(\delta + H_\delta, L_\delta; \mathcal{O}_*)$ is the set of stochastic processes indexed by $t \in \mathcal{O}_*$ for which the conditions (H1) to (H3) hold true. The quantity $\alpha = \delta + H_\delta$ is the local regularity of the process on $\mathcal{O}_*$, while L_δ is the Hölder constant of the $\delta-$th derivative of the trajectories.*

See, *e.g.*, [7] for some examples and references on processes satisfying the mild condition in (H2), which we impose on the $\delta-$th derivative of the trajectories. Examples include, but are not limited to, stationary or stationary increment processes X. The condition in (H3) serves to derive the exponential bound for the concentration of the local regularity estimator. By definition, when the local regularity α is an integer, $\delta = \alpha - 1$ and $H_\delta = 1$. The following result provides insight on the embedding structure of the spaces $\mathcal{X}(\cdot, \cdot; \mathcal{O}_*)$.

Lemma 7. *Let $\mathcal{O}_* \subset \mathcal{T}$ and assume that $\Delta_* \leq 1$. Let $\delta \in \mathbb{N}^*$ and assume that X restricted to $\mathcal{O}_*$ belongs to $\mathcal{X}(\delta + H_\delta, L_\delta; \mathcal{O}_*)$ for some $0 < H_\delta \leq 1$ and $L_\delta > 0$. Then, for any $d \in \{0, \dots, \delta - 1\}$, there exist two positive real numbers L_d and M_d such that*

$$\left|\mathbb{E}\left[(\nabla^d X_t - \nabla^d X_s)^2\right] - L_d^2|t - s|^2\right| \leq M_d^2|t - s|^2\Delta_*^{H_{d+1}}, \quad s, t \in \mathcal{O}_*,$$

with $H_{d+1} = \mathbf{1}_{\{d \neq \delta-1\}} + H_\delta \mathbf{1}_{\{d=\delta-1\}}$.

Lemma 7 indicates that whenever $\delta \geq 1$, for any $d \in \{0, \dots, \delta - 1\}$, the process X restricted to $\mathcal{O}_*$ belongs to $\mathcal{X}(d + 1, L_d; \mathcal{O}_*)$.

3.3.2 The local regularity estimation method

Let us assume that X restricted to $\mathcal{O}_*$ belongs to $\mathcal{X}(\delta + H_\delta, L_\delta; \mathcal{O}_*)$ for some $\delta \in \mathbb{N}$, $0 < H_\delta \leq 1$ and $L_\delta > 0$. Our first goal is to construct an estimator of $\alpha = \delta + H_\delta$. To do so, we first have to estimate H_d for any $d = 0, \dots, \delta$. For simplicity, let us denote

$$\theta_d(s, t) = \mathbb{E}\left[(\nabla^d X_t - \nabla^d X_s)^2\right] \quad s, t \in \mathcal{O}_*.$$

Since the restriction of the process X belongs to $\mathcal{X}(d + H_d, L_d; \mathcal{O}_*)$, with $H_d = \mathbf{1}_{\{d \neq \delta\}} + H_\delta \mathbf{1}_{\{d=\delta\}}$, we have

$$\theta_d(s, t) \approx L_d^2|t - s|^{2H_d} \quad \text{if } \Delta_* \text{ is small.} \tag{3.11}$$

Now, let t_1 and t_3 be such that $[t_1, t_3] \subset \mathcal{O}_*$ and $t_3 - t_1 = \Delta_*/2$. Denote by t_2 the middle point of $[t_1, t_3]$. It is easily seen that

$$H_d \approx \tilde{H}_d = \frac{\log(\theta_d(t_1, t_3)) - \log(\theta_d(t_1, t_2))}{2\log(2)} \quad \text{if } \Delta_* \text{ is small.}$$

Consider a learning sample of N_0 curves, generated according to (3.1), which yield

$$(Y_m^{(n)}, T_m^{(n)}) \in \mathbb{R} \times \mathcal{T}, \quad 1 \le m \le M_n,\ 1 \le n \le N_0.$$

Then, given a nonparametric estimator $\widetilde{\nabla^d X_t}$ of $\nabla^d X_t$, for $t \in \mathcal{O}_*$, we define a natural estimator of $\tilde{H}_d$, and thus of H_d, as

$$\hat{H}_d = \frac{\log\left(\hat{\theta}_d(t_1, t_3)\right) - \log\left(\hat{\theta}_d(t_1, t_2)\right)}{2\log(2)}, \tag{3.12}$$

where

$$\hat{\theta}_d(s,t) = \frac{1}{N_0} \sum_{n=1}^{N_0} \left(\widetilde{\nabla^d X_t}^{(n)} - \widetilde{\nabla^d X_s}^{(n)} \right)^2.$$

By the definition of the local regularity, $\delta = \min\{d \in \mathbb{N} : H_d < 1\}$, which suggests to define

$$\hat{\delta} = \min\{d = 0, \dots, D : \hat{H}_d < 1 - \varphi(\mathfrak{m})\},$$

for some decreasing function $\varphi(\cdot)$ which is defined later. Typically it could decrease as fast as a negative power of $\log(\mathfrak{m})$. The estimator of the local regularity is then

$$\hat{\alpha} = \hat{\delta} + \hat{H}_{\hat{\delta}}. \tag{3.13}$$

Our estimator $\hat{\alpha}$ is related to the estimator introduced in [59]. However, here we propose to smooth the trajectories to calculate $\hat{H}_d$ for $d = 0$, while Golovkine et al. [59] use directly the noisy measurements of the trajectories. To derive the properties of the local regularity estimator, for simplicity, we suppose that $\mathfrak{m}$ is given. In applications, it could be estimated by the average of the realizations of M.

3.3.3 Concentration properties of the local regularity estimator

The quality of the estimators $\hat{\delta}$ and $\hat{H}_d$ depends on the quality of the generic nonparametric estimators $\widetilde{\nabla^d X}$ of $\nabla^d X$. To quantify their behavior, we consider the local $\mathbb{L}^p$-risk

$$R_p(d) = R_p(d; \mathcal{O}_*) = \sup_{t \in \mathcal{O}_*} \mathbb{E}\left(|\xi_d(t)|^p\right), \quad \text{where} \quad \xi_d(t) = \widetilde{\nabla^d X}_t - \nabla^d X_t.$$

Our method applies with any type of nonparametric estimator $\widetilde{\nabla^d X}$ (local polynomials, splines,...) as soon as, for any $p \in \mathbb{N}$, its $\mathbb{L}^p$-risk of is bounded.

(LP1) There exist two positive constants $\mathfrak{c}$ and $\mathfrak{C}$ such that

$$R_{2p}(d) \leq \frac{p!}{2} \mathfrak{c} \mathfrak{C}^{p-2}, \qquad \forall p \geq 1,\ d \in \{0, \ldots, \delta\}.$$

We can now derive an exponential bound for the concentration of all the estimators $\hat{H}_d$, $d \in \{0, \ldots, \delta\}$. To make this exponential bound useful for deriving optimal rates for our estimators of the mean and covariance functions, we will require the largest quadratic risk among $R_2(0), \ldots, R_2(\delta)$ to tend to zero as $\mathfrak{m}$ increase to infinity.

Lemma 8. *Assume that X restricted to $\mathcal{O}_*$ belongs to $\mathcal{X}(\delta + H_\delta, L_\delta; \mathcal{O}_*)$, for some integer $\delta \geq 0$ and $0 < H_\delta \leq 1$, and that (LP1) holds. Let*

$$A_1 = \max_{d \in \{0,\ldots,\delta\}} \left[\frac{2^{4H_d+3}}{L_d^2} \left(\sqrt{\frac{\mathfrak{a}}{\mathfrak{A}}} + 1 \right) \right]^{\frac{1}{2H_d}} \quad \text{and} \quad A_2 = \min_{d \in \{0,\ldots,\delta\}} \left[\frac{1}{2} \left(\frac{L_d}{M_d} \right)^2 \right]^{\frac{1}{H_{d+1}}} \wedge 1,$$

and define

$$\rho^* = \max_{d \in \{0,\ldots,\delta\}} \left\{ (R_2(d))^{\frac{1}{4H_d}} \right\}.$$

Assume that $A_1 \rho^* \leq \Delta^* \leq A_2$. Then for any $\epsilon > 0$ such that

$$\mathfrak{M} \max(\Delta_*^{H_\delta}, \Delta_*^{\beta_\delta}) < \epsilon \log(2) < 2 \quad \text{with} \quad \mathfrak{M} = 4 \max_{d \in \{0,\ldots,\delta\}} \left(\frac{M_d}{L_d} \right)^2,$$

we have

$$\max_{d \in \{0,\ldots,\delta\}} \mathbb{P}\left(|\hat{H}_d - H_d| > \epsilon \right) \leq 4 \exp\left(-\mathfrak{f} N_0 \epsilon^2 \Delta_*^{4H_\delta} \right),$$

where $\mathfrak{f}$ is a constant depending on H_δ and $\mathfrak{a}$, $\mathfrak{A}$, $\mathfrak{c}$, $\mathfrak{C}$, $\underline{a}_0, \ldots, \underline{a}_\delta$ and β_δ from the conditions (H1) to (H3) and (LP1).

Theorem 4. *Assume that the conditions of Lemma 8 hold true. Assume also that there exists* $\tau > 0$ *and* $B > 0$ *such that:*

$$\rho^* = \max_{d\in\{0,\dots,\delta\}} \left\{ (R_2(d))^{\frac{1}{4H_d}} \right\} \leq B\mathfrak{m}^{-\tau}.$$

Let $0 < \gamma < 1$ *and* $\Gamma > 0$*, and consider*

$$\Delta_* = \exp\left(-\log^{\gamma}(\mathfrak{m})\right) \quad \textit{and} \quad \varphi(\mathfrak{m}) = \log^{-\Gamma}(\mathfrak{m}).$$

Then, for any $\mathfrak{m}$ *larger than some constant* $\mathfrak{m}_0$ *depending on* B*,* τ*,* γ*,* Γ*,* H_δ*,* β_δ *and for* $\mathfrak{f}$ *the constant from Lemma 8, we have*

$$\mathbb{P}\left(|\hat{\alpha} - \alpha| > \varphi(\mathfrak{m})\right) \leq 8(1+\delta)\exp\left(-\mathfrak{f}N_0\varphi^2(\mathfrak{m})\Delta_*^{4H_\delta}\right).$$

The three quantities ρ^*, Δ_* and $\varphi(\mathfrak{m})$ are required to decrease to zero, as $\mathfrak{m}$ tends to infinity, in such way that $\rho^*/\Delta_* + \Delta^*/\varphi(\mathfrak{m}) \to 0$. We propose $\Gamma = 2$ and $\gamma = 1/2$. The choices of the rates for ρ^*, Δ_* and $\varphi(\mathfrak{m})$ satisfy some additional requirements. First, in order to achieve optimal rates of convergence for the mean and covariance estimators, the local regularity has to be estimated with a concentration rate faster than $\log^{-1}(\mathfrak{m})$. This is a consequence of the identity $\mathfrak{m}^{1/\log(\mathfrak{m})} = e$ for any $\mathfrak{m} > 1$. Second, we want to allow for reasonable rates of increase for N_0, the size of the learning set. In Theorem 4, N_0 can increase as slow as any positive power of $\mathfrak{m}$. Third, since $\tau > 0$ could arbitrarily small, the rate imposed on the nonparametric estimators $\widetilde{\nabla^d X}$ of $\nabla^d X$ is a very mild requirement which could be achieved by the common estimators, with random or fixed design, under mild conditions. See, for instance, [135, 2]. In particular, the required rate for the $\widetilde{\nabla^d X}$ can be obtained under general forms of heteroscedasticity.

3.3.4 From process regularity to trajectories regularity

Let us now connect the probabilistic concept of local regularity with the regularity of the sample paths considered as functions. Let $\mathcal{O}_*$ be some open subinterval of $\mathcal{T}$. We say that a function is β times differentiable on $\mathcal{O}_*$ if the function has an up to $\lfloor\beta\rfloor-$order derivative and the $\lfloor\beta\rfloor-$th derivative is Hölder continuous with exponent $\beta - \lfloor\beta\rfloor$. (For a real number a, let $\lfloor a\rfloor$ denotes the largest integer not exceeding a.) Let us call a Hölder space of local regularity β, the set of functions which are β times differentiable on $\mathcal{O}_*$.

By Assumption (H1), for almost all realizations of the process X, the derivatives of

the sample path exist up to order δ. Under a suitable moment condition on the increments the derivative process $\nabla^{\delta} X$, and using the refined version of Kolmogorov's criterion stated in Revuz and Yor [116], it can be proven that, for any $\delta < \beta < \delta + H_{\delta} = \alpha$, the sample path of the process X restricted to $\mathcal{O}_*$ belong to the Hölder space of exponent β over $\mathcal{O}_*$. As an example, the Brownian motion has a local regularity equal to $1/2$ on any open interval. Almost surely, the sample paths of the Brownian motion belong to any Hölder space of local regularity $\beta < 1/2$, but cannot have a Hölder continuity of order α, $\alpha \geq 1/2$. Hence, the probability theory indicates that imposing assumptions on the regularity of the sample paths could be a delicate issue. Indeed, even for some widely used examples, this regularity is not well defined in the sense required by the nonparametric statistics theory. Since the sample paths have a regularity which can be arbitrarily close to the (local) regularity of the process X as defined above, the probabilistic concept of local regularity seems more appropriate for establishing the rates of convergence for the mean and covariance estimators.

3.4 Optimal mean and covariance estimators

We now explain how to suitably select the bandwidths for the local polynomial smoothing of the trajectories, and next build mean and covariance function estimates. Our data-driven adaptive bandwidth rules lead to optimal rate estimates whenever the estimator of the local regularity concentrates to the true value faster than $\log^{-1}(\mathfrak{m})$. Theorem 4 then guarantees that the suitable rate of the bandwidth can be achieved with high probability.

Motivated by the applications, we allow the process X to have a piecewise constant regularity.

(E2) There exist $K \geq 1$ and $\min \mathcal{T} = t_0 < t_1 < \ldots t_{K-1} < t_K = \max \mathcal{T}$ such that, for each $\mathcal{T}_k^{\circ} = (t_{k-1}, t_k)$,

$$X \text{ restricted to } \mathcal{T}_k^{\circ} \text{ belongs to } \mathcal{X}(\delta_k + H_{\delta_k}, L_{\delta_k}; \mathcal{T}_k^{\circ}),$$

for some integer $\delta_k \geq 0$, $0 < H_{\delta_k} \leq 1$ and $L_{\delta_k} > 0$.

For any $t \in \mathcal{T}$, let α_t be the local regularity from Definition 1 corresponding to a small neighborhood of t, and let L_{δ}, with $\delta = \lfloor \alpha_t \rfloor$, denote the corresponding Hölder constant. By our assumptions, α_t is well defined everywhere except a finite number of points in

$\mathcal{T}$. For a small neighborhood of points like t_k, the values of α_t are different for t to the left and to the right of t_k. However, for practical purposes, any convention would work, such as for instance, setting α_{t_k} equal to the average of the neighboring values. Hereafter therefore we assume that α_t is well-defined everywhere on $\mathcal{T}$.

Hereafter, $\hat{\alpha}_t$ will be the estimator of α_t defined in (3.13) and (3.12), built with an independent learning sample, and with $\varphi(\mathfrak{m})$ replaced by $\log^{-2}(\widehat{\mathfrak{m}})$, where $\widehat{\mathfrak{m}} = N^{-1}\sum_{i=1}^{N} M_i$. If $t_1 < \ldots < t_{K-1}$ are known, then for each $t \in \mathcal{T}_k^\circ$, $\hat{\alpha}_t$ can be defined as in (3.13) and (3.12) with t_2 the middle point in $\mathcal{T}_k^\circ$. In practice the $t_1, \ldots, t_{K-1}$ are likely not given. One could then estimate the local regularity on a grid on points in $\mathcal{T}$, and let each point on the grid play the role of t_2 in (3.12). Such implementation is used in Section 3.5.

For each $1 \leq i \leq N$, we use a local polynomials (LP) approach to build suitable nonparametric estimators $\widehat{X}^i$. For $t_0 \in \mathcal{T}$, if the local regularity α_t is available, using the measurements $(Y_m^{(n)}, T_m^{(n)})$, $1 \leq m \leq M_i$, of a generic trajectory $X^{(i)}$, we consider the $LP(\lfloor \varsigma_{t_0} \rfloor)$ estimator defined by

$$\widehat{X}_t^i = \widehat{X}_t^i(h_t) = \sum_{m=1}^{M_i} Y_m^{(n)} W_m^{(i)}(t), \qquad 1 \leq i \leq N, \tag{3.14}$$

with a suitable bandwidth h_t which depends on α_t, and

$$W_m^{(i)}(t) = W_m^{(i)}(t;h) = \frac{1}{M_i h} U^\top(0) A_{M_i}^{(i)}(t,h)^{-1} U\left(\frac{T_m^{(n)} - t_0}{h}\right) K\left(\frac{T_m^{(n)} - t_0}{h}\right), \quad 1 \leq m \leq M_i,$$

with

$$A_{M_i}^{(i)}(t,h) = \frac{1}{M_i h} \sum_{m=1}^{M_i} U\left(\frac{T_m^{(n)} - t_0}{h}\right) U^\top\left(\frac{T_m^{(n)} - t_0}{h}\right) K\left(\frac{T_m^{(n)} - t_0}{h}\right).$$

Here, for any $z \in \mathbb{R}$, $U(z) = U(z;\alpha_t) = (1, z, \ldots, z^{\lfloor \varsigma_{t_0} \rfloor}/\lfloor \varsigma_{t_0} \rfloor!)^\top$ and $K : \mathbb{R} \to \mathbb{R}_+$ is a continuous kernel with the support in $[-1,1]$.

For a given $t_0 \in \mathcal{T}$ and each $1 \leq i \leq N$, the weights $W_m^{(i)}(t)$, $1 \leq m \leq M_i$ are well defined as soon as the matrix $A_{M_i}^{(i)}(t,h)$ is invertible. When $A_{M_i}^{(i)}(t,h)$ is not invertible, we consider the smoothing of the curve i at point t as degenerate and the curve i should not be considered in the construction of the estimator of $\mu(t)$. A similar reasoning applies to the covariance estimator. In the case of common design, for each t, the number of degenerate estimates $\widehat{X}_t^i$ is either equal to N or to zero. In the independent design case, this number could be any integer between 0 and N. A suitable bandwidth rule should be penalizing for the number of curves which are not considered for the estimation. In the

following sections we propose a natural way to penalize which adapts to the sparse and dense regimes. Moreover, the two types of designs are automatically handled.

3.4.1 Adaptive mean estimation

Let k_0 be some integer, and let $\mathbf{1}\{\cdot\}$ denote the indicator function. For any $t_0 \in \mathcal{T}$, define

$$w_i(t;h) = 1 \quad \text{if} \quad \sum_{m=1}^{M_i} \mathbf{1}\{|T_m^{(n)} - t| \leq h\} \geq k_0, \quad \text{and} \quad w_i(t;h) = 0 \text{ otherwise,} \tag{3.15}$$

and let

$$\mathcal{W}_N(t;h) = \sum_{i=1}^{N} w_i(t;h).$$

Our adaptive mean function estimator is

$$\widehat{\mu}_N^*(t) = \widehat{\mu}_N(t;h_\mu^*) \quad \text{with} \quad \widehat{\mu}_N(t;h) = \frac{1}{\mathcal{W}_N(t;h)} \sum_{i=1}^{N} w_i(t;h)\widehat{X}_t^i. \tag{3.16}$$

Here, $\widehat{X}_t^i$ is the local polynomial estimator $LP(\lfloor\hat{\alpha}_t\rfloor)$ defined in (3.14) with some suitable bandwidth h_μ^* which is defined below. The mean estimator $\widehat{\mu}_N(t;h)$ is a practical version of that defined in (3.2) which takes into account that some trajectories could have less than k_0 observation times between $t-h_\mu^*$ and $t+h_\mu^*$. The threshold defined by k_0 avoids considering degenerate $\widehat{X}_t^i$. For this purpose, it has to be greater than or equal to $\lfloor\hat{\alpha}_t\rfloor + 1$. The normalization of the mean estimator by $\mathcal{W}_N(t;h)$ is also implicitly used in the definition of the estimators proposed by [91] and [152].

To introduce our bandwidth rule, for any $h > 0$, $\alpha > 0$, let

$$c_i(t;h) = \sum_{m=1}^{M_i} \left|W_m^{(i)}(t;h)\right| \quad \text{and} \quad c_i(t;h,\alpha) = \sum_{m=1}^{M_i} \left|(T_m^{(n)} - t)/h\right|^\alpha \left|W_m^{(i)}(t;h)\right|, \tag{3.17}$$

and

$$\overline{C}_1(t;h,\alpha) = \frac{1}{\mathcal{W}_N(t;h)} \sum_{i=1}^{N} w_i(t;h)c_i(t;h)c_i(t;h,\alpha).$$

With the Nadaraya-Watson (NW) estimator, all the $c_i(t;h)$ are equal to 1. Moreover,

$$\overline{C}_1(t;h,\alpha) \approx \int |u|^\alpha K(u)du. \tag{3.18}$$

Using the equivalent kernels idea, see section 3.2.2 in [45], the same approximation could

be used in the case of local linear estimators. The accuracy of the approximation (3.18) could be high since it involves the $T_m^{(n)}$ close to t for all the curves with $w_i(t;h) = 1$. Next, using the rule $0/0 = 0$, let

$$\mathcal{N}_i(t;h) = \frac{w_i(t;h)}{\max_{1\leq m\leq M_i} |W_m^{(i)}(t;h)|} \quad \text{and} \quad \mathcal{N}_\mu(t;h) = \left[\frac{1}{\mathcal{W}_N^2(t;h)} \sum_{i=1}^{N} w_i(t;h) \frac{c_i(t;h)}{\mathcal{N}_i(t;h)}\right]^{-1}. \tag{3.19}$$

With the NW estimator, $\mathcal{N}_\mu(t;h)$ is equal to $\mathcal{W}_N(t;h)$ times the harmonic mean of $\mathcal{N}_i(t;h)$, over the curves with $w_i(t;h) = 1$. Moreover, under the mild condition (3.23) below, $\mathcal{N}_i(t;h) = O_{\mathbb{P}}(\mathfrak{m}h)$.

Quantities like $c_i(t;h)$, $c_i(t;h,\alpha)$ and $\max_m |W_m^{(i)}(t,h)|$ are commonly used to bound the risk of LP estimators, see also Section 3.2 above. For deriving theoretical results, $c_i(t;h)$ and $c_i(t;h,\alpha)$ are usually bounded by a constant. Meanwhile, the maximum of the absolute values of the LP weights could be bounded by a suitable constant divided by the number of observation times between $t - h_\mu^*$ and $t + h_\mu^*$. See [61] or [135]. In the context of functional data, we directly use the $c_i(t;h)$, $c_i(t;h,\alpha)$ and $\mathcal{N}_i(t;h)$ and thus entirely exploit the information contained in all the curves. The computational complexity remains low and the reward is a sharper risk bound, a better bandwidth choice and an improved mean estimator in finite samples.

We define the bandwidth for computing $\widehat{\mu}_N^*(t)$ such that it minimizes the mean squared difference between $\widehat{\mu}_N(t;h)$ and $\widetilde{\mu}_N(t)$. This leads us to define the optimal bandwidth

$$h_\mu^* = h_\mu^*(t) = \arg\min_{h>0} \mathcal{R}_\mu(t;h), \tag{3.20}$$

with

$$\mathcal{R}_\mu(t;h) = q_1^2 h^{2\hat{\alpha}_t} + \frac{q_2^2}{\mathcal{N}_\mu(t;h)} + q_3^2 \left[\frac{1}{\mathcal{W}_N(t;h)} - \frac{1}{N}\right], \tag{3.21}$$

and

$$q_1^2 = \frac{\overline{C}_1(t;h,2\hat{\alpha}_t)}{\lfloor \hat{\alpha}_t \rfloor !^2} \hat{L}_\delta^2\,, \quad q_2^2 = \sigma_{\max}^2\,, \quad q_3^2 = Var(X_t),$$

where $\sigma_{\max}$ is a bound for the function $\sigma(t,x)$ in (3.5) and $\hat{L}_\delta$ is an estimate of the Hölder constant L_δ from Assumption (E2). In Section 3.5, we propose a simple procedure to build $\hat{L}_\delta$ based on the preliminary nonparametric estimates of the sample paths used for $\hat{\alpha}_t$. We show in the Appendix that $2\mathcal{R}_\mu(t;h)$ is a sharp bound for $\mathbb{E}_{M,T}\left[\{\widehat{\mu}_N(t;h) - \widetilde{\mu}_N(t)\}^2\right]$. The maximization of $\mathcal{R}_\mu(t;h)$ can be easily performed on a grid of h values. With some

additional effort, $\mathcal{R}_\mu(t;h)$ can also be minimized with respect to k_0 in (3.15) over a small set of integers greater than $\lfloor \hat{\alpha}_t \rfloor$.

The bandwidth rule (3.20) could be used with both independent and common design. With common design, the $T_m^{(n)} \equiv T_m$ and $W_m^{(i)}(t;h) \equiv W_m(t;h)$ no longer depend on i and the solution h_μ^* will always be a value in the set of h such that $\mathcal{W}_N(t;h) = N$. Moreover, whenever $\mathcal{W}_N(t;h) = N$,

$$\overline{C}_1(t;h,2\hat{\alpha}_t) = \sum_{m=1}^{\mathfrak{m}} |W_m(t;h)| \sum_{m=1}^{\mathfrak{m}} \left|\frac{T_m - t}{h}\right|^{2\hat{\alpha}_t} |W_m(t;h)| \quad \text{and} \quad \mathcal{N}_\mu(t;h) = \frac{N \sum_{m=1}^{\mathfrak{m}} |W_m(t;h)|}{\max_{1\le m\le \mathfrak{m}} |W_m(t;h)|}.$$

In a data-driven way, h_μ^* automatically chooses between interpolation and smoothing.

The following result states that our estimator $\widehat{\mu}_N^*(t)$ defined by (3.16) and (3.20) achieves the optimal rate. For simplicity, we assume that

$$\limsup_{N,\mathfrak{m}\to\infty} \{\log(N)/\log(\mathfrak{m})\} < \infty, \tag{3.22}$$

a technical condition which matches general situations found in applications. We also impose the following mild technical condition in the independent design case:

$$\exists c_L, C_U > 0 \ \text{ such that } \ c_L \le M_i \mathfrak{m}^{-1} \le C_U, \quad \text{for all } N \text{ and } 1 \le i \le N. \tag{3.23}$$

Moreover, in the common design case, where $M_i \equiv \mathfrak{m}$ and the $T_1^{(i)}, \ldots, T_{\mathfrak{m}}^{(i)}$ are not changing with i, we suppose that:

$$\exists C_U \ge 1 \ \text{ such that } \ \max_{1\le m\le \mathfrak{m}-1} \{T_{m+1}^{(i)} - T_m^{(i)}\} \le C_U \min_{1\le m\le \mathfrak{m}-1} \{T_{m+1}^{(i)} - T_m^{(i)}\}. \tag{3.24}$$

Theorem 5. *Assume that $T_m^{(n)}$ are either independently drawn, have a continuous density which is bounded away from zero and (3.23) holds true, or $T_m^{(n)}$ represent the points of a common design satisfying (3.24). Moreover, (3.22) and Assumption (E2) hold true. For $t \in \mathcal{T}$, let $\hat{\alpha}_t$ be an estimator of $\alpha_t < 1$ computed on a separate sample such that $\hat{\alpha}_t - \alpha_t = o_{\mathbb{P}}(\log^{-1}(\mathfrak{m}))$. Then, the estimator $\widehat{\mu}_N^*(t) = \widehat{\mu}_N(t;h_\mu^*)$ defined by (3.16) and (3.20) satisfies*

$$\widehat{\mu}_N^*(t) - \widetilde{\mu}_N(t) = O_{\mathbb{P}}\left((N\mathfrak{m})^{-\frac{\alpha_t}{1+2\alpha_t}}\right) \quad \text{and} \quad \widehat{\mu}_N^*(t) - \mu(t) = O_{\mathbb{P}}\left((N\mathfrak{m})^{-\frac{\alpha_t}{1+2\alpha_t}} + N^{-1/2}\right),$$

in the independent design case. Meanwhile, with the common design,

$$\widehat{\mu}_N^*(t) - \widetilde{\mu}_N(t) = O_{\mathbb{P}}\left(\max\left\{(N\mathfrak{m})^{-\frac{\alpha_t}{1+2\alpha_t}}, \mathfrak{m}^{-\alpha_t}\right\}\right) = O_{\mathbb{P}}\left(\mathfrak{m}^{-\alpha_t}\right),$$

and

$$\widehat{\mu}_N^*(t) - \mu(t) = O_{\mathbb{P}}\left(\max\left\{(N\mathfrak{m})^{-\frac{\alpha_t}{1+2\alpha_t}}, \mathfrak{m}^{-\alpha_t}\right\} + N^{-1/2}\right) = O_{\mathbb{P}}\left(\mathfrak{m}^{-\alpha_t}\right).$$

The rates achieved by $\widehat{\mu}_N^*(t)$ are the best one could expect in view of the results of [19]. To avoid additional technical arguments, we only prove our result for a local regularity less than 1 and Nadaraya-Watson estimators $\widehat{X}_t^i$. We conjecture that it also true for $\alpha_t \geq 1$, but we leave formal justification for future work. The difference between the common and independent designs comes from the fact that, in order to avoid degenerate mean estimator, the bandwidth cannot decrease faster than $\mathfrak{m}^{-1}$.

3.4.2 Adaptive covariance function estimates

For any $s, t_0 \in \mathcal{T}$, $s \neq t$, define

$$w_i(s,t;h) = w_i(s;h)w_i(t;h) \qquad \text{and} \qquad \mathcal{W}_N(s,t;h) = \sum_{i=1}^N w_i(s,t;h),$$

with $w_i(s;h)$ and $w_i(t;h)$ as in (3.15). Our adaptive covariance function estimator is

$$\widehat{\Gamma}_N^*(s,t) = \widehat{\Gamma}_N(s,t;h_\Gamma^*) \quad \text{with} \quad \widehat{\Gamma}_N(s,t;h) = \widehat{\gamma}_N(s,t;h) - \widehat{\mu}_N^*(s)\widehat{\mu}_N^*(t), \tag{3.25}$$

where $\widehat{\mu}_N^*(s)$, $\widehat{\mu}_N^*(t)$ are defined according to (3.16) with the corresponding bandwidths, and

$$\widehat{\gamma}_N(s,t;h) = \frac{1}{\mathcal{W}_N(s,t;h)} \sum_{i=1}^N w_i(s,t;h)\widehat{X}_s^i\widehat{X}_t^i. \tag{3.26}$$

Here, $\widehat{X}_s^i$ and $\widehat{X}_t^i$ are the local polynomial estimators $LP(\lfloor\hat{\alpha}_s\rfloor)$ and $LP(\lfloor\hat{\alpha}_t\rfloor)$ built, respectively, with some suitable bandwidth h_Γ^* which is defined below. This covariance function estimator is a practical version of that defined in (3.3). The normalization of the mean estimator by $\mathcal{W}_N(s,t;h)$ is also implicitly used in the definition of the estimators proposed by [91] and [152].

We define the bandwidth for computing $\widehat{\gamma}_N(s,t;h)$, and eventually $\widehat{\Gamma}_N^*(s,t)$, such that it minimizes the mean squared difference between $\widehat{\gamma}_N(s,t;h)$ and the unfeasible estimator $\widetilde{\gamma}_N(s,t) = N^{-1}\sum_{i=1}^N X_s^{(i)}X_t^{(i)}$ of $\mathbb{E}(X_s^{(i)}X_t^{(i)})$. To this aim, we define modified versions of

$\mathcal{N}_i(t;h)$ and $\mathcal{N}_\mu(t;h)$, see (3.19), taking into account only the curves with $w_i(s,t;h)=1$:

$$\mathcal{N}_i(t|s;h)=\frac{w_i(s,t;h)}{\max_{1\le m\le M_i}|W_m^{(i)}(t,h)|},$$

and

$$\mathcal{N}_\Gamma(t|s;h)=\left[\frac{1}{\mathcal{W}_N^2(s,t;h)}\sum_{i=1}^{N}w_i(s,t;h)\frac{c_i(t;h)}{\mathcal{N}_i(t|s;h)}\right]^{-1},$$

where the $c_i(t;h)$ are defined as in (3.17). This idea leads us to define the optimal bandwidth as

$$h_\Gamma^*=h_\Gamma^*(s,t)=\arg\min_{h>0}\{\mathcal{R}_\Gamma(s|t;h)+\mathcal{R}_\Gamma(t|s;h)\},\tag{3.27}$$

with

$$\mathcal{R}_\Gamma(t|s;h)=\mathfrak{q}_1^2h^{2\hat{\alpha}_t}+\frac{\mathfrak{q}_2^2}{\mathcal{N}_\Gamma(t|s;h)}+\mathfrak{q}_3^2\left[\frac{1}{\mathcal{W}_N(s,t;h)}-\frac{1}{N}\right].\tag{3.28}$$

In the last equation, $\mathfrak{q}_\ell$, $1\le l\le 3$, are defined by:

$$\mathfrak{q}_1^2=2\mathbb{E}(X_s^2)\frac{\overline{\mathfrak{C}}_1(t;h,2\hat{\alpha}_t)}{\lfloor\hat{\alpha}_t\rfloor!^2}\hat{L}_\delta^2\qquad\mathfrak{q}_2^2=\sigma_{\max}^2\,\mathbb{E}(X_s^2),\qquad\mathfrak{q}_3^2=\frac{Var(X_sX_t)}{2},$$

where

$$\overline{\mathfrak{C}}_1(t|s;h,\alpha)=\frac{\sum_{i=1}^{N}w_i(s,t;h)c_i(t;h)c_i(t;h,\alpha)}{\mathcal{W}_N(s,t;h)}.$$

With NW or local linear estimators,

$$\overline{\mathfrak{C}}_1(t|s;h,\alpha)\approx\int|u|^\alpha K(u)du.$$

The definition of $\mathcal{R}_\Gamma(s|t;h)$ is the same as in (3.28) with the occurrences of s and t switched. The function of h minimized in (3.27) is a sharp bound for

$$\mathbb{E}_{M,T}\left[\{\widehat{\gamma}_N(s,t;h)-\widetilde{\gamma}_N(s,t)\}^2\right]/2.$$

The sum of the first two terms in the expressions of $\mathcal{R}_\Gamma(s|t;h)$ and $\mathcal{R}_\Gamma(t|s;h)$ represents the quadratic risk of our estimator of $\mathbb{E}(X_sX_t)$ compared to the unfeasible one based on the true values $X_s^{(i)}X_t^{(i)}$ from the curves yielding non-degenerate estimates $\widehat{X}_s^i\widehat{X}_t^i$. Like for the mean function, the third term in (3.28) penalizes for the number of curves which are dropped when calculating our estimator. The minimization of $\mathcal{R}_\Gamma(s|t;h)+\mathcal{R}_\Gamma(t|s;h)$ can be performed on a grid of values h. The minimization could be also done over a set of

values k_0 used in (3.15).

Like for the mean function, with obvious modifications, the definition (3.27) could be used with both independent and common design. Indeed, with common design, the solution h^*_Γ will always be a value in the set of h such that $\mathcal{W}_N(s,t;h) = N$. In a completely data-driven way, h^*_Γ will automatically choose between interpolation and smoothing.

Theorem 6. *Assume that the conditions of Theorem 5 are met for $s,t \in \mathcal{T}$ such that $s \neq t$. Moreover, $\sup_{t\in\mathcal{T}} \mathbb{E}(X_t^4) < \infty$. Let $\alpha(s,t) = \min\{\alpha_s, \alpha_t\}$ and assume $\alpha(s,t) < 1$. Then the estimator $\widehat{\Gamma}^*_N(s,t) = \widehat{\Gamma}^*_N(s,t;h^*_\Gamma)$ defined by (3.25) and (3.27) satisfies*

$$\widehat{\Gamma}^*_N(s,t) - \widetilde{\Gamma}_N(s,t) = O_{\mathbb{P}}\left((N\mathfrak{m})^{-\frac{\alpha(s,t)}{1+2\alpha(s,t)}}\right)$$

and

$$\widehat{\Gamma}^*_N(s,t) - \Gamma(s,t) = O_{\mathbb{P}}\left((N\mathfrak{m})^{-\frac{\alpha(s,t)}{1+2\alpha(s,t)}} + N^{-1/2}\right),$$

in the independent design case. Meanwhile with the common design,

$$\widehat{\Gamma}^*_N(s,t) - \widetilde{\Gamma}_N(s,t) = O_{\mathbb{P}}\left(\max\left\{(N\mathfrak{m})^{-\frac{\alpha(s,t)}{1+2\alpha(s,t)}}, \mathfrak{m}^{-\alpha(s,t)}\right\}\right) = O_{\mathbb{P}}\left(\mathfrak{m}^{-\alpha(s,t)}\right),$$

and

$$\widehat{\Gamma}^*_N(s,t) - \Gamma(s,t) = O_{\mathbb{P}}\left(\max\left\{(N\mathfrak{m})^{-\frac{\alpha(s,t)}{1+2\alpha(s,t)}}, \mathfrak{m}^{-\alpha(s,t)}\right\} + N^{-1/2}\right) = O_{\mathbb{P}}\left(\mathfrak{m}^{-\alpha(s,t)}\right).$$

The rates achieved by $\widehat{\Gamma}^*_N(s,t)$ are the best one could expect in view of the results of [18]. For now, we only prove our result for the case $\min\{\alpha_s, \alpha_t\} < 1$ and Nadaraya-Watson estimators $\widehat{X}^i_t$. We conjecture that it also true in the general case.

3.4.3 Estimator on the diagonal band of the covariance function

As mentioned in (3.3) and (3.4), we propose to use the estimator (3.26) only outside the diagonal set $\{(s,t) : |s-t| \leq \mathfrak{d}\}$, for some suitable $\mathfrak{d} > 0$. It remains to give a data-driven rule for choosing $\mathfrak{d}$ decreasing to zero, and propose an estimator for $\mathbb{E}(X_sX_t)$ when s and t are in the diagonal set. Let $\mathcal{T}^\circ_k$, $1 \leq k \leq K$, as defined in Assumption (E2). With a piecewise constant regularity, the value of $\mathfrak{d}$ depends on k.

To understand how to build a covariance estimator which is optimal in integrated

squared norm, for $1 \leq k \leq K$ and $s, t \in \mathcal{T}_k^\circ$ such that $s \leq t$, let $u = (s+t)/2$ and

$$\widetilde{D}(\mathfrak{d}) := \iint_{t-\mathfrak{d} \leq s \leq t} \left\{ \widetilde{\Gamma}_N(u - \mathfrak{d}/2, u + \mathfrak{d}/2) - \widetilde{\Gamma}_N(s,t) \right\}^2 dsdt.$$

Let α denote the local regularity corresponding to $u = (s+t)/2$. Under mild assumptions on the moments of X_t and $X_t - X_s$, we show in the Appendix that

$$\widetilde{D}(\delta) = O_{\mathbb{P}}\left(\mathfrak{d}^{2\alpha+1}\right). \tag{3.29}$$

Thus, for (s,t) inside the diagonal band, it suffices to use $\widehat{\Gamma}_N(u - \mathfrak{d}/2, u + \mathfrak{d}/2)$ defined according to (3.25) when $s \leq t$ and use the symmetry of the covariance function when $s > t$. That is, in order to estimate the covariance function at a point from the diagonal set, we simply apply the covariance function estimator, designed for outside the diagonal set, for the closest point on the boundary of the diagonal band. The rate (3.29) indicates that in order to make this a suitable estimator, $\mathfrak{d}$ should have a rate of decrease slower than the power $(2\alpha+1)^{-1}$ of the optimal rate achievable by a nonparametric estimator of the covariance. If $\hat{\alpha}$ is the estimate of α, we can take

$$\mathfrak{d} = \left\{ N^{-2} \sum_{i=1}^{N} (1/M_i) \right\}^c \quad \text{for some} \quad \frac{2\hat{\alpha}}{\{2\hat{\alpha}+1\}^2} < c < \frac{1}{2\hat{\alpha}+1},$$

for instance $c = \{2\hat{\alpha} + 1/2\}/\{2\hat{\alpha}+1\}^2$.

3.5 Empirical study

3.5.1 Implementation aspects

The risks $\mathcal{R}_\mu$ and $\mathcal{R}_\Gamma$ defined in (3.21) and (3.28), respectively, depend on L_δ^2 and the conditional variance bound $\sigma^2_{\max}$. In view of (3.11), if t_2 is the midpoint of $[t_1, t_3]$,

$$L_\delta^2 \approx \frac{1}{2}\left(\frac{\theta_\delta(t_2,t_3)}{|t_3 - t_2|^{2(\alpha-\delta)}} + \frac{\theta_\delta(t_1,t_2)}{|t_2 - t_1|^{2(\alpha-\delta)}} \right),$$

provided $t_3 - t_1 = \Delta_*/2$ is small. Given the estimate $\hat{\alpha}_t$ and the estimators $\hat{\theta}_\delta(t_2, t_3)$ and $\hat{\theta}_\delta(t_1, t_2)$ as in (3.12), with $\delta = \lfloor \hat{\alpha}_t \rfloor$, we then define a natural estimator of L_δ^2 as

$$\widehat{L}_\delta^2 \approx \frac{1}{2}\left(\frac{\hat{\theta}_{\lfloor \hat{\alpha}_t \rfloor}(t_2, t_3)}{|t_3 - t_2|^{2(\hat{\alpha}_t - \lfloor \hat{\alpha}_t \rfloor)}} + \frac{\hat{\theta}_{\lfloor \hat{\alpha}_t \rfloor}(t_1, t_2)}{|t_2 - t_1|^{2(\hat{\alpha}_t - \lfloor \hat{\alpha}_t \rfloor)}} \right).$$

To estimate the conditional variance bound, let us first consider the case where $\sigma^2(t, x)$ does not depend on x. In this case, one can compute

$$\widehat{\sigma}^2(t) = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{2|\mathcal{S}_i|} \sum_{m \in \mathcal{S}_i} \left[Y_m^{(i)} - Y_{m-1}^{(i)} \right]^2, \tag{3.30}$$

where $\mathcal{S}_i$ is a subset of indices m for the $i-$th trajectory. When the variance of the errors is considered constant, $\mathcal{S}_i$ can be the set $\{2, 3, \ldots, M_i\}$. When the variance depends on t_0, one could define $\mathcal{S}_i$ as the set of indices corresponding to the K_0 values $T_m^{(i)}$ closest to t. The theory allows a choice such as $K_0 = \lfloor \hat{\mathfrak{m}} \exp\left(-\{\log\log \hat{\mathfrak{m}}\}^2\right) \rfloor$, with $\hat{\mathfrak{m}} = N^{-1}\sum_{i=1}^{N} M_i$. Then $\sigma^2_{\max}$ could be $\max_{t \in \mathcal{T}} \widehat{\sigma}^2(t)$, and this choice was used in our empirical investigation. When the variance of the errors also depends on the realizations X_u, in general it is no longer possible to consistently estimate $\sigma^2(u, X_u)$. However, the simple inspection of the squared differences between the observations $Y_m^{(i)}$ and the presmoothed values allows to build a bound $\sigma^2_{\max}$. In our simulation experiments, $\sigma^2_{\max}$ obtained from the estimate (3.30), with $\mathcal{S}_i$ given by K_0, works well.

3.5.2 Simulation design

Our simulation study is based on the Household Active Power Consumption dataset which was sourced from the UC Irvine Machine Learning Repository. This dataset contains diverse energy related features gathered in a house located near Paris, every minute between December 2006 and November 2010. In total, its represents around 2 million data points. Here, we are only interested in the daily voltage and we only consider the days without missing values in the measurements. The extracted dataset contains 708 voltage curves with an uniform common design with 1440 points.

The 708 voltage curves are used to build a mean function $\mu(\cdot)$, and a covariance function $\Gamma(\cdot, \cdot)$. We also derive a conditional variance function $\sigma^2(t, x)$ for the noise which we plot in Figure 3.1a. Next, we generate samples of independent trajectories from the Gaussian process characterized by these $\mu(\cdot)$ and $\Gamma(\cdot, \cdot)$. Their local regularity is approxi-

mately equal to 0.7. Finally, we add the heteroscedastic noise. A random sample of curves generated according to our simulation setup are plotted in Figure 3.1b. The details on the construction of our simulation setup are provided in the Supplementary Material.

We consider an experiment, replicated 500 times. In *Experiment*, $N \in \{40, 100, 200\}$, $\mathfrak{m} \in \{40, 100, 200\}$, and $(1-p)\mathfrak{m} \leq M_i \leq (1+p)\mathfrak{m}$ with $p = 0.2$.

For the presmoothing, we use the `locpoly` function in `R`. Given that most of the local regularity estimates are between 0 and 1, our mean and covariance functions estimators are built with the NW smoother. Moreover, the value k_0 in (3.15) is set equal to 2, and we use the biweight kernel $K(t) = (15/16)\left(1-t^2\right)^2 \mathbf{1}_{[-1,1]}(t)$. The estimates $\hat{\alpha}_t$ and the estimates of the mean and covariance functions are obtained using the same data. That means we did not use a *learning sample* for $\hat{\alpha}_t$. An implementation of the method used in the four experiments is available as a `R` package on Github at the URL adress: https://github.com/StevenGolovkine/funestim.

3.5.3 Mean estimation

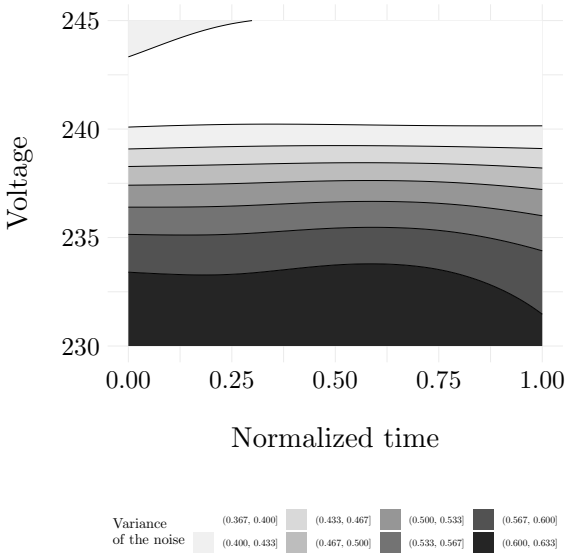
Concerning the estimation of the mean, the estimates $\hat{\alpha}_t$ are computed using (3.12), on a uniform grid of 50 points t_2 between 0.05 and 0.95, with $t_3 - t_1 = \Delta_*/2 = \exp(-\log^{1/2}(\hat{\mathfrak{m}}))$. For each value of the 50 estimates $\hat{\alpha}_t$, we compute the optimal bandwidths h_μ^* by minimization with respect to h over a logarithmic grid of 151 points.

Our mean estimator is compared to that of [19], denoted $\widehat{\mu}_{CY}$, and [152], denoted $\widehat{\mu}_{ZW}$. To compute $\widehat{\mu}_{CY}$, we use `smooth.splines` in `R`, with the $M_1 + \ldots + M_N$ data points $(Y_m^{(n)}, T_m^{(n)})$. To obtain $\widehat{\mu}_{ZW}$, we use the `R` package `fdapace`, see [24]. To compare the accuracy of the estimators, we use the ISE risk with respect to the target. For any f and g real-valued functions defined on $[0,1]$, the ISE is defined as

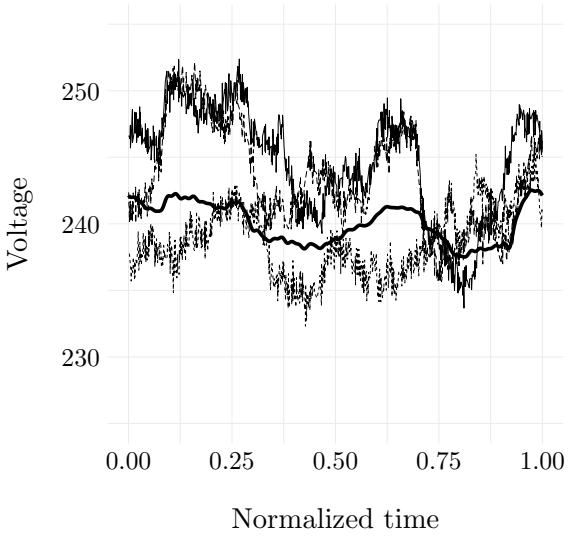
$$\mathrm{ISE}(f, g) = \|f - g\|^2 = \int_{[0,1]} \{f(t) - g(t)\}^2 dt.$$

We approximate the integral using the mean estimates on a uniform grid of 101 points and the trapezoidal rule. For each configuration N, $\mathfrak{m}$, p, and each of the 500 samples, we compute the ISE*s* with respect to the infeasible $\widetilde{\mu}$, and the ISE*s* with respect to the mean function μ used for generating the samples. The 101 bandwidth values used for our estimator are obtained from the 50 optimal bandwidths h_μ^* by linear interpolation.

The results obtained in *Experiment* are plotted in the Figure 3.2, on a logarithmic scale. Our mean function estimator reveals good performance. It provides a much more



(a) The conditional variance function $\sigma^2(t, x)$



(b) The mean function $\mu(t)$ (bold line) and three randomly generated noisy trajectories

Figure 3.1: The simulation setup in *Experiment*

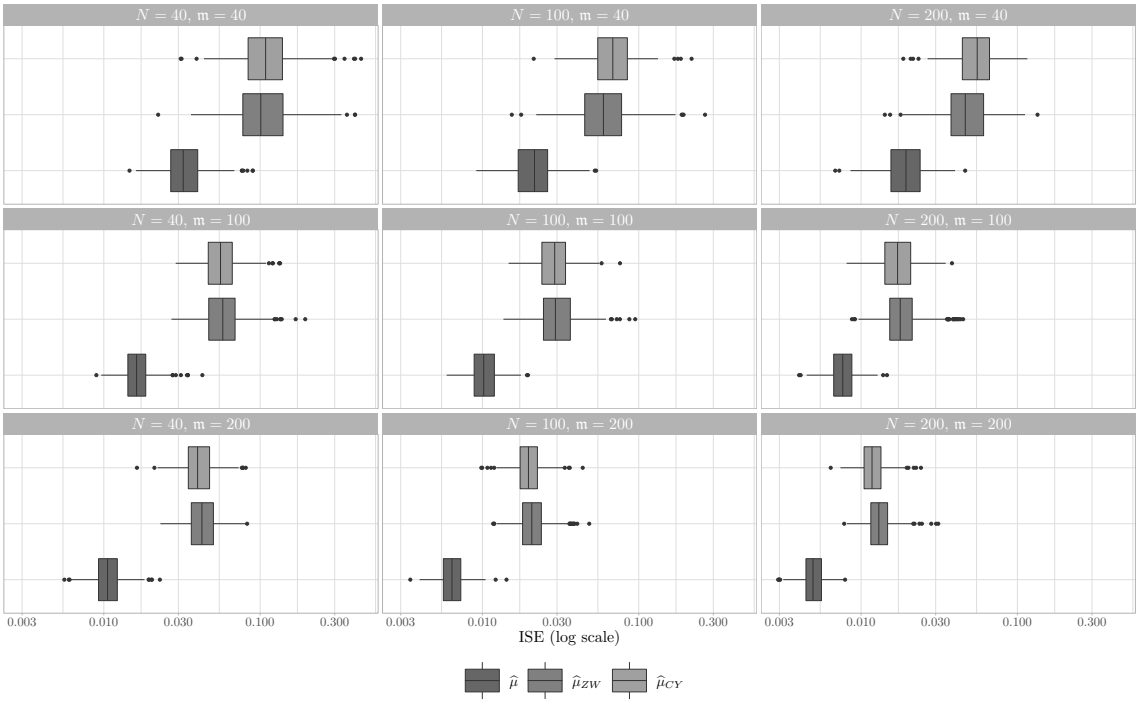
accurate estimate of the infeasible empirical mean function $\widetilde{\mu}$. When compared to the true mean function, except the cases $N \in \{100, 200\}$ and $\mathfrak{m} = 40$, our estimator outperforms the competitors. In that cases, our estimator, $\widehat{\mu}_{CY}$ and $\widehat{\mu}_{ZW}$ have similar performance. The fact that the advantage of our estimator wanes in these cases is explained by the poorer performance of the infeasible empirical mean function.

3.5.4 Covariance estimation

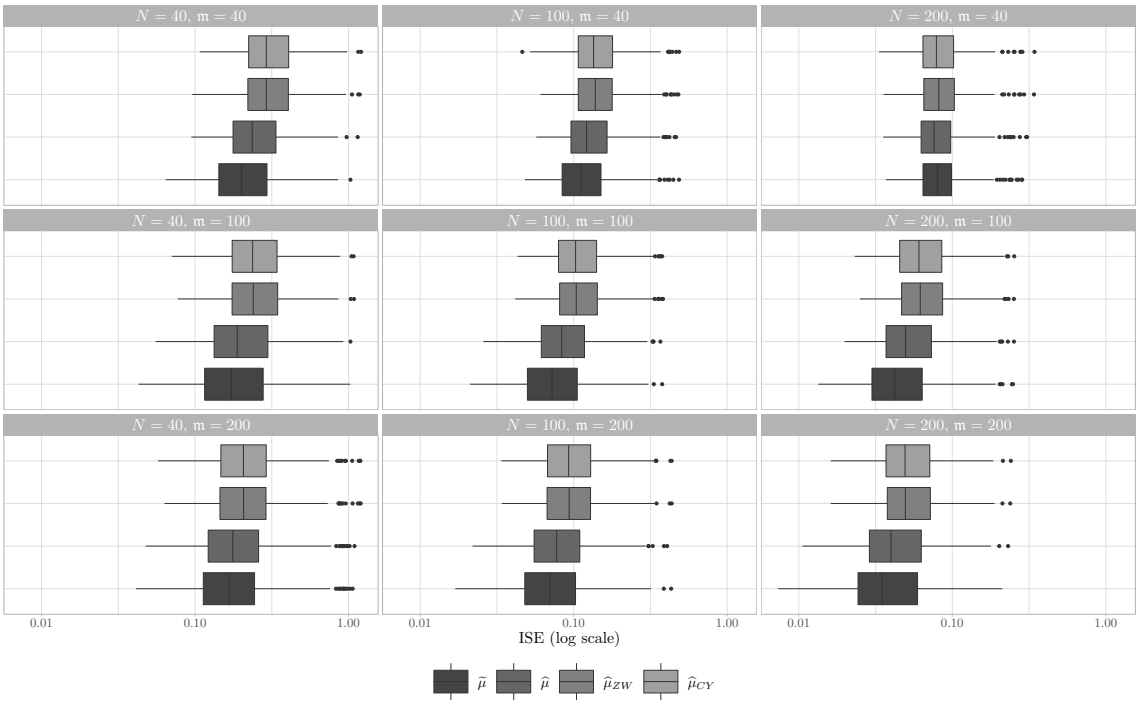
Concerning the estimation of the covariance, the estimates $\hat{\alpha}_t$ are computed using (3.12), on a uniform grid of 10 points t_2 between 0.05 and 0.95, with

$$t_3 - t_1 = \Delta_*/2 = \exp(-\log^{1/2}(\hat{\mathfrak{m}})).$$

For each values s, t in the grid, we compute the optimal bandwidths $h^*_\Gamma(s, t)$ by minimization over a logarithmic grid of 41 points between 0.01 and 0.1. Our covariance estimator is compared to the ones from [18], denoted $\widehat{\Gamma}_{CY}$, and from [152], denoted $\widehat{\Gamma}_{ZW}$. We compute $\widehat{\Gamma}_{CY}$ using the `R` package **`ssfcov`**, see [18]. For $\widehat{\Gamma}_{ZW}$, we use the `R` package **`fdapace`**, see



(a) ISE with respect to the empirical mean $\tilde{\mu}$ in each simulated sample



(b) ISE with respect to the true mean function μ

Figure 3.2: Results from *Experiment* on the log-scale

[24]. The risk we consider is defined as

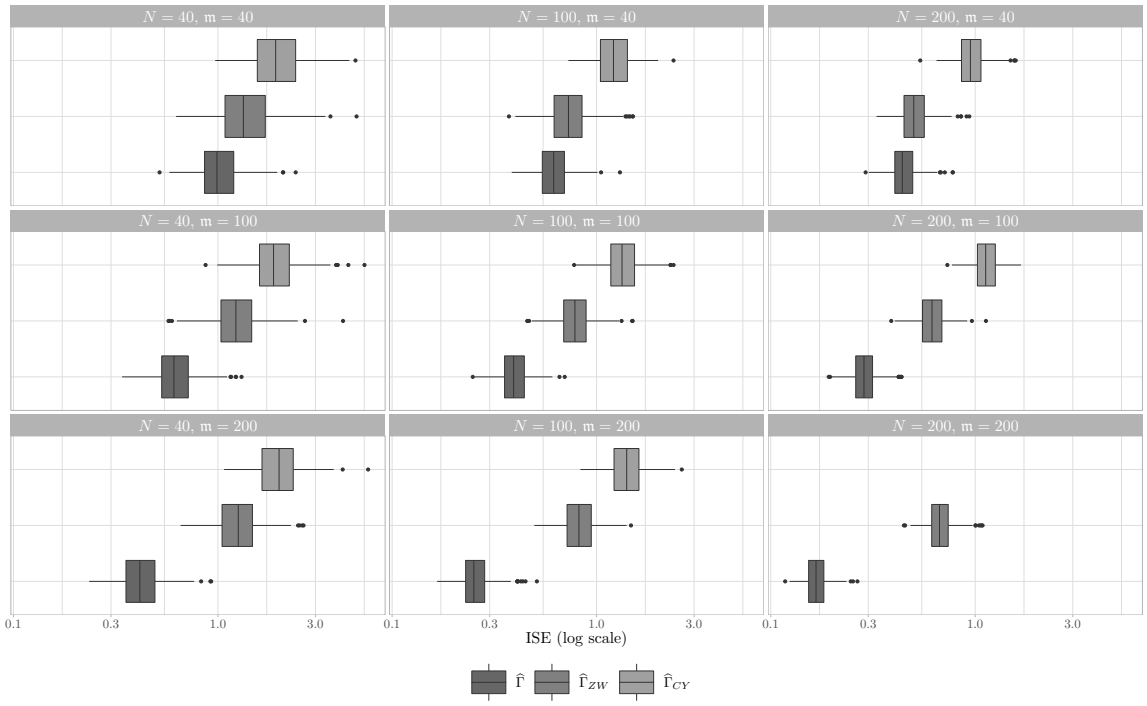
$$\mathrm{ISE}(f,g) = \|f-g\|^2 = \int_{[0,1]}\int_{[0,1]}\{f(s,t)-g(s,t)\}^2 dsdt,$$

approximated by the trapezoidal rule applied with a uniform grid of 101×101 points. The 101×101 bandwidth values used for our estimator are obtained from the 10×10 optimal bandwidths h_Γ^* by linear interpolation. For each configuration N, $\mathfrak{m}$, p, and each of the 500 samples, we compute the ISE*s* with respect to the infeasible $\widetilde{\Gamma}$, and the ISE*s* with respect to the covariance function Γ used for generating the samples.

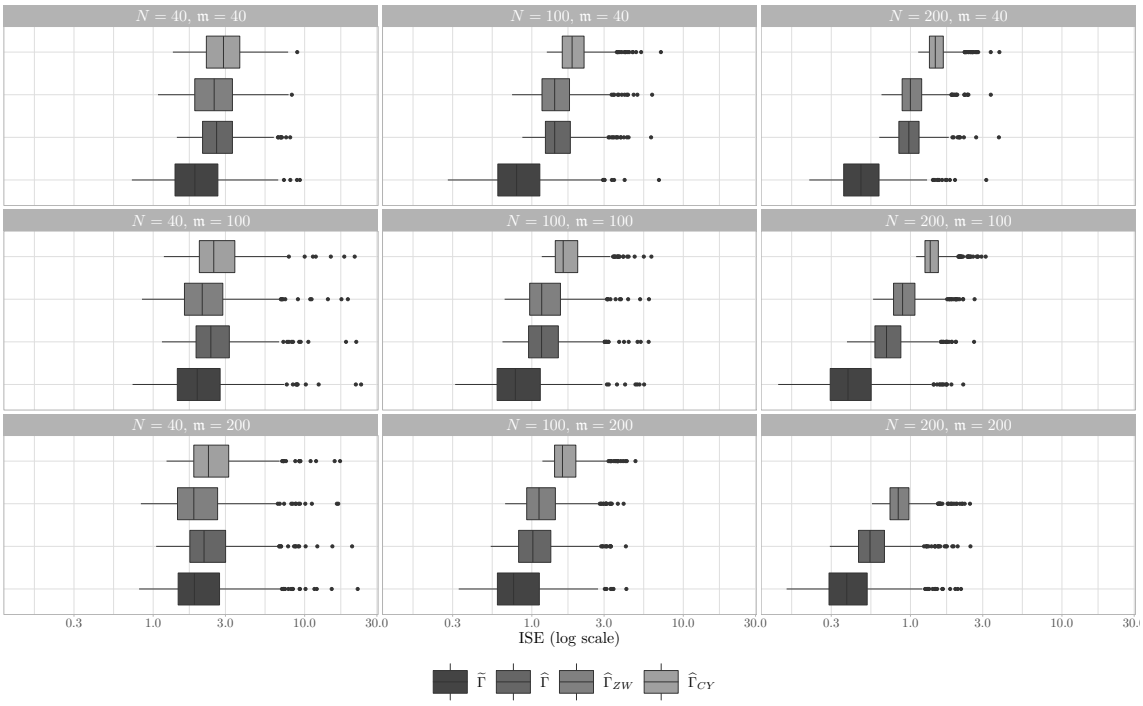
The results obtained in *Experiment* are plotted in the Figure 3.3, on a logarithmic scale. We were not able to calculate $\widehat{\Gamma}_{CY}$ when $M = 200$ and $N = 200$, and therefore some cases are not reported. Our estimator provides the most accurate approximation of $\widetilde{\Gamma}$. Our estimator and $\widehat{\Gamma}_{ZW}$ show better accuracy for estimating Γ in all cases considered. Meanwhile, our estimator performs similarly or slightly better than $\widehat{\Gamma}_{ZW}$. The advantage of our approach increases with N. Let us point out the clear advantage for our estimator in terms of computation times. For instance, with $N = 100$ and $M = 100$, our covariance estimator is computed about three times faster that of [152] (with fixed bandwidth) and about thirty times faster than that of [18].

3.6 Discussion and conclusions

We propose new nonparametric estimators for the mean and covariance function. They are built using a novel 'smoothing first, then estimate' strategy based on univariate local polynomials. The main novelty comes from the fact that the optimal bandwidths for the local polynomials are selected by minimization of suitable penalized quadratic risks. The penalized risks for the mean and the covariance functions are quite similar, could be easily built from data and be optimized on a grid of bandwidths. What distinguishes them from the usual sum between the squared bias and the variance, is a penalty for the fact that not all the curves have enough observation points to be included in the final estimator. Removing curves from the nonparametric estimators of the mean and covariance functions is an aspect which characterizes practically all smoothing-based approaches. Indeed, to entirely benefit from the replication feature of functional data, one has to determine the amount of smoothing for the mean and covariance estimation using all the curves. In this case some curves could present too few observation points and thus will be dropped. This is



(a) ISE with respect to the empirical covariance $\widetilde{\Gamma}$ in each simulated sample



(b) ISE with respect to the true covariance function Γ

Figure 3.3: Results from *Experiment* on the log-scale

more likely to happen in the so-called sparse regime. To our best knowledge, our bandwidth choice procedure is the first attempt to explicitly account for this aspect. We thus build estimators which achieve the optimal rates of convergence in a completely adaptive, data-driven way. The theoretical results are derived under very mild conditions. In particular, the curves could be observed with heteroscedastic errors at discrete observations points. These points could be common to all curves or they could change randomly from one curve to another. In the case of the common observation points, our procedure automatically chooses between smoothing and interpolation, the latter being known to be rate optimal, but is not necessarily the best solution with finite samples.

Our nonparametric estimation approach relies on a probabilistic concept of local regularity for the sample paths of the process generating the curves. In some common examples, this local regularity is related to the polynomial decrease rate of the eigenvalues of the covariance operator, a characteristic of the data generating process widely used in the literature and usually supposed to be known. The local regularity also determines the regularity of the trajectories, the usual concept used in nonparametric regression. It is well-known that the optimal rates, in the minimax sense, for estimating the mean and covariance functions depend on the regularity of the trajectories. Moreover, the so-called sparse and dense regimes, commonly invoked in functional data analysis, are defined using the regularity of the trajectories, which usually is supposed to be known. We therefore propose a novel simple estimator of the local regularity of the process and we use it to build our penalized quadratic risk. Applied to real data, the local regularity estimator reveals that the regularity of the trajectories could be quite far from what is usually assumed in the existing theoretical contributions.

Our method performs quite well in simulations and outperforms the main competitors when the mean and covariance functions have a regularity close to that of the trajectories. The reason is that, in some sense, our nonparametric estimators are as close as possible to the empirical mean and covariance, respectively, which are the ideal estimators if the trajectories were observed at any point without error. In the case where the mean and covariance function are smoother than the trajectories, our penalized quadratic risk should be built using the mean or covariance functions' regularity instead of the trajectories' regularity. However, for now, the estimation of the regularity of the mean or covariance function remains an open problem.

Appendix

3.A Details on the definition (3.21)

To explain our empirical risk bound $\mathcal{R}_\mu(t;h)$ defined in (3.21), let

$$\widetilde{\mu}_W(t;h) = \frac{1}{\mathcal{W}_N(t;h)} \sum_{i=1}^{N} w_i(t;h) X_t^{(i)},$$

be the unfeasible estimator of $\mu(\cdot)$ using only the curves for which $\widehat{X}_t^i$ is well-defined. Using the definitions, see (3.7), we then have,

$$\begin{aligned}
&\mathbb{E}_{M,T}\left[\{\widetilde{\mu}_N(t) - \widehat{\mu}_N(t;h)\}^2\right] \\
&\quad = \mathbb{E}_{M,T}\left[\left\{\widetilde{\mu}_N(t) - \widetilde{\mu}_W(t;h) - \frac{1}{\mathcal{W}_N(t;h)} \sum_{i=1}^{N} w_i(t;h)\left(B_t^{(i)} + V_t^{(i)}\right)\right\}^2\right] \\
&\quad \leq 2\mathbb{E}_{M,T}\left[\{\widetilde{\mu}_N(t) - \widetilde{\mu}_W(t;h)\}^2\right] + 2\mathbb{E}_{M,T}\left[\left\{\frac{1}{\mathcal{W}_N(t;h)} \sum_{i=1}^{N} w_i(t;h)\left(B_t^{(i)} + V_t^{(i)}\right)\right\}^2\right] \\
&\quad =: 2E_1 + 2E_2.
\end{aligned}$$

In the following, for simplicity, we write w_i and $\mathcal{W}_N$ instead of $w_i(t;h)$ and $\mathcal{W}_N(t;h)$, respectively. Since

$$\widetilde{\mu}_W(t;h) - \widetilde{\mu}_N(t) = \frac{1}{\mathcal{W}_N} \sum_{i=1}^{N} \left\{X_t^{(i)} - \mu(t)\right\}\left\{w_i - \frac{\mathcal{W}_N}{N}\right\},$$

and the trajectories are drawn independently, we have

$$E_1 = \frac{Var(X_t)}{\mathcal{W}_N^2} \sum_{i=1}^{N} \left\{w_i - \frac{\mathcal{W}_N}{N}\right\}^2 = Var(X_t)\left\{\frac{1}{\mathcal{W}_N} - \frac{1}{N}\right\}.$$

For E_2, let us first look at the bias part. By the arguments used in the proof of Proposition 1.13 in [135] and the Cauchy-Schwarz inequality, we have

$$\mathbb{E}_{M,T}\left[\left\{\frac{1}{\mathcal{W}_N} \sum_{i=1}^{N} w_i B_t^{(i)}\right\}^2\right] \leq \frac{1}{\mathcal{W}_N} \sum_{i=1}^{N} w_i \mathbb{E}_{M,T}\left[\left\{B_t^{(i)}\right\}^2\right]$$

$$
\begin{aligned}
&\leq \frac{1}{\lfloor \hat{\alpha}_t \rfloor !^2} \times \frac{1}{\mathcal{W}_N} \sum_{i=1}^{N} w_i \left\{ \sum_{m=1}^{M_i} \left| W_m^{(i)}(t) \right| \right. \\
&\qquad \left. \times \sum_{m=1}^{M_i} \mathbb{E}\left(\left\{ \nabla^{\lfloor \hat{\alpha}_t \rfloor} X^n(T_m^{(n)}) - \nabla^{\lfloor \hat{\alpha}_t \rfloor} X_t^{(i)} \right\}^2 \mid \mathcal{T}_{obs}^{(i)} \right) |W_m^{(i)}(t)| \right\} \\
&= \frac{L_\delta^2 h^{2\hat{\alpha}_t} \{1 + o_{\mathbb{P}}(1)\}}{\lfloor \hat{\alpha}_t \rfloor !^2} \times \frac{1}{\mathcal{W}_N} \sum_{i=1}^{N} w_i \left\{ \sum_{m=1}^{M_i} \left| W_m^{(i)}(t) \right| \times \sum_{m=1}^{M_i} \left| \frac{T_m^{(n)} - t}{h} \right|^{2\hat{\alpha}_t} |W_m^{(i)}(t)| \right\} \\
&=: \frac{L_\delta^2 h^{2\hat{\alpha}_t}}{\lfloor \hat{\alpha}_t \rfloor !^2} \times \overline{C}_1(t; h, 2\hat{\alpha}_t) \times \{1 + o_{\mathbb{P}}(1)\}.
\end{aligned}
$$

For the first equality we use the condition (H2) on the event $\{\lfloor \hat{\alpha}_t \rfloor = \delta\}$. By Theorem 4, this event has an exponentially small probability. When the kernel function is supported on $[-1, 1]$, we could use the bound

$$
\mathbb{E}_{M,T}\left[\left\{ \frac{1}{\mathcal{W}_N} \sum_{i=1}^{N} w_i B_t^{(i)} \right\}^2 \right] \leq \frac{L_\delta^2 h^{2\hat{\alpha}_t} \{1 + o_{\mathbb{P}}(1)\}}{\lfloor \hat{\alpha}_t \rfloor !^2} \times \frac{1}{\mathcal{W}_N} \sum_{i=1}^{N} w_i \left\{ \sum_{m=1}^{M_i} \left| W_m^{(i)}(t) \right| \right\}^2.
$$

In the case $\lfloor \hat{\alpha}_t \rfloor = 0$, with the Nadaraya-Watson estimator, we could use the more refined bound

$$
\begin{aligned}
&\mathbb{E}_{M,T}\left[\left\{ \frac{1}{\mathcal{W}_N} \sum_{i=1}^{N} w_i B_t^{(i)} \right\}^2 \right] \\
&\qquad \leq \frac{L_0^2 h^{2\hat{\alpha}_t} \{1 + o_{\mathbb{P}}(1)\}}{\lfloor \hat{\alpha}_t \rfloor !^2} \times \frac{1}{\mathcal{W}_N} \sum_{i=1}^{N} w_i \left\{ \sum_{m=1}^{M_i} \left| \frac{T_m^{(n)} - t}{h} \right|^{2\hat{\alpha}_t} W_m^{(i)}(t) \right\} \\
&\qquad \approx \frac{L_0^2 h^{2\hat{\alpha}_t} \{1 + o_{\mathbb{P}}(1)\}}{\lfloor \hat{\alpha}_t \rfloor !^2} \times \int |u|^{2\hat{\alpha}_t} K(u) du.
\end{aligned}
$$

Using the equivalent kernels idea, see section 3.2.2 in [45], the bound on the last line of the last display could be used in the case of local linear estimators. To complete the bound for E_2, note that by construction, $\mathbb{E}_{M,T}\left\{ V_t^{(i)} B_t^{(i)} \right\} = 0$ and

$$
\mathbb{E}_{M,T}\left\{ V_t^{(i)} B_t^{(j)} \right\} = \mathbb{E}_{M,T}\left\{ V_t^{(i)} V_t^{(j)} \right\} = 0, \quad \forall 1 \leq i \neq j \leq N.
$$

Up to negligible terms, we can then write

$$
E_2 \leq \mathbb{E}_{M,T}\left[\left\{ \frac{1}{\mathcal{W}_N} \sum_{i=1}^{N} w_i B_t^{(i)} \right\}^2 \right] + \frac{1}{\mathcal{W}_N^2} \sum_{i=1}^{N} w_i \mathbb{E}_{M,T}\left[\left\{ V_t^{(i)} \right\}^2 \right]
$$

$$\leq h^{2\hat{\alpha}_t} \frac{L_\delta^2}{\lfloor \hat{\alpha}_t \rfloor !^2} \overline{C}_1(t; h, 2\hat{\alpha}_t) + \frac{\sigma_{\max}^2}{\mathcal{W}_N^2} \sum_{i=1}^{N} w_i \left\{ \max_m \left| W_m^{(i)}(t; h) \right| \times \sum_{m=1}^{M_i} \left| W_m^{(i)}(t; h) \right| \right\}$$
$$= h^{2\hat{\alpha}_t} \frac{L_\delta^2}{\lfloor \hat{\alpha}_t \rfloor !^2} \overline{C}_1(t; h, 2\hat{\alpha}_t) + \frac{\sigma_{\max}^2}{\mathcal{W}_N^2} \sum_{i=1}^{N} w_i \frac{c_i(t; h)}{\mathcal{N}_i(t; h)},$$

where $\sigma_{\max}$ is a bound for the function $\sigma(t, x)$ in (3.5).

3.B Proofs

Proof of Theorem 5. For simplicity, $\widehat{\mu}_N^*$ is built with $k_0 = 1$ and the uniform kernel. The lines below adapt to other choices of k_0 and $K(\cdot)$ at the price of more involved technical arguments. First, let us prove that

$$\frac{1}{\mathcal{W}_N(t; h)} - \frac{1}{N} \leq \max \left\{ h^{2\alpha_t}, \, \mathcal{N}_\mu^{-1}(t; h) \right\} O_{\mathbb{P}}(1), \tag{3.31}$$

provided $N\mathfrak{m}h \to \infty$ and $h \to 0$. Note that if $\mathcal{W}_N(t; h) = 0$, then necessarily $\mathcal{N}_\mu(t; h) = 0$. The property (3.31) is implied by the following ones: there exist two constants $\mathfrak{c}_1, \mathfrak{c}_2 > 0$ such that

$$\frac{1}{\mathcal{N}_\mu(t; h)} \geq \frac{\mathfrak{c}_1 \{1 + o_{\mathbb{P}}(1)\}}{N\mathfrak{m}h}, \tag{3.32}$$

and

$$\frac{1}{\mathcal{W}_N(t; h) + 1} \leq \max \left\{ \frac{1}{N+1}, \, \frac{\mathfrak{c}_2 \{1 + o_{\mathbb{P}}(1)\}}{N\mathfrak{m}h} \right\}. \tag{3.33}$$

Indeed, the latter two properties imply

$$\frac{1}{\mathcal{W}_N(t; h) + 1} - \frac{1}{N+1} = \max \left\{ 0, \, \frac{1}{N} \left(\frac{1}{\mathfrak{m}h} - 1 \right) \right\} O_{\mathbb{P}}(1)$$
$$\leq \max \left\{ h^{2\alpha_t}, \, \frac{1}{\mathcal{N}_\mu(t; h)} \right\} O_{\mathbb{P}}(1),$$

and from this we obtain (3.31) because

$$\{\mathcal{W}_N(t; h) + 1\}^{-1} - \{N+1\}^{-1} = \left\{ \mathcal{W}_N^{-1}(t; h) - N^{-1} \right\} \frac{N\{N+1\}}{\mathcal{W}_N(t; h)\{\mathcal{W}_N(t; h) + 1\}}.$$

To justify (3.32) and (3.33), we omit the arguments t and h. For instance, we write $\mathcal{W}_N$ and $\mathcal{N}_\mu$ instead of $\mathcal{W}_N(t; h)$ and $\mathcal{N}_\mu(t; h)$, respectively.

Using the fact that the harmonic mean is less than or equal to the mean we obtain

$$\frac{1}{\mathcal{N}_\mu(t;h)} \geq \frac{c_i}{\sum_{i=1}^N w_i \mathcal{N}_i},$$

with $c_i = c_i(t;h)$ and $\mathcal{N}_i = \mathcal{N}_i(t;h)$ defined in (3.17) and (3.19), respectively. In the case of $\widehat{\alpha}_t < 1$, for all i we have $c_i \equiv 1$. To justify (3.32) it suffices to prove that there exists a sequence of variables C_N such that almost surely $0 < \liminf_N C_N < \limsup_N C_N < \infty$, and

$$C_N\{1 + o_{\mathbb{P}}(1)\} = \frac{\sum_{i=1}^N w_i \mathcal{N}_i}{N\mathfrak{m}h}. \tag{3.34}$$

For this purpose, let us notice that in the case of an NW estimator with a uniform kernel, when $k_0 = 1$,

$$\sum_{i=1}^N w_i \mathcal{N}_i = \sum_{i=1}^N \sum_{m=1}^{M_i} \mathbf{1}\{|T_m^{(n)} - t| \leq h\}.$$

Thus $\sum_{i=1}^N w_i \mathcal{N}_i$ is a binomial random variable with $M_1 + \ldots + M_N$ trials and success probability

$$p(h) = p(h;t) = \int_{t-h}^{t+h} f_T(s)ds \approx 2f_T(t)h,$$

provided f_T, the density of the $T_m^{(n)}$, is continuous at t. Since $N\mathfrak{m}h \to \infty$, by Bernstein's inequality the property (3.34) holds true with

$$C_N = 2f_T(t)\frac{M_1 + \ldots + M_N}{N\mathfrak{m}}.$$

Condition (3.23) guarantees that almost surely C_N stays away from zero and infinity. The property (3.32) follows.

For justifying (3.33), let us first recall that by definition, $\mathcal{W}_N(t;h) \leq N$. Thus it remains to show that

$$\frac{1}{\mathcal{W}_N(t;h) + 1} \leq \frac{\mathfrak{c}_2\{1 + o_{\mathbb{P}}(1)\}}{N\mathfrak{m}h}.$$

For this purpose, it suffices to show that there exists a positive constant $\mathfrak{c}_{\mathcal{W}} > 0$ such that

$$\min\{N, \mathfrak{c}_{\mathcal{W}} N\mathfrak{m}h\{1 + o_{\mathbb{P}}(1)\}\} \leq \mathcal{W}_N(t;h). \tag{3.35}$$

Let $\mathbb{P}_M(\cdot) = \mathbb{P}(\cdot \mid M_i, 1 \leq i \leq N)$. We now note that

$$p_i = p_i(t) := \mathbb{P}_M(w_i = 1) = 1 - [1 - p(h)]^{M_i}.$$

Next, note that the variable $\mathcal{W}_N(t;h)$ is a sum of N independent Bernoulli variables with probabilities p_i. In the case where $\mathfrak{m}h \geq \underline{c} > 0$ for some constant $\underline{c}$, since $\{1-p(h)\}^{-1/p(h)} > e$ for any $h < 1/2$, we can write

$$p_i = 1 - [1-p(h)]^{M_i} \geq 1 - [1-p(h)]^{c_L\mathfrak{m}} > 1 - \exp(-c_L\mathfrak{m}p(h)) \\ \approx 1 - \exp(-2f_T(t)c_L\mathfrak{m}h) \geq 1 - \exp(-2f_T(t)c_L\underline{c}),$$

and the approximations are valid as soon as $h \to 0$. Thus, in the case where $\mathfrak{m}h \geq \underline{c} > 0$, $\liminf p_i > 0$. Then, Bernstein's inequality yields the property (3.35). More precisely, $\mathcal{W}_N(t;h)$ will increase at a rate at least as fast as $[1-\exp(-2f_T(t)c_L\underline{c})]N$. In the case $\mathfrak{m}h \to 0$ we have

$$p_i = \mathbb{P}_M(w_i = 1) \approx 1 - \exp(-M_i p(h)) \approx M_i p(h) \geq 2c_L f_T(t)\mathfrak{m}h.$$

Again, Bernstein's inequality guarantees that

$$\frac{\mathcal{W}_N(t;h)}{p_1 + \ldots + p_N} = 1 + o_{\mathbb{P}}(1).$$

Since $p_1 + \ldots + p_N \geq 2c_L f_T(t) \times N\mathfrak{m}h$, condition (3.35) follows with $\mathfrak{c}_{\mathcal{W}} = 2c_L f_T(t)$.

Finally, to complete the proof in the independent design case, it suffices first to notice that from above, we have

$$\max\left\{h^{2\alpha_t},\ \mathcal{N}_\mu^{-1}(t;h)\right\} = O_{\mathbb{P}}\left(h^{2\alpha_t} + (N\mathfrak{m}h)^{-1}\right),$$

which is minimized by h with the rate $(N\mathfrak{m})^{-2\alpha_t/\{2\alpha_t+1\}}$. Next, condition (3.22) guarantees that $\log(N\mathfrak{m})/\log(\mathfrak{m})$ is bounded, and thus $h^{2\hat{\alpha}_t} = h^{2\alpha_t}\{1 + o_{\mathbb{P}}(1)\}$ whenever $\hat{\alpha}_t - \alpha_t = o_{\mathbb{P}}(\log^{-1}(\mathfrak{m}))$. For the rate of $\widehat{\mu}_N^*(t) - \mu(t)$ we simply add the rate of $\widetilde{\mu}_N(t) - \mu(t)$.

With a common design $\mathcal{W}_N(t;h)$ can only take the values 0 or N^{-1}. Thus the penalty introduced by $\mathcal{W}_N(t;h)^{-1} - N^{-1}$ plays a different role. It constrains the bandwidth to be greater than or equal to the lengths of the intervals $[T_m^{(i)}, T_{m+1}^{(i)}]$ including t. By condition (3.24), this means that the rate of convergence of $\widehat{\mu}_N^*(t) - \widetilde{\mu}_N(t)$ could not be faster than $O_{\mathbb{P}}\left(\mathfrak{m}^{-\,2\alpha_t}\right)$. This aspect is automatically included in the definition of $\mathcal{R}_\mu(t;h)$ because, under the constraint $\mathfrak{m}h \geq c_L/C_U$,

$$O_{\mathbb{P}}\left(\max\left\{h^{2\alpha_t},\ \mathcal{N}_\mu^{-1}(t;h)\right\}\right) = O_{\mathbb{P}}\left(\max\left\{h^{2\alpha_t},\ (N\mathfrak{m}h)^{-1},\ N^{-1}\right\}\right) = O_{\mathbb{P}}\left(\mathfrak{m}^{-\,2\alpha_t}\right).$$

Finally, α_t can be replaced by $\hat{\alpha}_t$ using the arguments from the independent design case. □

Proof of equation (3.29). To prove the rate of $\widetilde{D}(\mathfrak{d})$, we use the assumptions: $\sup_{t\in\mathcal{T}} \mathbb{E}(X_t^4) < \infty$ and there exists a constant c such that

$$\mathbb{E}(\{X_s - X_t\}^4) \leq c\mathbb{E}^2(\{X_s - X_t\}^2), \quad \forall s,t \in \mathcal{T}.$$

Omitting the integration domain, we can now write

$$\begin{aligned}
\mathbb{E}[\widetilde{D}(\mathfrak{d})] &\leq 2\iint \mathbb{E}\left[\left(\frac{1}{N}\sum_{i=1}^{N}\left\{X_s^{(i)}X_t^{(i)} - X_{\frac{s+t-\mathfrak{d}}{2}}^{(i)}X_{\frac{s+t+\mathfrak{d}}{2}}^{(i)}\right\}\right)^2\right] dsdt \\
&\quad + 2\iint \mathbb{E}\left[\left(\left\{\frac{1}{N}\sum_{i=1}^{N}X_s^{(i)}\right\}\left\{\frac{1}{N}\sum_{i=1}^{N}X_t^{(i)}\right\} - \left\{\frac{1}{N}\sum_{i=1}^{N}X_{\frac{s+t-\mathfrak{d}}{2}}^{(i)}\right\}\left\{\frac{1}{N}\sum_{i=1}^{N}X_{\frac{s+t+\mathfrak{d}}{2}}^{(i)}\right\}\right)^2\right] dsdt \\
&= 2\widetilde{D}_1 + 2\widetilde{D}_2.
\end{aligned}$$

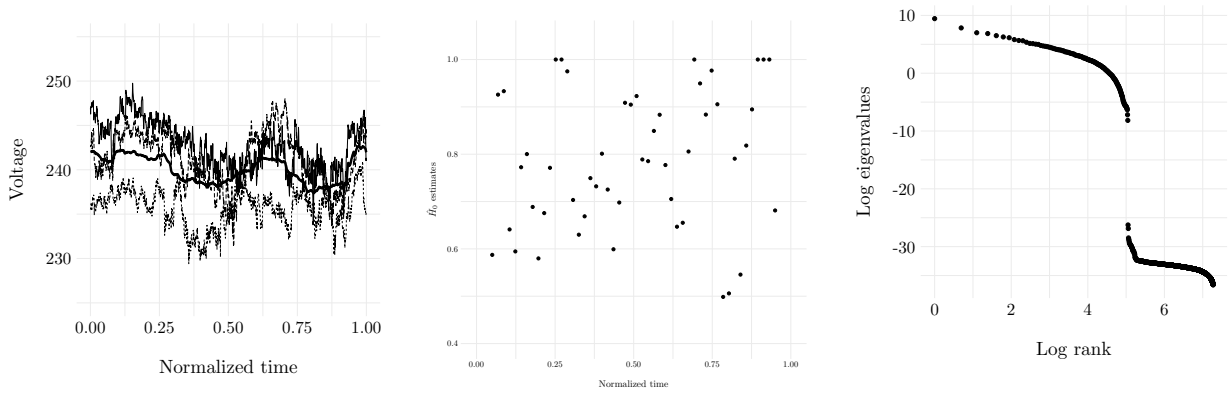
By the Cauchy-Schwarz inequality,

$$\mathbb{E}\left[\left\{X_s^{(i)}X_t^{(i)} - X_{\frac{s+t-\mathfrak{d}}{2}}^{(i)}X_{\frac{s+t+\mathfrak{d}}{2}}^{(i)}\right\}^2\right] = O\left(\mathfrak{d}^{2\alpha}\right),$$

where α is the local regularity on $\mathcal{T}_k^\circ$ to which s, t, $(s+t+\mathfrak{d})/2$ and $(s+t-\mathfrak{d})/2$ eventually belong. We deduce that $\widetilde{D}_1 = O\left(\mathfrak{d}^{2\alpha+1}\right)$. On the other hand, by repeatedly application of the Cauchy-Schwarz inequality and the moment conditions imposed above,

$$\begin{aligned}
\widetilde{D}_2 &\leq 2\iint \mathbb{E}^{1/2}\left[\left(\frac{1}{N}\sum_{i=1}^{N}\{X_s^{(i)} - X_{\frac{s+t-\mathfrak{d}}{2}}^{(i)}\}\right)^4\right]\mathbb{E}^{1/2}\left[\left(\frac{1}{N}\sum_{i=1}^{N}X_t^{(i)}\right)^4\right] dsdt \\
&\quad + 2\iint \mathbb{E}^{1/2}\left[\left(\frac{1}{N}\sum_{i=1}^{N}X_{\frac{s+t-\mathfrak{d}}{2}}^{(i)}\right)^4\right]\mathbb{E}^{1/2}\left[\left(\frac{1}{N}\sum_{i=1}^{N}\{X_t^{(i)} - X_{\frac{s+t+\mathfrak{d}}{2}}^{(i)}\}\right)^4\right] dsdt \\
&= O\left(\mathfrak{d}^{2\alpha+1}\right).
\end{aligned}$$

Gathering facts, we deduce that $\widetilde{D}(\mathfrak{d}) = O_{\mathbb{P}}(\mathfrak{d}^{2\alpha+1})$. □



(a) Empirical mean and sample curves from the data

(b) Local regularity estimates

(c) The logarithm of the eigenvalues onto the log of their rank

Figure 3.4: Extracted Household Active Power Consumption data: voltage curves

3.C Additional simulation results

In this section we recall and add details on the construction of the simulation setup. Moreover, we provide more details on the implementation of our estimators and present results obtained from additional experiments.

3.C.1 Description of the real data set used to build the simulations

For the simulation experiments reported in this article we build mean and covariance functions, as well as conditional variance function, using a real data set. More precisely, our simulation study is based on the Household Active Power Consumption dataset which was sourced from the UC Irvine Machine Learning Repository (https://archive.ics.uci.edu/ml/datasets/Individual+household+electric+power+consumption). An implementation of the simulation methods is available as a `R` package on Github at the URL address https://github.com/StevenGolovkine/simulater.

3.C.2 Construction of the simulation design

For building the mean function, we consider the following model

$$\mu(t) = \beta_0 t + \sqrt{2} \sum_{k=1}^{50} \{\beta_{1,k} \cos(2k\pi t) + \beta_{2,k} \sin(2k\pi t)\}, \quad t \in [0,1]. \tag{3.36}$$

The coefficients β are estimated by LASSO regression with the outcomes given by the 1440 values of the empirical mean of the 708 curves and the covariates obtained with the uniform grid of 1440 points in $[0,1]$. The penalized regression was done using the function `glmnet` from the `R` package `glmnet`. The regularity of the mean function is controlled using the penalty parameter λ. The mean function obtained with $\lambda = \exp(-5.5)$, which was used in *Experiments*, is plotted with plain line in Figure 3.1b.

To build the covariance function for the simulation design, we first smooth each curve using the function `smooth.splines` in `R` software, and we compute the empirical covariance of the sample of smoothed curves on the latice grid with 1440×1440 points in $[0,1]^2$. We next fit a two dimensional local linear kernel smoother to this empirical covariance with a bandwidth 0.01 using the function `Lwls2D` from the `R` package `fdapace`. We use this model to estimate a smoothed covariance on a 2880×2880 equispaced grid. We refined the grid on points in $[0,1]^2$ to obtained a higher numerical precision for the target quantities. Then, after computing the eigenvalues of this smoothed covariance, we fit a linear model onto the logarithm of the 6th to the 105th eigenvalues onto the logarithm of their rank (the R^2 is equal to 0.9373). The slope is -2.4118 (standard error equal to 0.063), which corresponds to a local regularity equal to 0.7059=(2.4118-1)/2. We use this model to predict the eigenvalues from 6 to 2880. The covariance matrix we use for simulations is then that rebuilt using this new set of 2880 eigenvalues and the eigenfunctions of the smoothed covariance. The eigenvalues of the smoothed covariance are plotted in Figure 3.4. The eigenvalues used to fit the linear model correspond to the first almost linear part, after removing the first five eigenvalues, which resulted in an improved fit.

To estimate the conditional variance of the noise $\sigma^2(t,x)$, for each $1 \leq i \leq 708$, we first fit a smoothing spline curve to the subset of every second voltage measurement, that is 720 outcome values. Next we use the fitted splines to predict the other 720 voltage measurements on each curve. We compute the squares of the residuals from all the curves, that is 720 times 708 squared residuals. We finally define $\sigma^2(t,x)$ as the fitted function obtained from a generalized additive model (GAM) applied with all the squared residuals and using a full tensor product smooth on $[0,1] \times [\min_i X^{(i)}; \max_i X^{(i)}]$. See Wood [143]. The model allows us to estimate the conditional variance of the noise at every point in $[0,1] \times [\min_i X^{(i)}; \max_i X^{(i)}]$, as it will be needed with an independent design setup. The plot of the fitted function $\sigma^2(t,x)$ for $(t,x) \in [0,1] \times [230,245]$, is given on Figure 3.1a.

To simulate our data, we first fix N and $\mathfrak{m}$ and $p \in (0,1)$. Next, for each $1 \leq i \leq N$, we generate M_i according to a uniform distribution on the interval $[(1-p)\mathfrak{m}, (1+p)\mathfrak{m}]$. Given

M_i, we draw the observations times $T_m^{(n)} \in [0,1]$, $1 \leq m \leq M_i$ according to a uniform distribution on $[0,1]$. Next, we compute the mean function at the times $T_m^{(n)}$ using the model (3.36). We subset the 2880×2880 covariance matrix to build the $M_i \times M_i-$covariance matrix corresponding to the $T_m^{(n)}$. If the considered point (s,t) does not exists in the large matrix, we use the closest one. We use the function `mvrnorm` from the `R` package `MASS` to generate the M_i measurements $X^n(T_m^{(n)})$. Finally, we generate the error term $\varepsilon_m^{(n)} = \sigma(T_m^{(n)}, X^n(T_m^{(n)}))u_m^{(n)}$, with $u_m^{(n)}$ a standard Gaussian variable and the function $\sigma(\cdot,\cdot)$ fitted using to the GAM, and define $Y_m^{(n)} = X^n(T_m^{(n)}) + \varepsilon_m^{(n)}$. A random sample of three curves generated according to this type of simulation setup, obtained with $\mathfrak{m} = 200$ and $p = 0.2$, are plotted in Figure 3.1b.

Chapter 4

Clustering multivariate functional data using unsupervised binary trees

Abstract: *We propose a model-based clustering algorithm for a general class of functional data for which the components could be curves or images. The random functional data realizations could be measured with error at discrete, and possibly random, points in the definition domain. After having recovered the true signals using an estimate of the local regularity of the process, the idea is to build a set of binary trees by recursive splitting of the observations. The number of groups are determined in a data-driven way. The new algorithm provides easily interpretable results and fast predictions for online data sets. Results on simulated datasets reveal good performance in various complex settings. The methodology is applied to the analysis of vehicle trajectories on a German roundabout. This article has been submit for publication in the* Annals of Applied Statistics *journal, reference in the field of applied statistics. Part of this study was presented in two international conferences:* StatMod2020 - Statistical Modeling with Applications *and* CMStatistics 2020.

Contents

4.1 Introduction 148

4.2 Model and methodology 150

4.2.1 Notion of multivariate functional data 150

4.2.2 A mixture model for curves 151

4.2.3 Multivariate Karhunen-Loève representation 154

4.3 Parameters estimation 157

4.3.1 Estimation of mean and covariance 158

4.3.2 Derivation of the MFPCA components 159
4.4 Multivariate functional clustering 160
4.4.1 Building the maximal tree 160
4.4.2 Joining step 164
4.4.3 Classification of new observations 165
4.5 Empirical analysis 166
4.5.1 Simulation experiments 169
4.5.2 Real data analysis: the rounD dataset 176
4.6 Extension to images 182
4.7 Conclusion 184
Appendix 184
4.A Proofs 185

4.1 Introduction

Motivated by a large number of applications ranging from sports to the automotive industry and healthcare, there is a great interest in modeling observation entities in the form of a sequence of possibly vector-valued measurements, recorded intermittently at several discrete points in time. Functional Data Analysis considers such data as being values on the realizations of a stochastic process, recorded with some error, at discrete random times. The purpose of FDA is to study such trajectories, also called curves or functions. See, *e.g.*, [108, 138, 73, 152, 64] for some recent references. The amount of such data collected grows rapidly as does the cost of their labeling. Thus, there is an increasing interest in methods that aim to identify homogeneous groups within functional datasets.

Clustering procedures for functional data have been widely studied in the last two decades, see for instance, [38, 39, 25, 83] and references therein. See also Bouveyron et al. [13] for a recent textbook. In particular, for Gaussian processes, Tarpey and Kinateder [129] show that the cluster centers found with k-means are linear combinations of the eigenfunctions from the fPCA. A discriminative functional mixture model is developed in [11] for the analysis of a bike sharing system from cities around the world.

Algorithms built to handle multivariate functional data have gained much attention in the last few years. Some of these methods are based on k-means algorithm with a specific

distance function adapted to multivariate functional data. See *e.g* [124, 133, 78, 149]). Kayano et al. [87] consider Self-Organizing Maps built on the coefficients of the curves into orthonormalized Gaussian basis expansion. Jacques and Preda [84] developed model-based clustering for multivariate functional data. In their model, they assume a cluster-specific Gaussian distribution for the principal component scores. The eigen-elements are estimated using an approximation of the curves into a finite dimensional functional space and are cluster specific. This model is also used in [29]. Schmutz et al. [120] have recently extended the previous model by modeling all principal components whose estimated variances are non-null. The underlying model for these methods usually consider only amplitude variations. Unlike others, Park and Ahn [102] present a specific model for functional data to consider phase variations. Finally, Traore et al. [134] propose a mix between dimension reduction and nonparametric approaches by deriving the envelope and the spectrum from the curves and have applied it to nuclear safety experiments.

In this contribution, we propose a clustering algorithm based on the recursive construction of binary trees to recover the groups. Our idea extends the Clustering using Unsupervised Binary Trees (CUBT) method [48, 53] to the functional setting, and we therefore call it functional Clustering using Unsupervised Binary Trees (fCUBT). At each node of the tree, a model selection test is performed, after expanding the multivariate into a well chosen basis. Similarly to [104], using the BIC, we test whether there is evidence that the data structure is a mixture model or not. The method is also similar to [31] which is used for multivariate data. A significant advantage of our procedure compared to [84, 120], is its ability to estimate the number of groups within the data while this number have to be pre-specified in the other methods. Moreover, the tree structure allows us to consider only a small number of principal components at each node of the tree and not to estimate a global number of components for the clustering. Considering tree methods designed for functional data, Staerman et al. [126] extend the popular Isolation Forest algorithm used for anomaly detection, to the functional context. Our fCUBT algorithm is flexible enough to allow for classes defined by certain types of phase variations.

The remainder of the paper is organized as follows. In Section 4.2, we define a model for a mixture of curves for multivariate functional data with the coordinates having possibly different definition domains. Given a dataset, that is a set of, possibly noisy, intermittent measures of an independent sample of realizations of the stochastic process, in Section 4.3, we explain how compute the different quantities that are required in the clustering procedure. In Section 4.4, we develop the construction of our clustering algorithm, named

fCUBT. In Section 4.5, we study the behavior of fCUBT and compare its performance with competing methods both on simulated and real datasets. Our algorithm performs well to estimate the number of groups in the data as well as grouping similar objects together. Once the tree has been grown, it can be used to predict the labels given new observations. The prediction accuracy is compared with the ones derived from supervised methods, and exhibits good performance. A real data application on vehicle trajectories analysis illustrates the effectiveness of our approach. Section 4.6 presents an extension of the method to images data based on the eigendecomposition of the image observations using the Functional Candecomp/Parafac Tensor Power Algorithm (FCP-TPA) [1]. The proofs are left to the Appendix.

4.2 Model and methodology

4.2.1 Notion of multivariate functional data

The structure of our data, referred to as *multivariate functional data*, is very similar to that presented in [64]. The data consist of independent trajectories of a vector-valued stochastic process $X = (X_1, \dots, X_P)^\top$, $P \geq 1$. (Here and in the following, for any matrix A, $A^\top$ denotes its transpose.) For each $1 \leq p \leq P$, let I_p be a rectangle in some Euclidean space $\mathbb{R}^{d_p}$ with $d_p \geq 1$, as for instance, $I_p = [0,1]^{d_p}$. Each coordinate $X_p : I_p \to \mathbb{R}$ is assumed to belong to $\mathcal{L}^2(I_p)$, the Hilbert space of squared-integrable real-valued functions defined on I_p, endowed with the usual inner product that we denote by $\langle \cdot, \cdot \rangle$. Thus X is a stochastic process indexed by $\boldsymbol{t} = (t_1, \dots, t_P)$ belonging to the $P-$fold Cartesian product $\boldsymbol{I} \coloneqq I_1 \times \cdots \times I_P$ and taking values in the $P-$fold Cartesian product space $\mathcal{H} \coloneqq \mathcal{L}^2(I_1) \times \cdots \times \mathcal{L}^2(I_P)$.

We consider the function $\langle\!\langle \cdot, \cdot \rangle\!\rangle : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$,

$$\langle\!\langle f, g \rangle\!\rangle \coloneqq \sum_{p=1}^{P} \langle f_p, g_p \rangle = \sum_{p=1}^{P} \int_{I_p} f_p(t_p) g_p(t_p) \mathrm{d}t_p, \quad f, g \in \mathcal{H}.$$

Then, $\mathcal{H}$ is a Hilbert space with respect to the inner product $\langle\!\langle \cdot, \cdot \rangle\!\rangle$. See [64]. We denote by $|\!|\!| \cdot |\!|\!|$ the norm induced by $\langle\!\langle \cdot, \cdot \rangle\!\rangle$. Let $\mu : \boldsymbol{I} \to \mathcal{H}$ denote the mean function of the process X, that is

$$\mu(\boldsymbol{t}) \coloneqq \mathbb{E}(X(\boldsymbol{t})), \quad \boldsymbol{t} \in \boldsymbol{I}.$$

4.2.2 A mixture model for curves

The standard model-based clustering approaches consider that data is sampled from a mixture of probability densities on a finite dimensional space. As pointed out by Jacques and Preda [84], this approach is not directly applicable to functional data since the notion of probability density generally does not exist for functional random variables. See also [37]. Consequently, model-based clustering approaches assume a mixture of parametric distributions on the coefficients of the representation of the realizations of the process X in a basis. This is also the way we proceed in the following.

Let K be a positive integer, and let Z be a discrete random variable taking values in $\{1, \ldots, K\}$ such that

$$\mathbb{P}(Z = k) = p_k \quad \text{with} \quad p_k > 0 \quad \text{and} \quad \sum_{k=1}^{K} p_k = 1.$$

The variable Z is a latent variable, also called the class label, representing the cluster membership of the realizations of the process.

Let $X = \{X(\boldsymbol{t})\}_{\boldsymbol{t} \in \boldsymbol{I}}$ be a P-dimensional stochastic process. Although X could be defined on a set of vectors and could be vector-valued, we shall call *curves* its independent realizations which generate the data. We consider that the stochastic process follow a *functional mixture model with K components*, that is it allows for the following decomposition:

$$X(\boldsymbol{t}) = \sum_{k=1}^{K} \mu_k(\boldsymbol{t}) \mathbf{1}_{\{Z=k\}} + \sum_{j \geq 1} \xi_j \phi_j(\boldsymbol{t}), \quad \boldsymbol{t} \in \boldsymbol{I}, \tag{4.1}$$

where

- $\mu_1, \ldots, \mu_K \in \mathcal{H}$ are the mean curves per cluster.
- $\{\phi_j\}_{j \geq 1}$ is an orthonormal basis of $\mathcal{H}$, that is $\langle\langle \phi_j, \phi_{j'} \rangle\rangle = 0$ and $\|\phi_j\|^2 = 1$, $\forall 1 \leq j \neq j' < \infty$.
- ξ_j, $j \geq 1$ are real-valued random variables which are conditionally independent given Z. For each $1 \leq k \leq K$, the conditional distribution of ξ_j given $Z = k$ is a zero-mean Gaussian distribution with variance $\sigma_{kj}^2 \geq 0$, for all $j \geq 1$. Moreover, $\sum_{k=1}^{K} \sum_{j \geq 1} \sigma_{kj}^2 < \infty$ and for any $k \neq k'$, $\sum_{j \geq 1} |\sigma_{kj}^2 - \sigma_{k'j}^2| > 0$.

The condition that $\{\phi_j\}_{j \geq 1}$ is an orthonormal basis is not really necessary; it just

allows us to write some technical conditions in a simpler way. Note that our assumptions imply $\sum_{j\geq 1}\mathrm{Var}(\xi_j) < \infty$ and thus $\mathbb{E}\|X\|^2 < \infty$. Note also that the definition of the processes X (4.1) does not require us to know the basis $\{\phi_j\}_{j\geq 1}$. However, for inference purposes, one needs to consider a representation in a workable basis. The following result presents the relationship between the two representations.

Lemma 9. *Let X be defined as in (4.1) for some orthonormal basis $\{\phi_j\}_{j\geq 1}$. Let $\{\psi_j\}_{j\geq 1}$ be another orthonormal basis in $\mathcal{H}$ and consider*

$$c_j = \langle\langle X-\mu, \psi_j\rangle\rangle, \quad j\geq 1 \quad \textit{where} \quad \mu(\cdot) = \sum_{k=1}^{K} p_k\mu_k(\cdot). \tag{4.2}$$

Then

$$c_j \mid Z = k \sim \mathcal{N}\left(m_{kj}, \tau_{kj}^2\right),$$

where

$$m_{kj} = \langle\langle \mu_k - \mu, \psi_j\rangle\rangle \quad \textit{and} \quad \tau_{kj}^2 = \sum_{l\geq 1}\langle\langle \phi_l, \psi_j\rangle\rangle^2\sigma_{kl}^2.$$

Moreover,

$$\mathrm{Cov}\left(c_i, c_j \mid Z = k\right) = \sum_{l\geq 1}\langle\langle \phi_l, \psi_i\rangle\rangle\langle\langle \phi_l, \psi_j\rangle\rangle\sigma_{kl}^2, \quad i,j\geq 1.$$

Each c_j has a centered Gaussian distribution and, for any $i,j\geq 1$,

$$\mathrm{Cov}(c_i,c_j) = \sum_{k=1}^{K} p_k\left(\sum_{l\geq 1}\langle\langle \phi_l, \psi_i\rangle\rangle\langle\langle \phi_l, \psi_j\rangle\rangle\sigma_{kl}^2 + \langle\langle \mu_k-\mu, \psi_i\rangle\rangle\langle\langle \mu_k-\mu, \psi_j\rangle\rangle\right).$$

Remark. Lemma 9 shows that, no matter what the user's choice may be for workable orthonormal basis $\{\psi_j\}_{j\geq 1}$, the clusters will be preserved after expressing the realizations of the process into this basis. However, depending on the aim, some bases might be more suitable than another.

Remark. A careful look at the proof of Lemma 9 reveals that it remains true even if $\{\phi_j\}_{j\geq 1}$ is not an orthonormal basis, which could be appealing for extending our framework. For instance, consider the case where the clusters of curves correspond to representations in different bases. As an illustration, let us consider the case $K=2$ and let $\{\phi_{1,j}\}_{j\geq 1}$ and $\{\phi_{2,j}\}_{j\geq 1}$ be orthonormal systems. However, the union of these sets is not necessarily an orthonormal system. The centered realizations from each cluster corresponds to the representations $\sum_{j\geq 1}\eta_{1,j}\phi_{1,j}(\boldsymbol{t})$ and $\sum_{j\geq 1}\eta_{2,j}\phi_{2,j}(\boldsymbol{t})$, respectively. Here, η_{1j}, η_{2j}, $j\geq 1$ are

independent zero-mean Gaussian variables with variances $\sigma_{1,j}^2$ and $\sigma_{2,j}^2$, respectively, such that $\sum_{j\geq 1}\{\sigma_{1,j}^2+\sigma_{2,j}^2\}<\infty$. Then, we can write the process X in the form (4.1) : $X(\boldsymbol{t})-\mathbb{E}[X(\boldsymbol{t})]=\sum_{j\geq 1}\xi_j\phi_j(\boldsymbol{t})$ with

$$\xi_{2j-1}=\mathbf{1}_{\{Z=1\}}\eta_{1,j}+\mathbf{1}_{\{Z=2\}}\delta_0 \quad \text{and} \quad \xi_{2j}=\mathbf{1}_{\{Z=1\}}\delta_0+\mathbf{1}_{\{Z=2\}}\eta_{2,j}, \quad j\geq 1,$$

and

$$\phi_{2j-1}(\boldsymbol{t})=\phi_{1,j}(\boldsymbol{t}) \quad \text{and} \quad \phi_{2j}(\boldsymbol{t})=\phi_{2,j}(\boldsymbol{t}), \quad j\geq 1,\ \boldsymbol{t}\in\boldsymbol{I},$$

where δ_0 is the Dirac mass at the origin. Thus each ξ_j is a mixture between two centered normal variables with positive and zero variance, respectively. Then, given any orthonormal basis $\{\psi_j\}_{j\geq 1}$, the $c_j=\langle\langle X-\mu,\psi_j\rangle\rangle$ will still have the properties described in Lemma 9.

Remark. Our framework and Lemma 9 could also capture mixtures induced by some phase variations. This kind of modeling approaches received increasing attention recently. See, *e.g.*, [94, 102]. Indeed, let $h_k:\boldsymbol{I}\to\boldsymbol{I}$, $1\leq k\leq K$, be a finite number of diffeomorphisms of $\boldsymbol{I}$. For instance, when $P=1$ and $I_P=[0,1]$, each of the maps h_k or $1-h_k$ could be a function such as $\{bt^a+(1-b)\}^c$ for some $a,c>0$ and $0<b\leq 1$. We can then define $\{\phi_{k,j}\}_{j\geq 1}$ as $\phi_{k,j}(\boldsymbol{t})=\overline{\phi}(h_k(\boldsymbol{t}))$, $\boldsymbol{t}\in\boldsymbol{I}$, $k=1,\ldots,K$, where $\{\overline{\phi}_j\}_{j\geq 1}$ is some reference orthonormal basis. Next, we consider that each cluster corresponds to a representation in a basis $\{\phi_{k,j}\}_{j\geq 1}$, $1\leq k\leq K$. Remark 4.2.2 shows that Lemma 9 remains valid. Let us note that one would obtain a mixture of mixtures if each representation in a basis $\{\phi_{k,j}\}_{j\geq 1}$ is a mixture as in (4.1). This mixture of mixtures is still a mixture in the sense of (4.1), and this illustrates the general framework corresponding to our modeling approach.

In applications, one cannot use an infinite number of terms in the representation of X, and has to truncate such a representation. The following lemma, for which a Gaussian assumption is not required, show that such a truncation could be arbitrarily accurate.

Lemma 10. *Let X be defined as in (4.1) for some orthonormal basis $\{\phi_j\}_{j\geq 1}$. Let $\{\psi_j\}_{j\geq 1}$ be another orthonormal basis in $\mathcal{H}$ with to which X has the decomposition*

$$X(\boldsymbol{t})=\sum_{j\geq 1}c_j\psi_j(\boldsymbol{t}), \quad \boldsymbol{t}\in\boldsymbol{I}.$$

Then

$$\lim_{J\to\infty}\mathbb{E}\left(\left\|X-X_{\lceil J\rceil}\right\|^2\right)=0, \quad \textit{where} \quad X_{\lceil J\rceil}(t)=\sum_{j=1}^{J}c_j\psi_j(\boldsymbol{t}),\boldsymbol{t}\in\boldsymbol{I}.$$

4.2.3 Multivariate Karhunen-Loève representation

Let C denote the $P \times P$ matrix-valued covariance function which, for $\boldsymbol{s}, \boldsymbol{t} \in \boldsymbol{I}$, is defined as

$$C(\boldsymbol{s}, \boldsymbol{t}) \coloneqq \mathbb{E}\left(\{X(\boldsymbol{s}) - \mathbb{E}(X(\boldsymbol{s})\}\{X(\boldsymbol{t}) - \mathbb{E}(X(\boldsymbol{t}))\}^\top\right), \quad \boldsymbol{s}, \boldsymbol{t} \in \boldsymbol{I}.$$

More precisely, for $1 \leq p, q \leq P$, the (p, q)th entry of the matrix $C(\boldsymbol{s}, \boldsymbol{t})$ is the covariance function between the pth and the qth components of the process X:

$$C_{p,q}(s_p, t_q) \coloneqq \mathbb{E}\left(\{X_p(s_p) - \mathbb{E}(X_p(s_p)\}\{X_q(t_q) - \mathbb{E}(X_q(t_q))\}\right), \quad s_p \in I_p, t_q \in I_q.$$

Following [64], herein it is assumed that

$$\max_{1 \leq p \leq P} \sup_{s_p \in I_p} \int_{I_q} C_{p,q}^2(s_p, t_q) \mathrm{d}t_q < \infty,$$

and that, for all $1 \leq p \leq P$, $C_{p,q}(s_p, \cdot)$ is uniformly continuous in the sense that

$$\forall \epsilon > 0 \; \exists \delta > 0 \; : \; |t_q - t_q'| < \delta \; \Rightarrow \; \max_{1 \leq p \leq P} \sup_{s_p \in I_p} \left| C_{p,q}(s_p, t_q) - C_{p,q}(s_p, t_q') \right| < \epsilon.$$

In particular, $C_{p,q}(\cdot, \cdot)$ belongs to $\mathcal{L}^2(I_p \times I_q)$.

Let $\Gamma : \mathcal{H} \mapsto \mathcal{H}$ denote the covariance operator of X, defined as the integral operator with kernel C. That is, for $f \in \mathcal{H}$ and $\boldsymbol{t} \in \boldsymbol{I}$, the qth component of $\Gamma f(\boldsymbol{t})$ is given by

$$(\Gamma f)_q(t_q) \coloneqq \langle\!\langle C_{\cdot,q}(\cdot, t_q), f(\cdot) \rangle\!\rangle, \quad t_q \in \mathcal{T}_q.$$

By the results in [64], the covariance operator Γ is a linear, self-adjoint and positive operator. Moreover, Γ is a compact operator. By the theory of Hilbert-Schmidt operators, *e.g.* [114, Chapter VI], there exists a complete orthonormal basis $\{\varphi_j\}_{j \geq 1} \subset \mathcal{H}$ and a sequence of real numbers $\lambda_1 \geq \lambda_2 \geq \ldots \geq 0$ such that

$$\Gamma \varphi_j = \lambda_j \varphi_j \quad \text{and } \lambda_j \to 0 \text{ as } j \to \infty.$$

The λ_j's are the eigenvalues of the covariance operator Γ and the φ_j's are the associated eigenfunctions. Then, a process X as defined in (4.1) allows for the Karhunen-Loève

representation

$$X(\boldsymbol{t}) = \mu(\boldsymbol{t}) + \sum_{j\geq 1} \mathfrak{c}_j \varphi_j(\boldsymbol{t}), \quad \boldsymbol{t} \in \boldsymbol{I}, \qquad \text{with} \qquad \mathfrak{c}_j = \langle\!\langle X - \mu, \varphi_j \rangle\!\rangle, \tag{4.3}$$

and $\mathrm{Cov}(\mathfrak{c}_j, \mathfrak{c}_l) = \lambda_j \mathbf{1}_{\{j=l\}}$. Let us call $\{\varphi_j\}_{j\geq 1}$ the MFPCA basis.

Let $J \geq 1$ and assume that $\lambda_1 > \lambda_2 \ldots > \lambda_J > \lambda_{J+1}$, which in particular, implies that the first J eigenvalues are nonzero. By an easy extension of [73, Theorem 3.2], we can deduce that, up to a sign, the elements of the MFPCA basis are characterized by the following property :

$$\begin{aligned} \varphi_1 &= \arg\max_{\varphi} \langle\!\langle \Gamma\varphi, \varphi \rangle\!\rangle \quad \text{such that} \quad |\!|\!| \varphi |\!|\!| = 1, \\ \varphi_j &= \arg\max_{\varphi} \langle\!\langle \Gamma\varphi, \varphi \rangle\!\rangle \quad \text{such that} \quad |\!|\!| \varphi |\!|\!| = 1 \text{ and } \langle\!\langle \varphi, \varphi_l \rangle\!\rangle = 0, \ \ \forall l < j \leq J. \end{aligned} \tag{4.4}$$

The following lemma shows that the MFPCA basis is the one which will induce the most accurate truncation for a given J. Therefore, among the workable bases one could use in practice, the MFPCA basis is likely to be a privileged one.

Lemma 11. *Let X be defined as in (4.1). Let $\{\psi_j\}_{j\geq 1}$ be some orthonormal basis in $\mathcal{H}$ and $\{\varphi_j\}_{j\geq 1}$ be the MFPCA basis. Let μ be the mean curve as defined in (4.2). Then, for any $J \geq 1$ such that $\lambda_1 > \lambda_2 \ldots > \lambda_J > \lambda_{J+1}$,*

$$\mathbb{E}\left(\left| |\!|\!| X - \mu - \sum_{j=1}^{J} \langle\!\langle X - \mu, \psi_j \rangle\!\rangle \psi_j \right| |\!|\!|^2 \right) \geq \mathbb{E}\left(\left| |\!|\!| X - \mu - \sum_{j=1}^{J} \mathfrak{c}_j \varphi_j \right| |\!|\!|^2 \right).$$

By Lemma 9, whenever X is defined as in (4.1), for any $J \geq 1$, the distribution of the vector $\mathfrak{c} = (\mathfrak{c}_1, \ldots, \mathfrak{c}_J)^\top$ defined in (4.3) is a centered (multivariate) Gaussian Mixture Model (GMM) distribution

$$g(\mathfrak{c}) = \sum_{k=1}^{K} p_k f_k(\mathfrak{c} | \mathbf{m}_{J,k}, \Sigma_{J,k}), \quad \mathfrak{c} \in \mathbb{R}^J, \tag{4.5}$$

where, for each $1 \leq k \leq K$, $f_k(\cdot | \mathbf{m}_{J,k}, \Sigma_{J,k})$ is the probability density function of a multivariate Gaussian distribution of the kth given the values of its parameters $\mathbf{m}_{J,k}$ and $\Sigma_{J,k}$ such that

$$\mathbf{m}_{J,k} = (\langle\!\langle \mu_k - \mu, \varphi_1 \rangle\!\rangle, \ldots, \langle\!\langle \mu_k - \mu, \varphi_J \rangle\!\rangle)^\top,$$

and the $(i,j)-$entry of the matrix $\Sigma_{J,k}$ given by

$$\mathrm{Cov}\left(\mathfrak{c}_i, \mathfrak{c}_j \mid Z = k\right) = \sum_{l\geq 1} \langle\langle \phi_l, \varphi_i \rangle\rangle \langle\langle \phi_l, \varphi_j \rangle\rangle \sigma^2_{kl}, \quad 1 \leq i, j \leq J.$$

It is worthwhile to point out that the GMM defined in (4.5) is consistent with respect to J in the sense that, for any $J \geq 1$, $(\mathfrak{c}_1, \ldots, \mathfrak{c}_J)^\top$ and $(\mathfrak{c}_1, \ldots, \mathfrak{c}_J, \mathfrak{c}_{J+1})^\top$ have the same label Z, which also coincides with the label of the curve X. In particular, the label Z of a curve in the sample, could be identified by $\mathfrak{c}_1$. Moreover, since $\mathrm{Var}(\mathfrak{c}_1) > \mathrm{Var}(\mathfrak{c}_j)$, $\forall j > 1$, it is likely that the first coefficient in the Karhunen-Loève representation is quite informative for identifying the mixture structure.

Let

$$X_{\lceil J \rceil}(\boldsymbol{t}) = \mu(\boldsymbol{t}) + \sum_{j=1}^{J} \mathfrak{c}_j \varphi_j(\boldsymbol{t}), \quad \boldsymbol{t} \in \boldsymbol{I}, \quad J \geq 1, \tag{4.6}$$

be the truncated Karhunen-Loève expansion of the process X and

$$X_{p,\lceil J_p \rceil}(t_p) = \mu_p(t_p) + \sum_{j=1}^{J_p} \mathbf{c}_{p,j} \rho_{p,j}(t_p), \quad t \in I_p, \quad J_p \geq 1, \quad 1 \leq p \leq P, \tag{4.7}$$

be the truncated Karhunen-Loève expansion of the components of the process X. Happ and Greven [64] derive a direct relationship between the truncated representations (4.7) of the single elements X_p and the truncated representation (4.6) of the multivariate functional data X. We recall this result in the following lemma.

Lemma 12 (Happ and Greven [64], Proposition 5)**.** *Let $X_{\lceil J \rceil}$ be the truncation of the process X as in (4.6) and $X_{p,\lceil J_p \rceil}$ be the truncation of the process X_p for each $1 \leq p \leq P$ as in (4.7).*

1. *Let Γ_p be the univariate covariance operator associated with X_p. The positive eigenvalues of Γ_p, $\lambda_{p,1} \geq \ldots \geq \lambda_{p,J_p} > 0, J_p < J$ correspond the positive eigenvalues of the matrix $\mathbf{A}_p \in \mathbb{R}^{J\times J}$ with entries*

$$\mathbf{A}_{p,jj'} = (\lambda_j \lambda_{j'})^{1/2} \langle \varphi_{p,j}, \varphi_{p,j'} \rangle, \quad j, j' = 1, \ldots, J.$$

The eigenfunctions of Γ_p are given by

$$\rho_{p,j}(t_p) = (\lambda_{p,j})^{-1/2} \sum\nolimits_{j'=1}^{J} \lambda_{j'}^{1/2} [\boldsymbol{u}_{p,j}]_{j'} \varphi_{p,j'}(t_p), \quad t_p \in I_p, \ m = 1, \ldots, J_p,$$

where $\boldsymbol{u}_{p,j}$ *denotes an (orthonormal) eigenvector of* $\mathbf{A}_p$ *associated with eigenvalue* $\lambda_{p,j}$ *and* $[\boldsymbol{u}_{p,j}]_{j'}$ *denotes the* j'*-th entry of this vector. Finally, the univariates scores are*

$$\mathfrak{c}_{p,j} = \langle X_p, \rho_{p,j} \rangle = (\lambda_{p,j})^{-1/2} \sum_{j'=1}^{J} \lambda_{j'}^{1/2} [\boldsymbol{u}_{p,j}]_{j'} \sum_{j''=1}^{J} \mathfrak{c}_{j''} \langle \varphi_{p,j'}, \varphi_{p,j''} \rangle.$$

2. *The positive eigenvalues of* Γ, $\lambda_1, \ldots, \lambda_J > 0$, *with* $J \leq \sum_{p=1}^{P} J_p =: J_+$ *are the positive eigenvalues of the matrix* $\mathbf{Z} \in \mathbb{R}^{J_+ \times J_+}$ *consisting of blocks* $\mathbf{Z}^{(pp')} \in \mathbb{R}^{J_p \times J_{p'}}$ *with entries*

$$\mathbf{Z}_{jj'}^{(pp')} = \mathrm{Cov}\left(\mathfrak{c}_{p,j}, \mathfrak{c}_{p',j'}\right), \quad j = 1, \ldots, J_p, \ j' = 1, \ldots, J_{p'}, \ p, p' = 1, \ldots, P.$$

The eigenfunctions of Γ *are given by*

$$\varphi_{p,j}(t_p) = \sum_{j'=1}^{J_p} [\boldsymbol{v}_j]_{p,j'} \rho_{p,j}(t_p), \quad t_p \in I_p, \ j = 1, \ldots, J,$$

where $[\boldsymbol{v}_j]_p \in \mathbb{R}^{J_p}$ *denotes the* p*-th block of an (orthonormal) eigenvector* $\boldsymbol{v}_j$ *of* $\mathbf{Z}$ *associated with eigenvalue* λ_j*. The scores are given by*

$$\mathfrak{c}_j = \sum_{p=1}^{P} \sum_{j'=1}^{J_p} [\boldsymbol{v}_j]_{p,j'} \mathfrak{c}_{p,j'}.$$

Lemma 12 is particularly useful as it allows us to compute the scores of the P-dimensional stochastic process X using the scores of each of the P components X_p. However, when P grows, it might not be suitable to perform P univariate fPCA from a computational point of view. Recently, Hu and Yao [75] extend this result to large-dimensional processes by imposing a sparsity assumption. This possible extension will be investigated in future work.

4.3 Parameters estimation

In real data applications, the realizations of X are usually measured with error at discrete, and possibly random, points in the definition domain. Therefore, let us consider N curves $X^{(1)}, \ldots, X^{(n)}, \ldots, X^{(N)}$ generated as a random sample of the P-dimensional stochastic process X with continuous trajectories. For each $1 \leq n \leq N$, and given a vector of positive integers $\boldsymbol{M}_n = (M_{n,1}, \ldots, M_{n,P}) \in \mathbb{R}^P$, let $T_{\boldsymbol{m}}^{(n)} = (T_{m_1}^{(n)}, \ldots, T_{m_P}^{(n)}), 1 \leq m_p \leq M_{n,p}, 1 \leq$

$p \leq P$, be the random observation times for the curve $X^{(n)}$. These times are obtained as independent copies of a variable $\boldsymbol{T}$ taking values in $\boldsymbol{I}$. The vectors $\boldsymbol{M}_1, \ldots, \boldsymbol{M}_N$ represent an independent sample of an integer-valued random vector $\boldsymbol{M}$ with expectation $\boldsymbol{\mu_M}$ which increases with N. We assume that the realizations of X, $\boldsymbol{M}$ and $\boldsymbol{T}$ are mutually independent. The observations associated with a curve, or trajectory, $X^{(n)}$ consist of the pairs $(Y_{\boldsymbol{m}}^{(n)}, T_{\boldsymbol{m}}^{(n)}) \in \mathbb{R}^P \times \boldsymbol{I}$, where $\boldsymbol{m} = (m_1, \ldots, m_P), 1 \leq m_p \leq M_{n,p}$, $1 \leq p \leq P$, and $Y_{\boldsymbol{m}}^{(n)}$ is defined as

$$Y_{\boldsymbol{m}}^{(n)} = X^{(n)}(T_{\boldsymbol{m}}^{(n)}) + \varepsilon_{\boldsymbol{m}}^{(n)}, \quad 1 \leq n \leq N, \tag{4.8}$$

with the $\varepsilon_{\boldsymbol{m}}^{(n)}$ being independent copies of a centered error random vector $\boldsymbol{\varepsilon} \in \mathbb{R}^P$ with finite variance. We use the notation $X^{(n)}(\boldsymbol{t})$ for the value at $\boldsymbol{t}$ of the realization $X^{(n)}$ of X. The N-sample of X is composed of two sub-populations: a *learning set* of N_0 curves to estimate the mixture components of the process X and a set of N_1 curves to be classified using the previous grouping as a classifier that we call the *online set*. Thus, $1 \leq N_0, N_1 < N$ and $N_0 + N_1 = N$. Let $X^{(1)}, \ldots, X^{(N_0)}$ denote the curves corresponding to the *learning set*. The learning sample will be used to estimate the mean and covariance functions, as well as the eigencomponents, of the process X. Our first objective is to construct a partition $\mathcal{U}$ of the space $\mathcal{H}$ using the learning sample. Then, the second aim is to use the partition $\mathcal{U}$ as a classifier for a possibly very large set of N_1 new curves. Let

$$X^{[1]} = X^{(N_0+1)}, \ldots, X^{[N_1]} = X^{(N)},$$

denote the curves from the *online set* to be classified using the partition $\mathcal{U}$.

4.3.1 Estimation of mean and covariance

In this section, we develop estimators for the mean and the covariance functions of a component $X_p, 1 \leq p \leq P$ from the process X. These estimators are used to compute estimators of eigenvalues and eigenfunctions of X_p for the Karhunen-Loève expansion (4.7). It is worthwhile to notice that, because of Lemma 12, we do not need to estimate the covariance between X_p and X_q for $p \neq q$.

Let $\widehat{X}_p^{(n)}$ be a suitable nonparametric estimator of the curve $X_p^{(n)}$ applied with the $M_{n,p}$ pairs $(Y_{m_p}^{(n)}, T_{m_p}^{(n)}), n = 1, \ldots, N_0$, as for instance a local polynomial estimator such as the one defined in [59]. With at hand, the $\widehat{X}_p^{(n)}$'s tuned for the mean function estimation,

we define

$$\widehat{\mu}_{p,N_0}(t_p) = \frac{1}{N_0}\sum_{n=1}^{N_0} \widehat{X}_p^{(n)}(t_p), \quad t_p \in I_p.$$

For the covariance function, following [146], we distinguish the diagonal from the non-diagonal points. With at hand, the $\widehat{X}_p^{(n)}$'s tuned for the covariance function estimation,

$$\widehat{C}_{p,p}(s_p, t_p) = \frac{1}{N_0}\sum_{n=1}^{N_0} \widehat{X}_p^{(n)}(s_p)\widehat{X}_p^{(n)}(t_p) - \widehat{\mu}_{p,N_0}(s_p)\widehat{\mu}_{p,N_0}(t_p), \quad s_p, t_p \in I_p, \ \ s_p \neq t_p. \quad (4.9)$$

The diagonal of the covariance is then estimated using two-dimensional kernel smoothing with $\widehat{C}_{p,p}(s_p, t_p), s_p \neq t_p$ as input data. See [146] for the details.

4.3.2 Derivation of the MFPCA components

Following Happ and Greven [64], using Lemma 12, we estimate the multivariate components for X by plugging the univariate components computed from each X_p. These estimations are done as follows. First, we perform an univariate fPCA on each of the components of X separately. For a component X_p, the eigenfunctions and eigenvectors are computed as a matrix analysis of the estimated covariance $\widehat{C}_{p,p}$, from (5.4). This results in a set of eigenfunctions $\left(\widehat{\rho}_{p,1}, \ldots, \widehat{\rho}_{p,J_p}\right)$ associated with a set of eigenvalues $\left(\widehat{\lambda}_{p,1}, \ldots, \widehat{\lambda}_{p,J_p}\right)$ for a given truncation integer J_p. Then, the univariate scores for a realization $X_p^{(n)}$ of X_p are given by $\widehat{\mathbf{c}}_{p,j}^{(n)} = \langle \widehat{X}_p^{(n)}, \widehat{\rho}_{p,j} \rangle$, $1 \leq j \leq J_p$. These scores might be estimated by numerical integration. However, in some cases, *e.g.* for sparse data, it may be more suitable to use the PACE method (see [146]). We then define the matrix $\mathcal{Z} \in \mathbb{R}^{N_0 \times J_+}$, where on each row we concatenate the scores obtained for the P components of the nth observation: $\left(\widehat{\mathbf{c}}_{1,1}^{(n)}, \ldots, \widehat{\mathbf{c}}_{1,J_1}^{(n)}, \ldots, \widehat{\mathbf{c}}_{P,1}^{(n)}, \ldots, \widehat{\mathbf{c}}_{P,J_P}^{(n)}\right)$. An estimate $\widehat{\mathbf{Z}} \in \mathbb{R}^{J_+ \times J_+}$ of the matrix $\mathbf{Z}$, from Lemma 12, is given by $\widehat{\mathbf{Z}} = (N_0 - 1)^{-1}\mathcal{Z}^\top\mathcal{Z}$. An eigenanalysis of the matrix $\widehat{\mathbf{Z}}$ is done to estimate the eigenvectors $\widehat{\boldsymbol{v}}_j$ and eigenvalues $\widehat{\lambda}_j$. And finally, the multivariate eigenfunctions are estimated with

$$\widehat{\varphi}_{p,j}(t_p) = \sum_{j'=1}^{J_p} [\widehat{\boldsymbol{v}}_j]_{p,j'} \widehat{\rho}_{p,j'}(t_p), \quad t_p \in I_p, \ 1 \leq j \leq J_+, \ 1 \leq p \leq P.$$

and the multivariate scores with

$$\widehat{\mathbf{c}}_j^{(n)} = \mathcal{Z}_{n,\cdot}\widehat{\boldsymbol{v}}_j, \quad 1 \leq n \leq N_0, \quad 1 \leq j \leq J_+.$$

The multivariate Karhunen-Loève expansion of the process X is thus

$$\widehat{X}^{(n)}(\boldsymbol{t}) = \widehat{\mu}_{N_0}(\boldsymbol{t}) + \sum_{j=1}^{J} \widehat{\mathfrak{c}}_j^{(n)} \widehat{\varphi}_j(\boldsymbol{t}), \quad \boldsymbol{t} \in \mathcal{T}.$$

where $\widehat{\mu}_{N_0}(\cdot) = (\widehat{\mu}_{1,N_0}(\cdot), \ldots, \widehat{\mu}_{P,N_0}(\cdot))$ is the vector of the estimated mean functions.

4.4 Multivariate functional clustering

Let $\mathcal{S}$ be a sample of realizations of the process X, defined in (4.1). We consider the problem of learning a partition $\mathcal{U}$ such that every element U of $\mathcal{U}$ gathers similar elements of $\mathcal{S}$. Our clustering procedure follows the idea of CUBT, considered by Fraiman et al. [48], which we adapt to functional data. In the following, we describe in detail the fCUBT algorithm.

4.4.1 Building the maximal tree

Let $\mathcal{S}_{N_0} = \{X^{(1)}, \ldots, X^{(N_0)}\}$ be a training sample composed of N_0 independent realizations of the stochastic process $X \in \mathcal{H}$ defined in (4.1). In the following, a tree $\mathfrak{T}$ is a full binary tree, meaning every node has zero or two children, which represents a nested partition of the sample $\mathcal{S}_{N_0}$. We will denote the depth of a tree $\mathfrak{T}$ by $\mathfrak{D} \geq 1$.

A tree $\mathfrak{T}$ starts with the root node $\mathfrak{S}_{0,0}$ to which we assign the whole space sample $\mathcal{S}_{N_0}$. Next, every node $\mathfrak{S}_{\mathfrak{d},\mathfrak{j}} \subset \mathcal{S}_{N_0}$ is indexed by the pair $(\mathfrak{d}, \mathfrak{j})$ where $\mathfrak{d}$ is the depth index of the node, with $0 \leq \mathfrak{d} < \mathfrak{D}$, and $\mathfrak{j}$ is the node index, with $0 \leq \mathfrak{j} < 2^{\mathfrak{d}}$. A non-terminal node $(\mathfrak{d}, \mathfrak{j})$ has two children, corresponding to disjoint subsets $\mathfrak{S}_{\mathfrak{d}+1,2\mathfrak{j}}$ and $\mathfrak{S}_{\mathfrak{d}+1,2\mathfrak{j}+1}$ of $\mathcal{S}_{N_0}$ such that

$$\mathfrak{S}_{\mathfrak{d},\mathfrak{j}} = \mathfrak{S}_{\mathfrak{d}+1,2\mathfrak{j}} \cup \mathfrak{S}_{\mathfrak{d}+1,2\mathfrak{j}+1}.$$

A terminal node $(\mathfrak{d}, \mathfrak{j})$ has no children.

A tree $\mathfrak{T}$ is thus defined using a top-down procedure by recursively splitting. At each stage, a node $(\mathfrak{d}, \mathfrak{j})$ is possibly split into two subnodes, namely the left and right child, with indices $(\mathfrak{d}+1, 2\mathfrak{j})$ and $(\mathfrak{d}+1, 2\mathfrak{j}+1)$, respectively, provided it fulfills some condition. A multivariate functional principal components analysis as presented in Section 4.3.2, with $\mathtt{n}_{\mathtt{comp}}$ components, where $\mathtt{n}_{\mathtt{comp}} \leq J$, is then conducted on the elements of $\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}$. This results in a set of eigenvalues $\Lambda_{\mathfrak{d},\mathfrak{j}} = \left(\lambda_{\mathfrak{d},\mathfrak{j}}^1, \ldots, \lambda_{\mathfrak{d},\mathfrak{j}}^{\mathtt{n}_{\mathtt{comp}}}\right)$ associated with a set of eigenfunctions $\Phi_{\mathfrak{d},\mathfrak{j}} = \left(\varphi_{\mathfrak{d},\mathfrak{j}}^1, \ldots, \varphi_{\mathfrak{d},\mathfrak{j}}^{\mathtt{n}_{\mathtt{comp}}}\right)$. The matrix of scores $C_{\mathfrak{d},\mathfrak{j}}$ is then defined with the columns built

with the projections of the elements of $\mathcal{S}_{\mathfrak{d},\mathfrak{j}}$ onto the elements of $\Phi_{\mathfrak{d},\mathfrak{j}}$. More precisely, to each $X^{(n)} \in \mathfrak{S}_{\mathfrak{d},\mathfrak{j}}$ there is a corresponding column of size $\mathtt{n_{comp}}$ defined as

$$\mathfrak{c}_{\mathfrak{d},\mathfrak{j}}^{(n)} = \left(\langle\langle X^{(n)} - \mu_{\mathfrak{d},\mathfrak{j}}, \varphi_{\mathfrak{d},\mathfrak{j}}^{1}\rangle\rangle, \ldots, \langle\langle X^{(n)} - \mu_{\mathfrak{d},\mathfrak{j}}, \varphi_{\mathfrak{d},\mathfrak{j}}^{\mathtt{n_{comp}}}\rangle\rangle\right)^{\top}, \tag{4.10}$$

where $\mu_{\mathfrak{d},\mathfrak{j}}$ is the mean curve within the node $\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}$.

We can retrieve the groups (clusters) of curves considering a mixture model as in Section 4.2.2 for the columns of the matrix of scores. At each node $\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}$, for each $K = 1, \ldots, K_{max}$, we fit a GMM as in (4.5) to the columns of the matrix $C_{\mathfrak{d},\mathfrak{j}}$. The resulting models are denoted as $\{\mathcal{M}_1, \ldots, \mathcal{M}_{K_{max}}\}$. To fit $\mathcal{M}_1$, we use the standard mean and variance matrix estimation of Gaussian distributions. To fit each of the models $\{\mathcal{M}_2, \ldots, \mathcal{M}_{K_{max}}\}$, we use an EM algorithm [40]. In particular, the EM algorithm assigns a label to each curve in the node. We next consider the BIC, defined in [121], and determine

$$\widehat{K}_{\mathfrak{d},\mathfrak{j}} = \underset{K=1,\ldots,K_{max}}{\arg\max}\ \mathrm{BIC}(\mathcal{M}_K) = \underset{K=1,\ldots,K_{max}}{\arg\max}\ \{2\log(\mathcal{L}_K) - \kappa \log|\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}|\}, \tag{4.11}$$

where $\mathcal{L}_K$ is the likelihood function of the $K-$components multivariate Gaussian mixture model $\mathcal{M}_K$ for the data at the node $\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}$, that is

$$\mathcal{L}_K = \prod_{n=1}^{|\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}|} \sum_{k=1}^{K} p_k f_k(\mathfrak{c}_{\mathfrak{d},\mathfrak{j}}^{(n)} | \mathbf{m}_{\mathtt{n_{comp}},\mathtt{k}}, \Sigma_{\mathtt{n_{comp}},\mathtt{k}}),$$

$\kappa = K + K\mathtt{n_{comp}} + K\mathtt{n_{comp}}(\mathtt{n_{comp}}+1)/2 - 1$ is the dimension of the model $\mathcal{M}_K$ and $|\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}|$ is the cardinality of the set $\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}$. If $\widehat{K}_{\mathfrak{d},\mathfrak{j}} > 1$, we split $\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}$ using the model $\mathcal{M}_2$, that is a mixture of two Gaussian vectors. Otherwise, the node is considered to be a terminal node and the construction of the tree is stopped for this node.

The recursive procedure continues downward until one of the following stopping rules are satisfied: there are less than `minsize` observations in the node or the estimation $\widehat{K}_{\mathfrak{d},\mathfrak{j}}$ of the number of clusters in the mode $\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}$ is equal to 1. The value of the positive integer `minsize` is set by the user. When the algorithm ends, a label is assigned to each leaf (terminal node). The resulting tree is referred to as the maximal binary tree. The Algorithm 3 presents an implementation of the construction of the tree.

Our algorithm provides a partition of the sample. Each observation belongs to a leaf which is associated with a unique label. In a perfect case, this tree will have the same number of leaves as the number of mixture components of X. In practice, it is rarely the

Algorithm 3: Construction of a tree $\mathfrak{T}$

Input: A training sample $\mathcal{S}_{N_0} = \{X^{(1)}, \ldots, X^{(N_0)}\} \subset \mathcal{H}$ and the hyperparameters $\mathtt{n_{comp}}$, K_{max} and $\mathtt{minsize}$.
Initialization: Set $(\mathfrak{d}, \mathfrak{j}) = (0, 0)$ and $\mathfrak{S}_{0,0} = \mathcal{S}_{N_0}$.
Computation of the MFPCA components: Perform a MFPCA with $\mathtt{n_{comp}}$ components on the data in the node $\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}$ and get the set of eigenvalues $\Lambda_{\mathfrak{d},\mathfrak{j}}$ associated with a set of eigenfunctions $\Phi_{\mathfrak{d},\mathfrak{j}}$. Build the matrix $C_{\mathfrak{d},\mathfrak{j}}$ defined in (4.10).
Estimation of the number of clusters: For each $K = 1, \ldots, K_{max}$, fit $K-$components GMM using an EM algorithm on the columns of the matrix $C_{\mathfrak{d},\mathfrak{j}}$. The models are denoted by $\{\mathcal{M}_1, \ldots, \mathcal{M}_{K_{max}}\}$. The number of mixture components is estimated by $\widehat{K}_{\mathfrak{d},\mathfrak{j}}$ defined in (5.5) using the BIC.
Stopping criterion: Test if the node indexed by $(\mathfrak{d}, \mathfrak{j})$ is a terminal node, that is if $\widehat{K}_{\mathfrak{d},\mathfrak{j}} = 1$ or if there are less than $\mathtt{minsize}$ elements in $\mathcal{S}_{\mathfrak{d},\mathfrak{j}}$. If the node is terminal, then stop the construction of the tree for this node, otherwise go to the next step.
Children nodes construction: A non-terminal node indexed by $(\mathfrak{d}, \mathfrak{j})$ is split into two subnodes as follows:

1. Fit a $2-$component GMM using an EM algorithm.
2. For each element of $\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}$, compute the posterior probability to belong to the first component.
3. Form the children nodes as

$$\mathfrak{S}_{\mathfrak{d}+1,2\mathfrak{j}} = \{\text{elements of } \mathfrak{S}_{\mathfrak{d},\mathfrak{j}} \text{ with posterior probability } \geq 1/2\}$$

and

$$\mathfrak{S}_{\mathfrak{d}+1,2\mathfrak{j}+1} = \mathfrak{S}_{\mathfrak{d},\mathfrak{j}} \setminus \mathfrak{S}_{\mathfrak{d}+1,2\mathfrak{j}}.$$

Recursion: Continue the procedure by applying the **Computation of the MFPCA components** step to the nodes $(\mathfrak{d}+1, 2\mathfrak{j})$ and $(\mathfrak{d}+1, 2\mathfrak{j}+1)$.
Output: A set of nodes $\{\mathfrak{S}_{\mathfrak{d},\mathfrak{j}},\ 0 \leq \mathfrak{j} < 2^{\mathfrak{d}},\ 0 \leq \mathfrak{d} < \mathfrak{D}\}$.

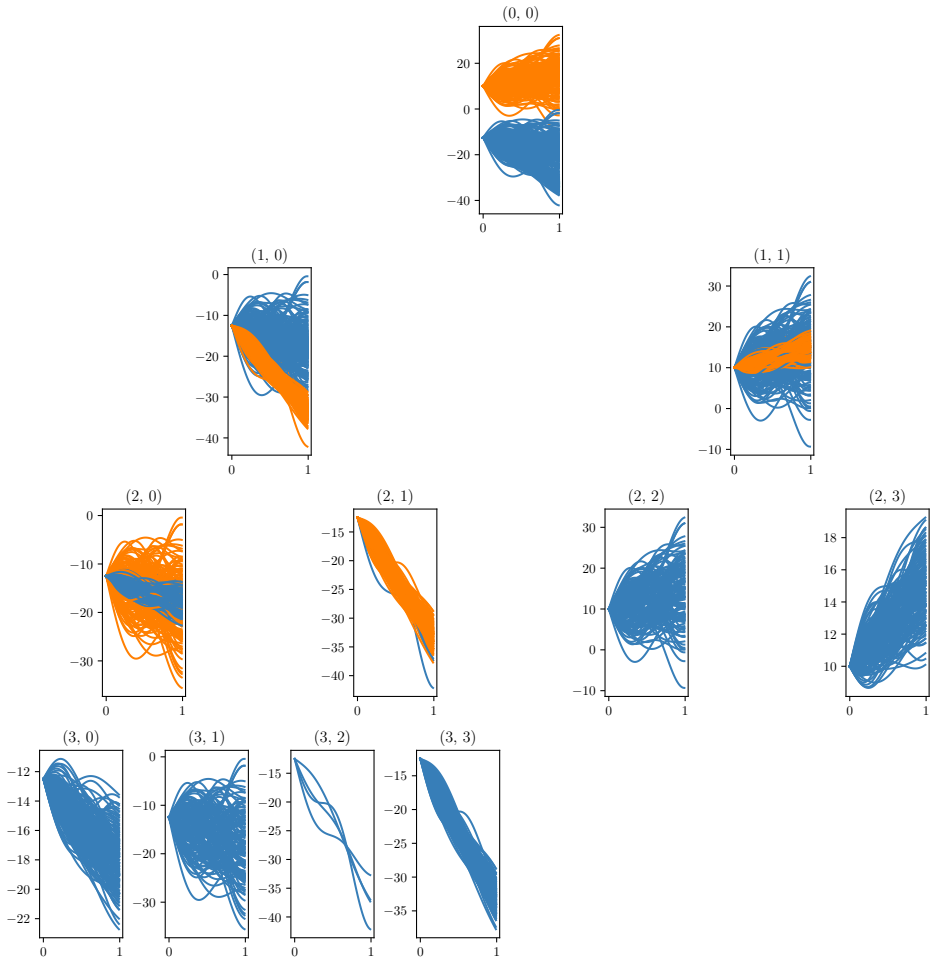


Figure 4.1: Illustration of maximal tree for Scenario 1 simulated data.

case, and the number of leaves may be much larger than the number of clusters. That is why an agglomerative step, that we call the joining step, should also be considered. Note that we do not have to pre-specify the number of clusters before performing the joining step. Moreover, if the number of clusters in the maximal tree is the one wanted by the user, it is not necessary to run the joining step.

An example of a maximal tree is given in Figure 4.1. It corresponds to a simulated dataset defined in Section 4.5, Scenario 1, where $\texttt{n}_{\texttt{comp}}$ is set to explain 95% of the variance at each node of the tree, $K_{max} = 5$ and $\texttt{minsize} = 10$. This tree has six leaves, whereas Scenario 1 contains only five clusters. In this illustration, the joining step will be helpful to join the group with only 4 curves, corresponding to the node $\mathfrak{S}_{3,2}$, with another group, hopefully the node $\mathfrak{S}_{3,1}$.

The original procedure, developed in [48], included a pruning step. For each sibling

node, this step computes a measure of dissimilarities between them and collapses them if this measure is lower than a predefined threshold. They need this step because their splitting criteria is based on the deviance reduction between a node and its children. The algorithm is stopped when this deviance reduction is less than a predefined threshold. So, the pruning step is used to eventually revise an undesirable split. In our case, the splitting criteria is not based on how much deviance we gain at each node of the tree but on an estimation of the number of modes of a Gaussian mixture model. Thus, the pruning step is not useful in our case.

This method has three hyperparameters that should be set by the user. The first one is the number of components kept when a MFPCA is run ($\texttt{n}_{\texttt{comp}}$). Its default value is set to explain 95% of the variance within the data. We do not need to have a large $\texttt{n}_{\texttt{comp}}$ because of the binary split situation. Thus, only few components are require to discriminate between two groups at each node of the tree. The second hyperparameter is the minimum number of elements within a node to be considered to be split (`minsize`). In practice, it should be larger than 2, otherwise, the computation of the MFPCA is not feasible. The default value is set to 10. The last one, K_{max}, refers to the number of Gaussian mixture models to try in order to decide if there is at least two clusters in the data. Its default value is set to 5 but, in practice, to reduce computation time, it could be set to 2. Thus, we will just compare the model with two modes against that with zero clusters.

4.4.2 Joining step

In this step, the idea is to join terminal nodes which do not necessarily share the same direct ascendant. Let $\mathcal{G} = (V, E)$ be a graph where $V = \{\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}, 0 \leq j < 2^{\mathfrak{d}}, 0 \leq \mathfrak{d} < \mathfrak{D} \mid \mathfrak{S}_{\mathfrak{d},\mathfrak{j}} \text{ is a terminal node}\}$ is a set of vertices and $E \subseteq \mathbf{E}$ is a set of edges with $\mathbf{E}$ the complete set of unordered pairs of vertices $\{(\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}, \mathfrak{S}_{\mathfrak{d}',\mathfrak{j}'}) \mid \mathfrak{S}_{\mathfrak{d},\mathfrak{j}}, \mathfrak{S}_{\mathfrak{d}',\mathfrak{j}'} \in V \text{ and } \mathfrak{S}_{\mathfrak{d},\mathfrak{j}} \neq \mathfrak{S}_{\mathfrak{d}',\mathfrak{j}'}\}$. Let us clarify the definition of the set of edges E. Consider $(\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}, \mathfrak{S}_{\mathfrak{d}',\mathfrak{j}'}) \in \mathbf{E}$. We are interested in the union of the nodes in the pair, $\mathfrak{S}_{\mathfrak{d},\mathfrak{j}} \cup \mathfrak{S}_{\mathfrak{d}',\mathfrak{j}'}$. After performing a MFPCA on the set $\mathfrak{S}_{\mathfrak{d},\mathfrak{j}} \cup \mathfrak{S}_{\mathfrak{d}',\mathfrak{j}'}$, we compute the matrix $C_{(\mathfrak{d},\mathfrak{j})\cup(\mathfrak{d}',\mathfrak{j}')}$ using (4.10). For $K = 1, \ldots, K_{max}$, we fit a K-components Gaussian mixture model using an EM algorithm on $C_{(\mathfrak{d},\mathfrak{j})\cup(\mathfrak{d}',\mathfrak{j}')}$. Then, if the estimated number of clusters $\widehat{K}_{(\mathfrak{d},\mathfrak{j})\cup(\mathfrak{d}',\mathfrak{j}')}$ using (5.5), is equal to 1, we add the pair to E. Finally, we have

$$E = \left\{(\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}, \mathfrak{S}_{\mathfrak{d}',\mathfrak{j}'}) \mid \mathfrak{S}_{\mathfrak{d},\mathfrak{j}}, \mathfrak{S}_{\mathfrak{d}',\mathfrak{j}'} \in V,\ \mathfrak{S}_{\mathfrak{d},\mathfrak{j}} \neq \mathfrak{S}_{\mathfrak{d}',\mathfrak{j}'} \text{ and } \widehat{K}_{(\mathfrak{d},\mathfrak{j})\cup(\mathfrak{d}',\mathfrak{j}')} = 1\right\}. \tag{4.12}$$

We associate with each element $(\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}, \mathfrak{S}_{\mathfrak{d}',\mathfrak{j}'})$ of E, the value of the BIC that corresponds to $\widehat{K}_{(\mathfrak{d},\mathfrak{j})\cup(\mathfrak{d}',\mathfrak{j}')}$. The edge of $\mathcal{G}$ that corresponds to the maximum of the BIC is then removed and the associated vertices are joined. Thus, there is one cluster less. This procedure is run recursively until no pair of nodes can be joined according to the BIC or only one node in the tree remains. Finally, unique labels are associated with each remaining element of V and the partition of $\mathfrak{S}_{0,0}$, denoted as $\mathcal{U}$, is returned. Algorithm 4 presents a possible implementation for the joining step.

Algorithm 4: Joining step

Input: A set of nodes $\{\mathfrak{S}_{\mathfrak{d},\mathfrak{j}},\ 0 \leq \mathfrak{j} < 2^{\mathfrak{d}},\ 0 \leq \mathfrak{d} < \mathfrak{D}\}$ and the hyperparameters $\mathtt{n_{comp}}$ and K_{max}.
Initialization: Build the set of terminal nodes

$$V = \{\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}, 0 \leq j < 2^{\mathfrak{d}}, 0 \leq \mathfrak{d} < \mathfrak{D} \mid \mathfrak{S}_{\mathfrak{d},\mathfrak{j}} \text{ is a terminal node}\}.$$

Creation of the graph: Build the set E defined in (5.6) and denote by $\mathcal{G}$ the graph (V, E). Associate with each edge $(\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}, \mathfrak{S}_{\mathfrak{d}',\mathfrak{j}'})$ the value of the BIC that corresponds to $\widehat{K}_{(\mathfrak{d},\mathfrak{j})\cup(\mathfrak{d}',\mathfrak{j}')}$.
Stopping criterion: If the set E is empty or the set V is reduced to a unique element, stop the algorithm.
Aggregation of two nodes: Let $(\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}, \mathfrak{S}_{\mathfrak{d}',\mathfrak{j}'})$ be the edge with the maximum BIC value. Then, remove this edge and replace the asssociated vertice by $\mathfrak{S}_{\mathfrak{d},\mathfrak{j}} \cup \mathfrak{S}_{\mathfrak{d}',\mathfrak{j}'}$.
Recursion: Continue the procedure by applying the **Creation of the graph** step with $\{V \setminus \{\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}, \mathfrak{S}_{\mathfrak{d}',\mathfrak{j}'}\}\} \cup \{\mathfrak{S}_{\mathfrak{d},\mathfrak{j}} \cup \mathfrak{S}_{\mathfrak{d}',\mathfrak{j}'}\}$.
Output: A partition $\mathcal{U}$ of $\mathfrak{S}_{0,0}$ and labels associated to each element of $\mathcal{S}_{N_0}$.

4.4.3 Classification of new observations

With at hand, a partition $\mathcal{U}$ of $\mathcal{S}_{N_0}$ obtained from a learning set of N_0 realizations of the process X, we aim to classify N_1 new trajectories of X from what we call, the online dataset. Denote $\mathcal{S}_{N_1}$ this set of new trajectories.

Let $\mathfrak{X}$ be an element of the set $\mathcal{S}_{N_1}$. The descent of the tree $\mathfrak{T}$ is performed as follows. Let $\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}, 0 \leq j < 2^{\mathfrak{d}}, 0 \leq \mathfrak{d} < \mathfrak{D}$ be the node at hand such that $\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}$ is not a terminal node. We compute the projection $\mathfrak{X}$ onto the eigenfunctions $\Phi_{\mathfrak{d},\mathfrak{j}}$. This results in the vector

$$\left(\langle\langle \mathfrak{X} - \mu_{\mathfrak{d},\mathfrak{j}}, \varphi^1_{\mathfrak{d},\mathfrak{j}}\rangle\rangle, \ldots, \langle\langle \mathfrak{X} - \mu_{\mathfrak{d},\mathfrak{j}}, \varphi^{\mathtt{n_{comp}}}_{\mathfrak{d},\mathfrak{j}}\rangle\rangle\right)^\top.$$

Then, using the 2-components GMM fitted on this node, we compute the posterior probability of $\mathfrak{X}$ belonging to each of the components. At this point, we have the probability for $\mathfrak{X}$ belonging to each node given it belongs to the parent node. Write $\texttt{Pa}(\mathfrak{S})$, the set of parent nodes of the node $\mathfrak{S}$ which includes the node itself. Denote $\mathbb{P}^\star(\cdot) := \mathbb{P}(\cdot|X^{(1)}, \ldots, X^{(N_0)})$. Then, the probability for $\mathfrak{X}$ to be in the node $\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}$ is given by

$$\mathbb{P}^\star(\mathfrak{X} \in \mathfrak{S}_{\mathfrak{d},\mathfrak{j}}) = \prod_{\mathfrak{S} \in \texttt{Pa}(\mathfrak{S}_{\mathfrak{d},\mathfrak{j}})} \mathbb{P}^\star(\mathfrak{X} \in \mathfrak{S} | \mathfrak{X} \in \texttt{Pa}(\mathfrak{S})).$$

We have $\mathbb{P}^\star(\mathfrak{X} \in \mathfrak{S}_{0,0}) = 1$. In order to compute the probabilities to belong to each element of the partition $\mathcal{U}$, let $\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}$ and $\mathfrak{S}_{\mathfrak{d}',\mathfrak{j}'}$ be two terminal nodes that have been joined in the joining step into the node $\mathfrak{S}_{\mathfrak{d},\mathfrak{j}} \cup \mathfrak{S}_{\mathfrak{d}',\mathfrak{j}'}$ and

$$\mathbb{P}^\star(\mathfrak{X} \in \mathfrak{S}_{\mathfrak{d},\mathfrak{j}} \cup \mathfrak{S}_{\mathfrak{d}',\mathfrak{j}'}) = \mathbb{P}^\star(\mathfrak{X} \in \mathfrak{S}_{\mathfrak{d},\mathfrak{j}}) + \mathbb{P}^\star(\mathfrak{X} \in \mathfrak{S}_{\mathfrak{d}',\mathfrak{j}'}),$$

because the sets $\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}$ and $\mathfrak{S}_{\mathfrak{d}',\mathfrak{j}'}$ are disjoint. Finally, each cluster is associated with a probability for the observation $\mathfrak{X}$ such that $\sum_{U \in \mathcal{U}} \mathbb{P}^\star(\mathfrak{X} \in U) = 1$. We assign $\mathfrak{X}$ to the cluster with the largest probability.

This procedure is run for each observation within $\mathcal{S}_{N_1}$ and it results to a partition $\mathcal{V}$ of $\mathcal{S}_{N_1}$. The Algorithm 5 presents a possible implementation for the classification of one observation.

4.5 Empirical analysis

By means of simulated data, in this section we illustrate the behavior of our clustering algorithm and compare it with some competitors. A real data application on a vehicle trajectory dataset is also carried out.

Our `fCUBT` procedure is compared to diverse competitors in the literature that are both designed for univariate and multivariate functional data: `FunHDDC` ([10, 120] and [119] for the **R** implementation) and `Funclust` ([82, 84] and [125] for the **R** implementation). Moreover, our approach competes with the methodology described in [78], which corresponds to the k-means algorithm with a suitable distance for functional data. In

Algorithm 5: Classify one observation

Input: A new realization $\mathfrak{X}$ of the process X, the complete tree $\mathfrak{T}$ and the partition $\mathcal{U}$.

for $\mathfrak{S}_{\mathfrak{d},\mathfrak{j}} \in \{\mathfrak{S}_{\mathfrak{d},\mathfrak{j}},\ 0 \leq \mathfrak{j} < 2^{\mathfrak{d}},\ 0 \leq \mathfrak{d} < \mathfrak{D}\}$ **do**

1. Compute the vector

$$\left(\langle\!\langle \mathfrak{X} - \mu_{\mathfrak{d},\mathfrak{j}}, \varphi^{1}_{\mathfrak{d},\mathfrak{j}} \rangle\!\rangle, \ldots, \langle\!\langle \mathfrak{X} - \mu_{\mathfrak{d},\mathfrak{j}}, \varphi^{\mathtt{n}_{\mathtt{comp}}}_{\mathfrak{d},\mathfrak{j}} \rangle\!\rangle\right)^{\top};$$

2. Compute the posterior probability to belong to each component of the 2-components GMM fitted on $\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}$;

3. Compute the probability to be in $\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}$ as

$$\mathbb{P}^{\star}(\mathfrak{X} \in \mathfrak{S}_{\mathfrak{d},\mathfrak{j}}) = \prod_{\mathfrak{S} \in \mathtt{Pa}(\mathfrak{S}_{\mathfrak{d},\mathfrak{j}})} \mathbb{P}^{\star}(\mathfrak{X} \in \mathfrak{S} | \mathfrak{X} \in \mathtt{Pa}(\mathfrak{S})).$$

end

for $U \in \mathcal{U}$ **do**

Compute $\mathbb{P}^{\star}(\mathfrak{X} \in U)$.

end

Output: A label for $\mathfrak{X}$ which is defined as $\arg\max_{U \in \mathcal{U}} \mathbb{P}^{\star}(\mathfrak{X} \in U)$.

particular, the methodology in [78] uses the following distances:

$$d_1(X,Y) = \left(\sum_{p=1}^{P} \int_{I_p} \left(X_p(t_p) - Y_p(t_p) \right)^2 dt_p \right)^{1/2} \quad \text{and}$$
$$d_2(X,Y) = \left(\sum_{p=1}^{P} \int_{I_p} \left(\frac{dX_p(t_p)}{dt_p} - \frac{dY_p(t_p)}{dt_p} \right)^2 dt_p \right)^{1/2},$$

where $dX_p(t_p)/dt_p$ is the first derivative of $X_p(t_p)$. These two methods are denoted as k-`means`-d_1 and k-`means`-d_2 in the following. We use the implementation developed in [72] for both univariate and multivariate functional data. We also compare our algorithm with a GMM, fitted using an EM algorithm, on the coefficients of a functional principal components analysis on the dataset with a fixed number of components, quoted as `FPCA+GMM` in the following. Finally, we consider only the first step of the `fCUBT` algorithm, which is the growth of the tree, to point out the usefulness of the joining step. The method will be referred as `Growing` in the following.

We are greatly interested in the ability of the algorithms to retrieve the true number of clusters K. When the true labels are available, the estimated partitions are compared with the true partition using the Adjusted Rand Index (ARI) [77], which is an "adjusted for chance" version of the Rand Index [113]. Let $\mathcal{U} = \{U_1, \ldots, U_r\}$ and $\mathcal{V} = \{V_1, \ldots, V_s\}$ be two different partitions of $\mathcal{S}_N$, *i.e*

$$U_i \subset \mathcal{S}_N, \quad 1 \leq i \leq r, \qquad V_j \subset \mathcal{S}_N, \quad 1 \leq j \leq s,$$
$$\mathcal{S}_N = \bigcup_{i=1}^{r} U_i = \bigcup_{j=1}^{s} V_j,$$
$$\text{and} \quad U_i \cap U_{i'} = \varnothing, \quad 1 \leq i, i' \leq r, \qquad V_j \cap V_{j'} = \varnothing, \quad 1 \leq j, j' \leq s.$$

Denote by $n_{ij} \coloneqq |U_i \cap V_j|, 1 \leq i \leq r, 1 \leq j \leq s$, the number of elements of $\mathcal{S}_N$ that are common to the sets U_i and V_j. Then, $n_{i\cdot} \coloneqq |U_i|$ (or $n_{\cdot j} \coloneqq |V_j|$) correspond to the number of elements in U_i (or V_i). With these notations, the ARI is defined as

$$\text{ARI}(\mathcal{U}, \mathcal{V}) = \frac{\sum_{i=1}^{r} \sum_{j=1}^{s} \binom{n_{ij}}{2} - \left[\sum_{i=1}^{r} \binom{n_{i\cdot}}{2} \sum_{j=1}^{s} \binom{n_{\cdot j}}{2} \right] / \binom{N}{2}}{\frac{1}{2} \left[\sum_{j=1}^{s} \binom{n_{\cdot j}}{2} + \sum_{j=1}^{s} \binom{n_{\cdot j}}{2} \right] - \left[\sum_{i=1}^{r} \binom{n_{i\cdot}}{2} \sum_{j=1}^{s} \binom{n_{\cdot j}}{2} \right] / \binom{N}{2}}. \tag{4.13}$$

4.5.1 Simulation experiments

We consider three simulations scenarios with varying degrees of difficulty. Each experience is repeated 500 times.

Scenario 1. In our first scenario, we consider that the random curves are observed without noise. The number of clusters is fixed at $K = 5$, $P = 1$, $I_1 = [0, 1]$. An independent sample of $N = 1000$ univariate curves is simulated according to the following model: for $t \in [0, 1]$:

$$\begin{aligned}
&\text{Cluster 1:} \quad X(t) = \mu_1(t) + c_{11}\phi_1(t) + c_{12}\phi_2(t) + c_{13}\phi_3(t),\\
&\text{Cluster 2:} \quad X(t) = \mu_1(t) + c_{21}\phi_1(t) + c_{22}\phi_2(t) + c_{23}\phi_3(t),\\
&\text{Cluster 3:} \quad X(t) = \mu_2(t) + c_{11}\phi_1(t) + c_{12}\phi_2(t) + c_{13}\phi_3(t),\\
&\text{Cluster 4:} \quad X(t) = \mu_2(t) + c_{21}\phi_1(t) + c_{22}\phi_2(t) + c_{23}\phi_3(t),\\
&\text{Cluster 5:} \quad X(t) = \mu_2(t) + c_{21}\phi_1(t) + c_{22}\phi_2(t) + c_{23}\phi_3(t) - 15t,
\end{aligned}$$

where ϕ_k's are the eigenfunctions of the Wiener process which are defined by

$$\phi_k(t) = \sqrt{2}\sin\left(\left(k - \frac{1}{2}\right)\pi t\right), \quad k = 1, 2, 3,$$

and the mean functions μ_1 and μ_2 by

$$\mu_1(t) = \frac{20}{1 + \exp(-t)} \quad \text{and} \quad \mu_2(t) = \frac{-25}{1 + \exp(-t)}.$$

The c_{ij}'s are random normal variables defined by

$$\begin{aligned}
&c_{11} \sim \mathcal{N}(0, 16), && c_{12} \sim \mathcal{N}(0, 64/9), && c_{13} \sim \mathcal{N}(0, 16/9),\\
&c_{21} \sim \mathcal{N}(0, 1), && c_{22} \sim \mathcal{N}(0, 4/9), && c_{23} \sim \mathcal{N}(0, 1/9).
\end{aligned}$$

The mixing proportions are equal, and the curves are observed on 101 equidistant points. As shown in Figure 4.2, the five clusters cannot be retrieved using only the mean curve per cluster: cluster 1 (blue) and 2 (orange) share the same mean function μ_1 and similarly cluster 3 (green) and 4 (pink) share the same mean function μ_2. As a consequence, clustering algorithms based on distances to the mean of the clusters, such as k-means, are not expected to perform well in this case. For `FunHDDC` and `Funclust`, the functional form of the data is reconstructed using a cubic B-spline basis, smoothing with 25 basis functions.

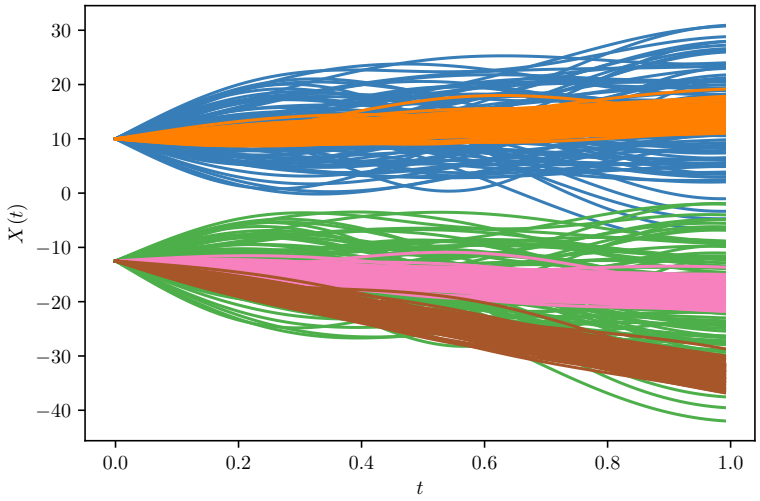


Figure 4.2: Simulated data for Scenario 1.

Scenario 2. The second simulation is a modification of the data simulation process of [120, scenario C]. Here, we consider that the measurements of the random curves are noisy. Thus, for this scenario, the number of clusters is fixed at $K = 5$, $P = 2$, $I_1 = I_2 = [0, 1]$. An independent sample of $N = 1000$ bivariate curves is simulated according to the following model : for $t_1, t_2 \in [0, 1]$,

$$
\begin{aligned}
\text{Cluster 1:} \quad & X_1(t_1) = h_1(t_1) + b_{0.9}(t_1), \\
& X_2(t_2) = h_3(t_2) + 1.5 \times b_{0.8}(t_2), \\
\text{Cluster 2:} \quad & X_1(t_1) = h_2(t_1) + b_{0.9}(t_1), \\
& X_2(t_2) = h_3(t_2) + 0.8 \times b_{0.8}(t_2), \\
\text{Cluster 3:} \quad & X_1(t_1) = h_1(t_1) + b_{0.9}(t_1), \\
& X_2(t_2) = h_3(t_2) + 0.2 \times b_{0.8}(t_2) \\
\text{Cluster 4:} \quad & X_1(t_1) = h_2(t_1) + 0.1 \times b_{0.9}(t_1), \\
& X_2(t_2) = h_2(t_2) + 0.2 \times b_{0.8}(t_2), \\
\text{Cluster 5:} \quad & X_1(t_1) = h_3(t_1) + b_{0.9}(t_1), \\
& X_2(t_2) = h_1(t_2) + 0.2 \times b_{0.8}(t_2).
\end{aligned}
$$

The functions h are defined, by $h_1(t) = (6 - |20t - 6|)_+ /4$, $h_2(t) = (6 - |20t - 14|)_+ /4$ and $h_3(t) = (6 - |20t - 10|)_+ /4$, for $t \in [0, 1]$. (Here, $(\cdot)_+$ denotes the positive part of the expression between the brackets.) The functions b_H are defined, for $t \in [0, 1]$, by

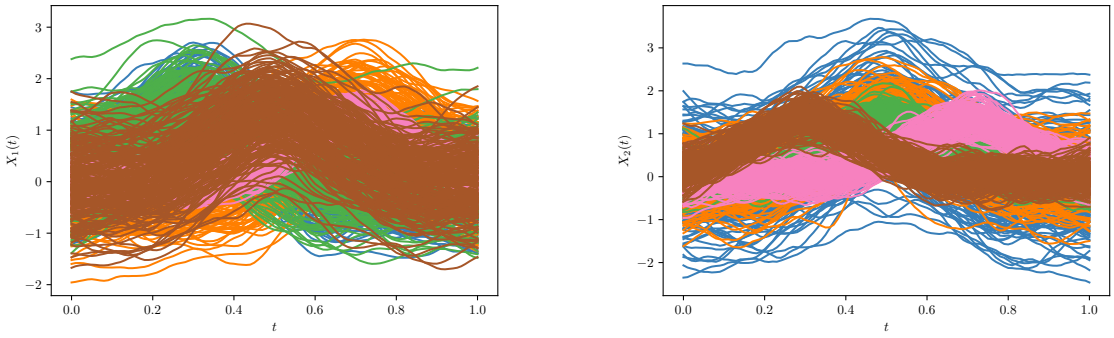


Figure 4.3: Simulated data for Scenario 2.

$b_H(t) = (1+t)^{-H} B_H(1+t)$ where $B_H(\cdot)$ is a fractional Brownian motion with Hurst parameter H. The mixing proportions are set to be equal.

The data to which we apply the clustering are obtained as in (4.8). Each component curve is observed at 101 equidistant points in $[0, 1]$. The bivariate error vectors have zero-mean Gaussian independent components with variance $1/2$. Figure 4.3 presents the smoothed version, using the methodology of [59], of the simulated data. As pointed out by [120], the different clusters cannot be identified using only one variable: cluster 1 (blue) is similar to cluster 3 (green) for variable $X_1(t)$ and in like manner, cluster 1 (blue) is like cluster 2 (orange) and cluster 3 (green) for variable $X_2(t)$. The brown and pink groups might be considered as "noise" clusters that aim to make discrimination between the other groups harder. Hence, clustering methods that are specialized for univariate data should fail to retrieve true membership using only $X_1(t)$ or $X_2(t)$. For `FunHDDC` and `Funclust`, the functional form of the data is reconstructed using a cubic B-spline basis, smoothing with 25 basis functions.

Scenario 3. The last simulation is the same as the second one, except we add some correlation between the components. So, for each $n \in \{1, \ldots, N\}$, we observe a realization of the vector $X = (X_1 + \alpha X_2, X_2)^\top$, where $\alpha = 0.4$. Figure 4.4 presents the smoothed version, using the methodology of [59], of the simulated data. For `FunHDDC` and `Funclust`, the functional form of the data is reconstructed using a cubic B-spline basis, smoothing with 25 basis functions.

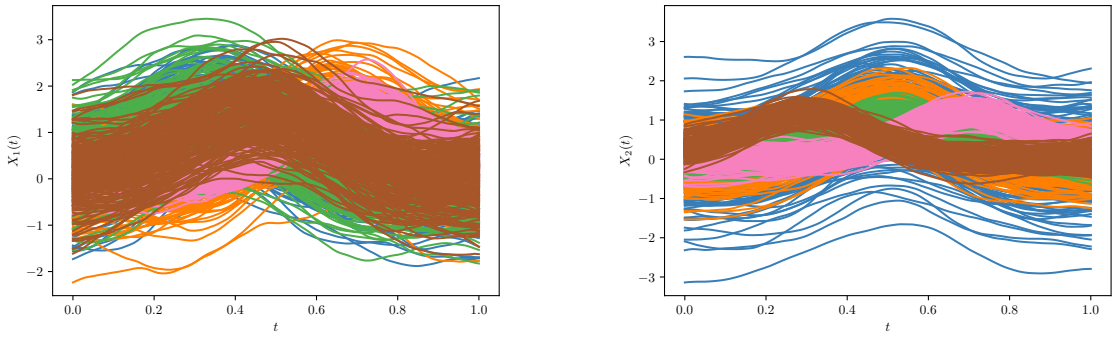


Figure 4.4: Simulated data for Scenario 3.

Model selection

In this section, we investigate the selection of the number of clusters for each of methods on each of the simulations. The way to return the number of clusters in a dataset depends on the algorithm. Thus, the selection for the model `fCUBT` and `Growing` is based on the BIC. Similarly, the BIC is also used for the `FunHDDC` algorithm. For all the other methods, we test all the models between $K = 1$ and $K = 8$, and return the model that maximizes the ARI criteria as the selected model. The simulation settings have been repeated 500 times and the model $[ab_kQ_kD_k]$ is used for the `FunHDDC` algorithm.

Table 4.1 summarizes the results of the 500 simulations for each scenario. We remark that the `fCUBT` algorithm performs well in retrieving the right number of clusters in all the scenarios, being the first or second method in terms of retrieving percentage. Quite surprisingly, the k-`means`-d_2 algorithm performs very well for the second and third scenarios. It indicates that the distances between the derivatives of the curves are much more informative than the distance between the original ones. The accuracy of the selection of the number of clusters in competitors, designed for functional data, `FunHDDC` and `Funclust`, is very poor. This result has also been pointed out in [148] where the simulated data are much simpler. In half of the case, `FunHDDC` algorithm does not find any cluster for the scenarios 2 and 3. Finally, the results on `Growing` show the usefulness of the joining step. Thus, at the end of the growing step, we may have a large number clusters (even greater than 10) but with very few data in some of them, and so the joining step allows us to get rid of them, and thus have more relevant clusters.

Scenario 1.	Number of clusters K									
Method	1	2	3	4	5	6	7	8	9	10+
`fCUBT`	-	-	-	-	0.982	0.018	-	-	-	-
`Growing`	-	-	-	-	0.596	0.234	0.080	0.028	0.026	0.036
`FPCA+GMM`	-	-	0.002	0.012	0.530	0.316	0.110	0.030	-	-
`FunHDDC`	-	0.492	0.378	0.114	0.014	0.002	-	-	-	-
`Funclust`	-	0.444	0.442	0.106	0.006	0.002	-	-	-	-
k-`means-`d_1	-	-	0.150	0.154	0.606	0.016	0.070	0.004	-	-
k-`means-`d_2	-	-	-	0.002	0.050	0.290	0.362	0.296	-	-

(a) Scenario 1.

Scenario 2.	Number of clusters K									
Method	1	2	3	4	5	6	7	8	9	10+
`fCUBT`	-	-	-	-	0.692	0.182	0.100	0.022	0.002	0.002
`Growing`	-	-	-	-	0.516	0.184	0.120	0.086	0.034	0.060
`FPCA+GMM`	-	-	-	-	0.266	0.450	0.248	0.036	-	-
`FunHDDC`	0.448	0.472	0.054	0.022	0.004	-	-	-	-	-
`Funclust`	-	0.284	0.174	0.152	0.156	0.136	0.072	0.026	-	-
k-`means-`d_1	-	-	0.002	0.018	0.046	0.080	0.178	0.676	-	-
k-`means-`d_2	-	-	0.056	0.116	0.822	0.004	0.002	-	-	-

(b) Scenario 2.

Scenario 3.	Number of clusters K									
Method	1	2	3	4	5	6	7	8	9	10+
`fCUBT`	-	-	-	-	0.664	0.238	0.074	0.022	0.002	-
`Growing`	-	-	-	-	0.604	0.182	0.082	0.062	0.026	0.044
`FPCA+GMM`	-	-	-	-	0.414	0.396	0.164	0.026	-	-
`FunHDDC`	0.508	0.492	-	-	-	-	-	-	-	-
`Funclust`	-	0.066	0.182	0.192	0.200	0.196	0.136	0.028	-	-
k-`means-`d_1	-	-	-	-	0.034	0.144	0.206	0.616	-	-
k-`means-`d_2	-	0.004	0.01	0.094	0.874	0.010	0.002	0.006	-	-

(c) Scenario 3.

Table 4.1: Number of clusters selected for each model, expressed as a proportion over 500 simulations

Benchmark with existing methods

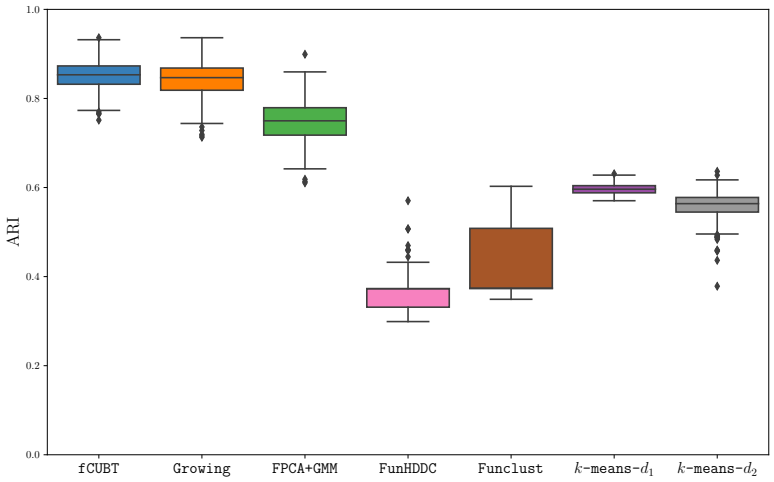
In this section, our algorithm is compared to competitors in the literature with respect to the ARI criteria on the three scenario settings. All the competitors are applied for $K = 1$ to $K = 8$ groups for each of the scenarios and we return the best ARI found regardless of the number of clusters.

Figure 4.5 presents clustering results for all the tested models. We see that our algorithm performs well in all the scenarios. On the contrary, competitors do not perform well on this simulated data. In particular, for the multivariate settings (scenario 2 and 3), `FunHDDC` has a lot of convergence issues, and as a consequence, its clustering performance is very poor. Both k-`means`-d_1 and k-`means`-d_2 demonstrate acceptable results although not as good as `fCUBT`. The `FPCA+GMM` algorithm has similar results as `fCUBT` in terms of ARI because ARI is not penalized when the number of clusters is not the true one. So, as long as the clusters are not mixed, the ARI will be good, even if a large cluster is split into multiple small ones. The same phenomenon appears in the case of `Growing` compare to `fCUBT`.

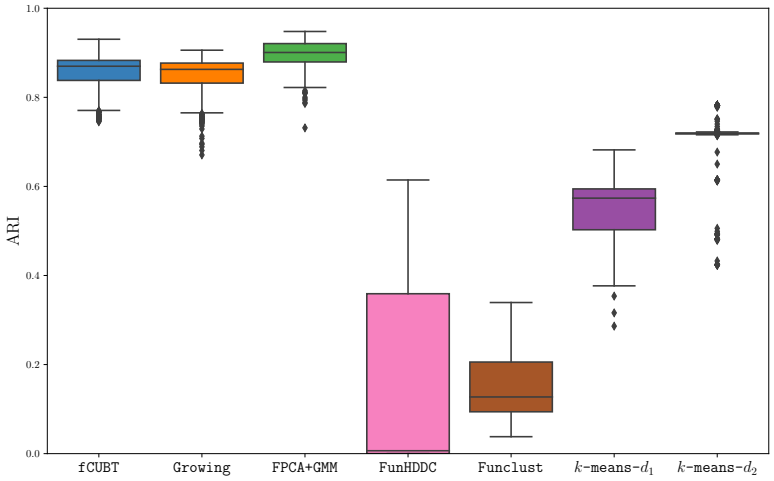
A comparison with supervised methods

We compare our functional clustering procedure with two supervised models. These models are the following: first, we perform a MFPCA to extract features that explain 99% of the variance within the data. Then, we fit a Gaussian Process Classifier (`GPC`) and a Random Forest Classifier (`Random Forest`) to the extracted features.

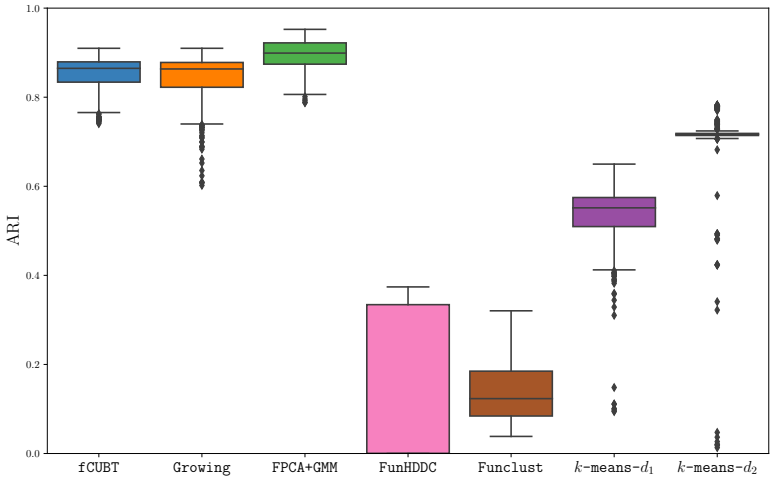
For each of the scenarios, we generate a sample of $N = 1000$ curves, considering them as the complete dataset. Then, we randomly sample 2/3 of the dataset to build the training set and the remaining ones form the test set. The supervised models are trained on the training set and we predict the outcome on the test set. The ARI measure is finally computed using the true labels and the prediction. For the `fCUBT` method, we only consider the test set to learn the clusters. The ARI is computed as usual using the clusters found and the true labels. However, here, we only consider `fCUBT` to compete with supervised models when it retrieves the right number of clusters. We performed the simulations 500 times, and the results are plotted in Figure 4.6. Between parenthesis, we have written the number of times `fCUBT` gets the right number of clusters over the 500 simulations, and so the number of simulations we examine for the computation of the ARI. We point out that this retrieving percentage is a lot smaller than that in the



(a) Scenario 1



(b) Scenario 2



(c) Scenario 3

Figure 4.5: Estimation of ARI for all tested models on 500 simulations.

previous section but it is due to the fact that here, we only considers a dataset of size $N = 330$ and not $N = 1000$ as before. We remark that our unsupervised method is as good as supervised ones when the true number of classes is found.

Comments on the classification of new set of curves

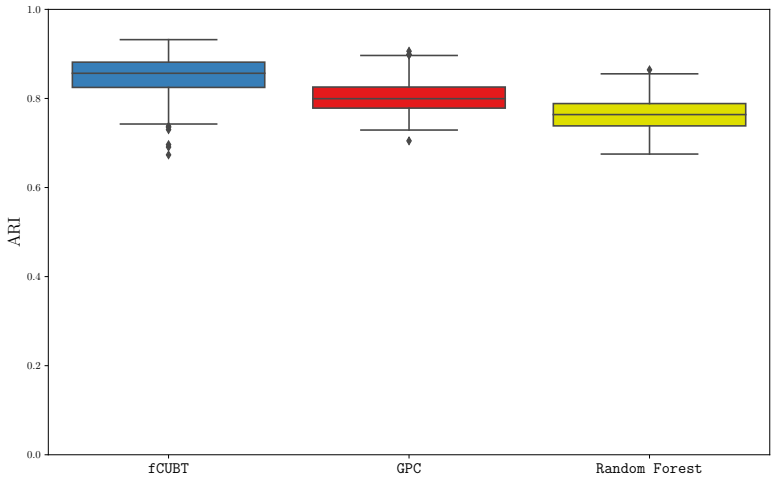
For each of the simulated scenarios, we apply the following process. We learn a tree $\mathfrak{T}$ as well as the partition $\mathcal{U}$ from the learning set $\mathcal{S}_{N_0}$. Different sizes of learning sets are considered, $N_0 = 200$, 500 or 1000. We generate a new set of data, $\mathcal{S}_{N_1}$, referred to as the online dataset, of size $N_1 = 1000$. As the data are simulated, we know the true labels of each observations within $\mathcal{S}_{N_1}$. Denote the true partition of $\mathcal{S}_{N_1}$ by $\mathcal{V}$. We then classify new observations from the online set $\mathcal{S}_{N_1}$ and denote the obtained partition by $\mathcal{V}'$. Partitions are compared using $\text{ARI}(\mathcal{V}, \mathcal{V}')$ defined in (4.13).

The simulations are performed 500 times, and the results are plotted in Figure 4.7. The three scenarios present similar patterns. Thus, when $N_0 = 200$, the partition $\mathcal{U}$ obtained using the `fCUBT` algorithm is not general enough to capture all elements within each cluster. In this case, the ARI is less than 0.8. When $N_0 = 500$, the partition $\mathcal{U}$ is now sufficiently accurate to represents the clusters ($\text{ARI} > 0.8$). However, the stability of the clusters is not guaranteed, regarding the large variance in the estimation of the ARI. Finally, when $N_0 = 1000$, we have both accurate partitioning $\mathcal{U}$ ($\text{ARI} > 0.8$) and stable clusters (low variance).

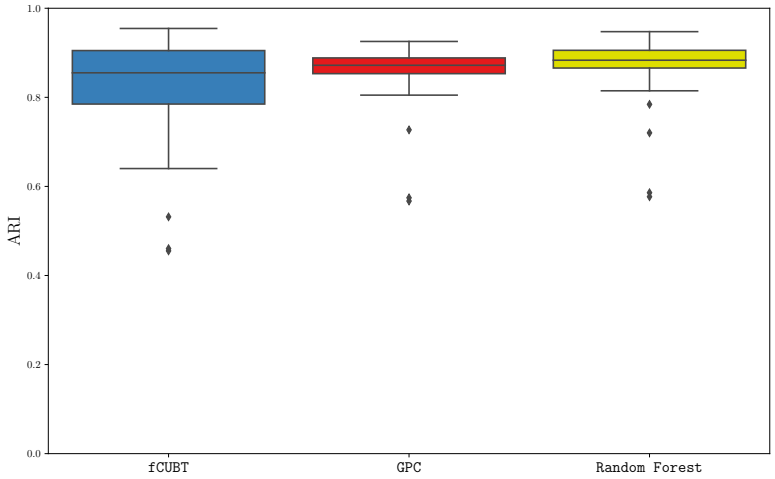
4.5.2 Real data analysis: the rounD dataset

In this section, our method is applied to a part of the rounD dataset [88], which is "naturalistic road user trajectories recorded at German roundabouts". This dataset is part of a set of vehicle trajectories data provided by the Institute for Automotive Engineering (ika) in RWTH Aachen University. One may cite the highD dataset (about highways) [89] and the inD dataset (about intersections) [8], such as other ones produced by ika. These datasets are particularly useful for studying the behavior of road users in some specific situations. They start to replace the NGSIM study [47], widely used in traffic flow studies, as a benchmark for models about trajectory prediction or classification because they provide more accurate data (see *e.g.* [81, 98, 97, 42, 5, 144] for some references).

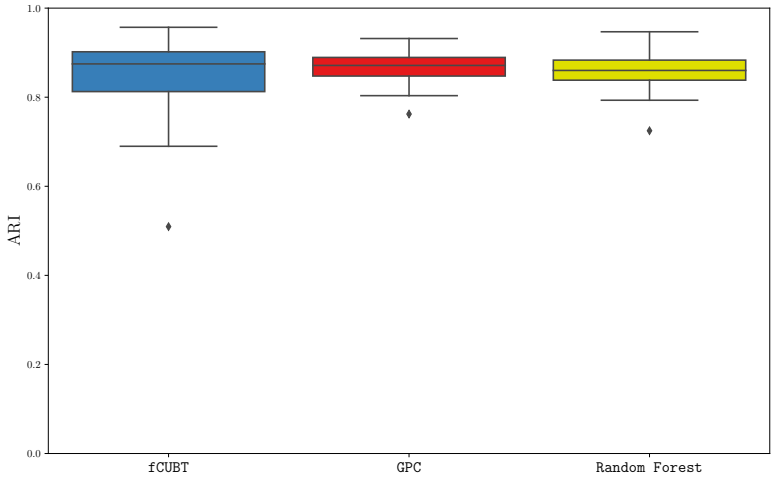
For our illustration, we consider a subset of the rounD dataset, that corresponds to one particular roundabout (see Figure 4.8). It contains 18 minutes of trajectories for



(a) Scenario 1 (257/500)

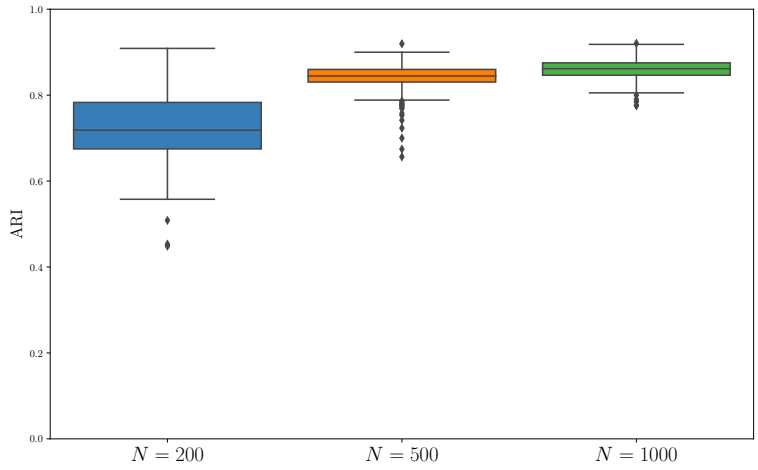


(b) Scenario 2 (172/500)

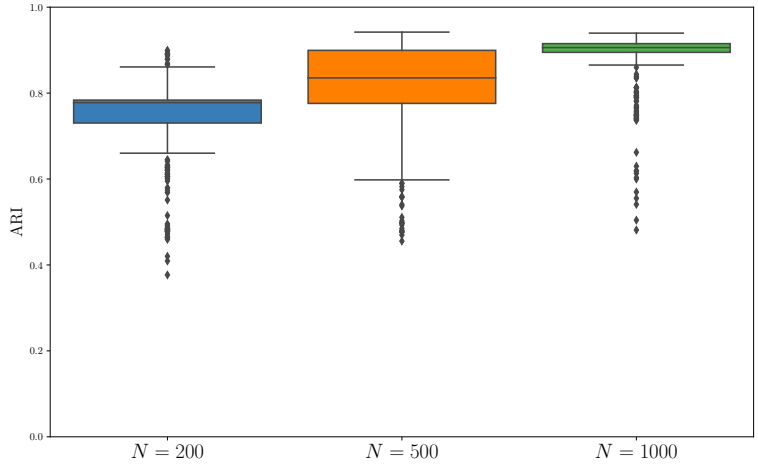


(c) Scenario 3 (188/500)

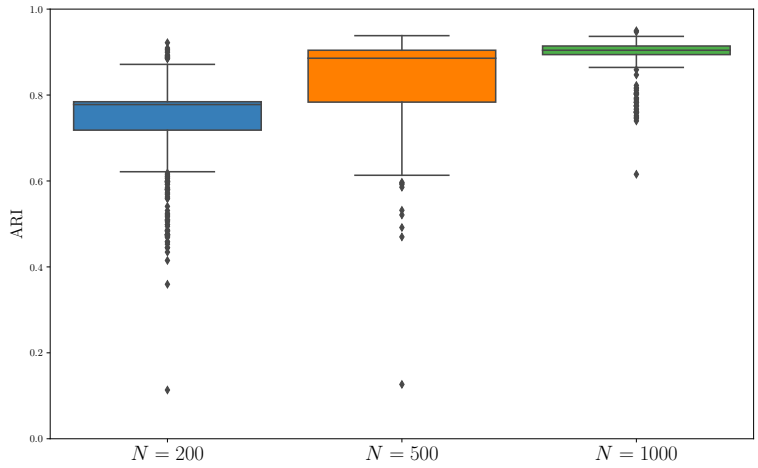
Figure 4.6: Estimation of ARI for the comparison with supervised models.



(a) Scenario 1



(b) Scenario 2



(c) Scenario 3

Figure 4.7: Estimation of ARI with respect to the size of the learning dataset when the tree is used as a supervised classifier.

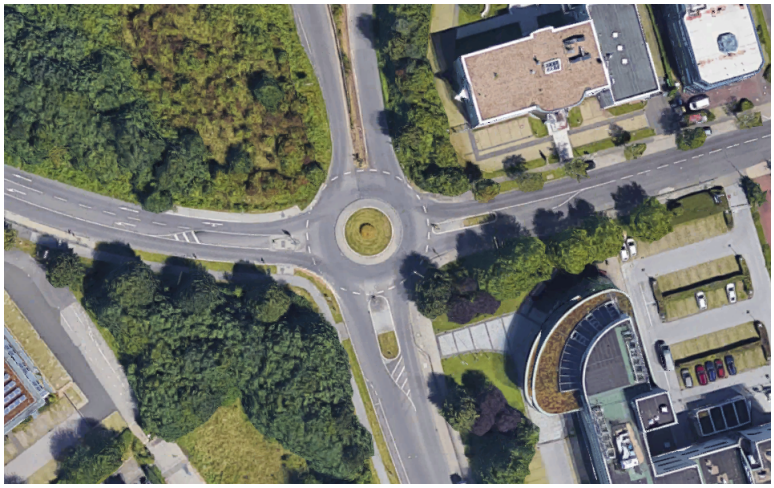
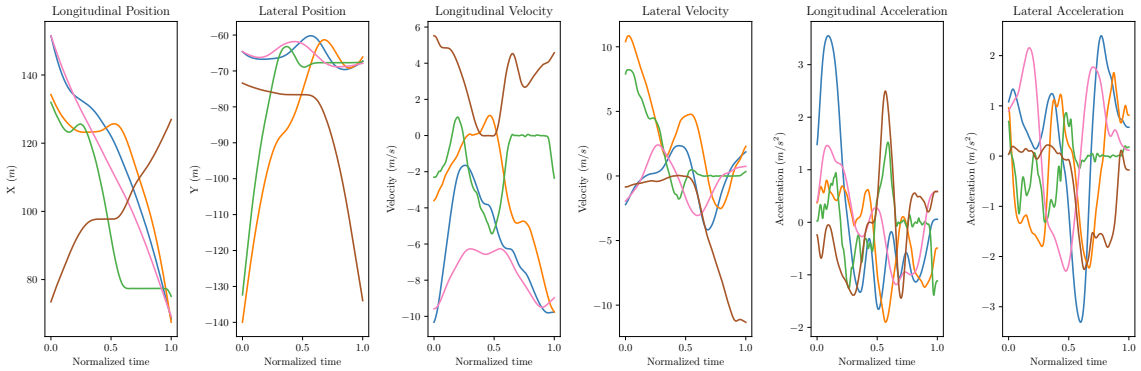
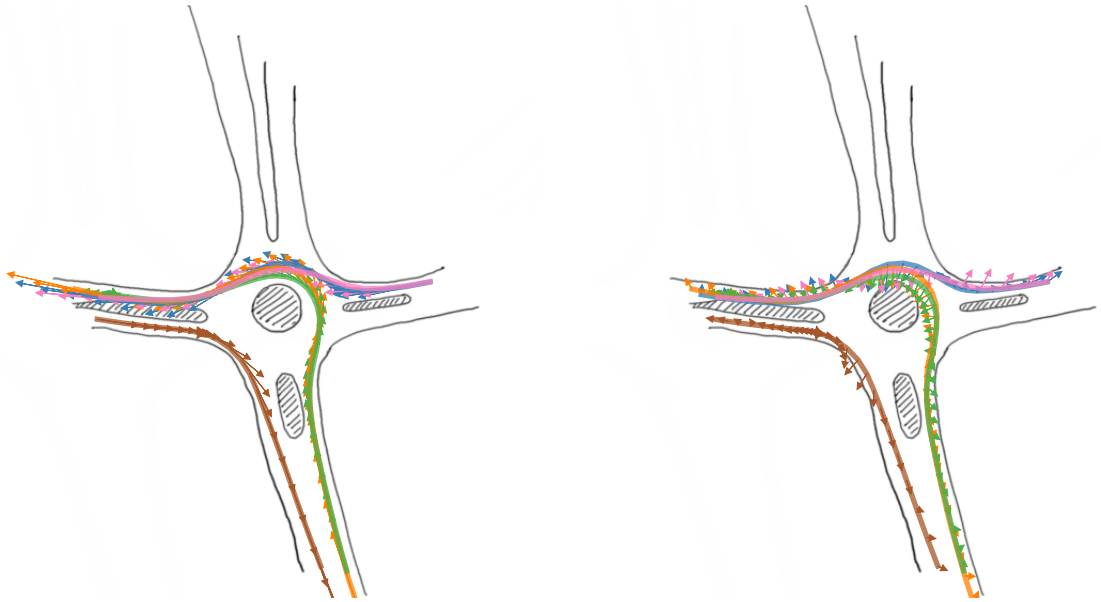


Figure 4.8: rounD dataset: the considered roundabout. Source: Google Maps

road users at 7a.m. We point out that the speed limit is 50km/h. In total, the dataset contains trajectories, velocities and accelerations for $N_0 = 348$ individual road users that passed through this roundabout during this period, recorded every 0.04s. The number of measurements for each curve varies from 131 to 1265. We rescale the measurement times for each of the 348 curves such that the first measurement corresponds to $t = 0$ and the last one to $t = 1$. Figure 4.9 presents a random sample of five observations extracted from the data. In order to have comparable results, we remove the pedestrians and the bikes from the data. Moreover, curves with less than 200 or more than 800 measurements are also removed. The curves with less than 200 points are probably the road users that are present when the recording starts (or ends), and thus, their trajectory will not be complete. The curves with more than 800 are likely to be trajectories with inconsistency. Finally, there are 328 remaining observations in the dataset. We aim to provide a clustering and give some physical interpretation of the clusters. Thus, our clustering procedure `fCUBT` is run on the cleaned data. We chose `n_comp` $= 1$ for both the growing and joining steps and we set $K_{max} = 3$ and `minsize` $= 25$. It returns 19 groups. Figure 4.10 presents an example of cluster we obtain. We remark that the trajectories with different entering and exiting of the roundabout are well split into different clusters. We point out that all atypical trajectories (*e.g.* people that turn around) are gather into a unique cluster. Some of trajectories that have the same enterings and exitings may be split into different clusters. This fact is due to different velocity and acceleration profiles; in particular, to differentiate the vehicles that have to stop when they arrive at the intersection from those that do not (see Figure 4.11).

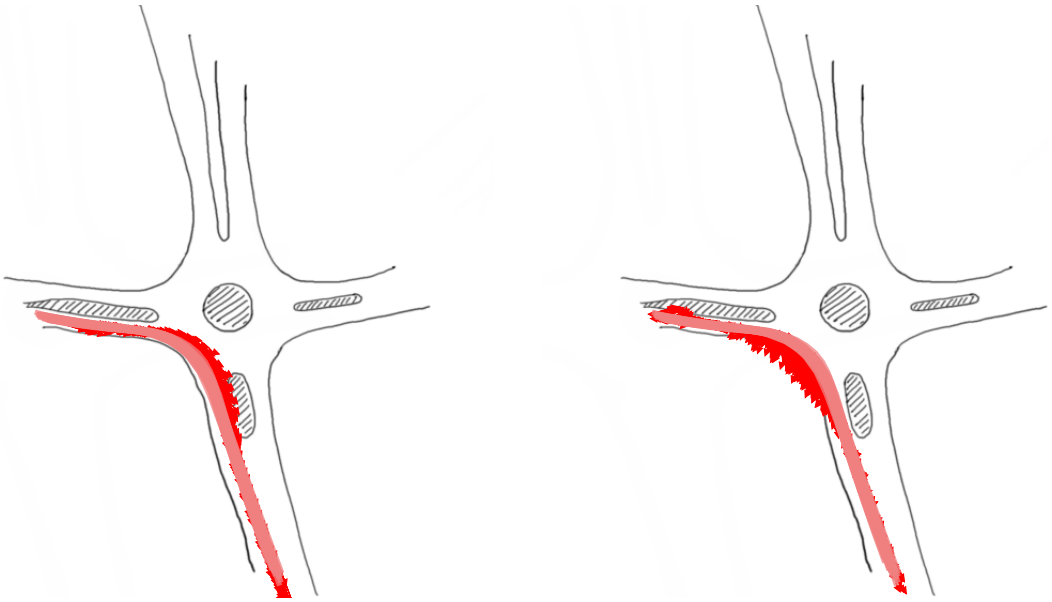


(a)



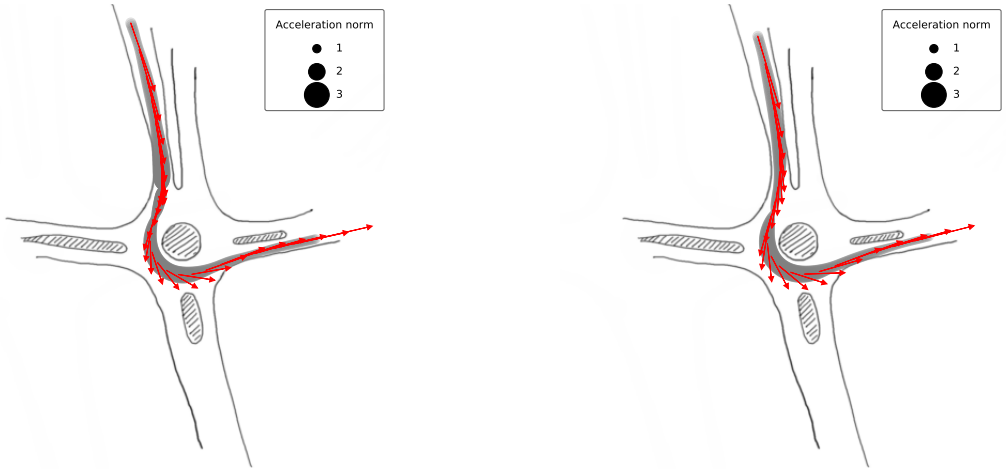
(b) (c)

Figure 4.9: rounD dataset illustration: a sample of five trajectories (a). The trajectories in the roundabout reference frame where the arrows represent the magnitude and direction of the velocity (b) and acceleration (c).



(a) (b)

Figure 4.10: rounD dataset: An example of a cluster found using the `fCUBT` method. The trajectories are plotted in the roundabout reference frame. Each red curve represents a trajectory of an observation and the red arrows represent the magnitude and direction of the velocity (a) and acceleration (b).



(a) (b)

Figure 4.11: rounD dataset: Two different clusters with similar trajectory shape but with different velocity and acceleration profiles: (a) Stop when arriving at the roundabout. (b) Do not stop when arriving at the roundabout.

4.6 Extension to images

The fCUBT algorithm was introduced above for multivariate functional data which could be defined on different domains, possibly of different dimensions. In this section, we present the results of a simulation experiment with a process X with two components, one defined on a compact interval on the real line, the other one defined on a square in the plane. In such situations, the univariate fPCA, done for each component for the computation of the MFPCA basis, is replaced by a suitable basis expansion for higher dimensional functions. In particular, the eigendecomposition of image data can be performed using the FCP-TPA for regularized tensor decomposition [1].

Scenario 4. As for the previous scenarios, the number of clusters is fixed at $K = 5$. Moreover, $P = 2$, $I_1 = [0, 1]$ and $I_2 = [0, 1] \times [0, 1]$. A sample of $N = 500$ curves is simulated according to the following model, for $s, t \in [0, 1]$:

$$\begin{aligned} X_1(t) &= a\phi_1(t) + b\phi_2(t) + c\phi_3(t), \\ X_2(s,t) &= d\phi_1(s)\phi_1(t) + e\phi_1(s)\phi_2(t) + f\phi_2(s)\phi_1(t) + g\phi_2(s)\phi_2(t), \end{aligned}$$

where ϕ_k's are the eigenfunctions of the Wiener process which are defined by

$$\phi_k(t) = \sqrt{2}\sin\left(\left(k - \frac{1}{2}\right)\pi t\right), \quad k = 1, 2, 3.$$

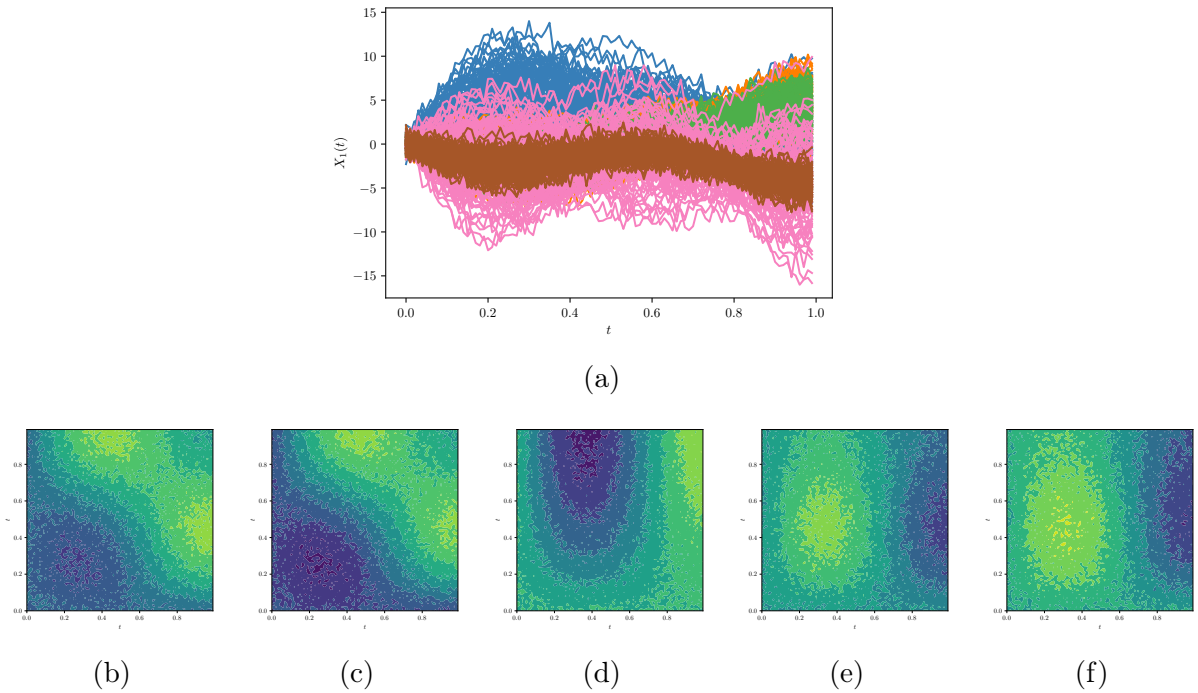
The coefficients a, b, c, d, e, f, g are random normal variables with parameters defined in Table 4.2. The mixing proportions are taken to be equal. The noisy curves are observed on 100 equidistant points and the noisy images are observed on a 2-D grid of 100×100 points. The measurement errors are introduced as in (4.8). The errors for X_1 are independent zero-mean Gaussian variables of variance $\sigma^2 = 0.05$, while the errors for X_2 are bivariate zero-mean Gaussian vectors with independent components of variance $\sigma^2 = 0.05$. The experiment was repeated 500 times.

By construction, the clusters cannot be retrieved using only the noisy curves or the noisy images. Thus, the clustering algorithm has to considered these features for the grouping. Examples of realizations from this simulation experiment are shown in Figure 4.12.

The results of the fCUBT procedure are given in the Table 4.3 for both the estimated number of clusters and the ARI. We remark that in more than half of the cases, our algorithm estimates the number of clusters correctly. However, as cluster 4 and 5 are hard to discriminate, it returns four estimated clusters. This phenomenon is also reflected in the

Scenario 4.	Coefficients (mean / std)						
	a	b	c	d	e	f	g
Cluster 1	$3, 0.5$	$2, 1.66$	$1, 1.33$	$4, 1$	$0, 0.5$	$0, 0.1$	$-2, 0.05$
Cluster 2	$1, 0.5$	$-2, 1$	$0, 1$	$4, 0.8$	$0, 0.7$	$0, 0.08$	$-2, 0.07$
Cluster 3	$1, 0.4$	$-2, 0.8$	$0, 0.8$	$-3, 1$	$-4, 0.5$	$0, 0.1$	$0, 0.05$
Cluster 4	$-2, 1$	$0, 2$	$-1, 2$	$0, 0.1$	$2, 0.1$	$0, 0.05$	$0, 0.025$
Cluster 5	$-2, 0.2$	$0, 0.5$	$-1, 0.5$	$0, 2$	$2, 1$	$0, 0.2$	$1, 0.1$

Table 4.2: Coefficients for $X_1(t)$ and $X_2(s,t)$.



(a)

(b) (c) (d) (e) (f)

Figure 4.12: Examples of simulated data for Scenario 4 : (a) 500 realizations of the process X_1. (b)–(f) One realization of each of the clusters from the process X_2.

	Number of clusters K			
	4	5	6	9
`fCUBT`	0.43	0.522	0.046	0.002

(a) Number of clusters selected for `fCUBT` for 500 simulations as a percentage.

	Quantile										
	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
`fCUBT`	0.70	0.76	0.77	0.77	0.78	0.78	0.96	0.98	0.99	0.99	1.0

(b) Quantile of the ARI for 500 simulations.

Table 4.3: Results for the Scenario 4.

ARI results. In fact, the ARI presents a bimodal distribution where one mode is centered around 0.98 and the other one around 0.77. It appears that when the ARI is around 0.98, the number of clusters is well estimated, while when the ARI is close to 0.77, it is not. Nevertheless, even if in some replications, the number of clusters is wrongly estimated, overall the results are good. This simulation experiment provides strong evidence that our algorithm could also be used in such complex situations to which, apparently, the existing clustering algorithms have not yet been extended.

4.7 Conclusion

The `fCUBT` algorithm has been proposed which is a model-based clustering method for functional data based on unsupervised binary trees. It works both on univariate and multivariate functional data defined in possibly different, multidimensional domains. The method is particularly suitable for finding the correct number of clusters within the data with respect to the model assumption. When the complete tree has been grown, `fCUBT` can be used for supervised classification. The open-source implementation can be accessed on Github[1] as well as scripts to reproduce the simulation and real-data analysis[2].

Appendix

1. https://github.com/StevenGolovkine/FDApy
2. https://github.com/StevenGolovkine/fcubt

4.A Proofs

Proof of the lemma 9. Using the linearity of the inner product, we may rewrite for each $j \geq 1, c_j$ as

$$c_j = \langle\langle X - \mu, \psi_j \rangle\rangle = \sum_{k=1}^{K} \langle\langle \mu_k, \psi_j \rangle\rangle \mathbf{1}_{\{Z=k\}} - \langle\langle \mu, \psi_j \rangle\rangle + \sum_{l \geq 1} \xi_l \langle\langle \phi_l, \psi_j \rangle\rangle.$$

Since a linear combination of independant Gaussian distributions is still Gaussian, the conditional distribution $c_j \mid Z = k$ has a Gaussian distribution for all $k \in \{1, \dots, K\}$, $j \geq 1$. Moreover, by the definition of X and the linearity of the inner product, for any $i, j \geq 1$ and $k \in \{1, \dots, K\}$,

$$\begin{aligned}\mathbb{E}\left(c_j \mid Z = k\right) &= \sum_{k'=1}^{K} \langle\langle \mu_{k'}, \psi_j \rangle\rangle \mathbb{E}\left(\mathbf{1}_{\{Z=k'\}} \mid Z = k\right) - \langle\langle \mu, \psi_j \rangle\rangle + \sum_{l \geq 1} \mathbb{E}\left(\xi_l \mid Z = k\right) \langle\langle \phi_l, \psi_j \rangle\rangle \\ &= \langle\langle \mu_k - \mu, \psi_j \rangle\rangle.\end{aligned}$$

Next, for any $i, j \geq 1$ and $k \in \{1, \dots, K\}$,

$$\begin{aligned}\mathrm{Cov}\left(c_i, c_j \mid Z = k\right) &= \mathbb{E}\left(c_i c_j \mid Z = k\right) - \mathbb{E}\left(c_i \mid Z = k\right) \mathbb{E}\left(c_j \mid Z = k\right) \\ &= \sum_{p=1}^{P} \sum_{q=1}^{P} \int_{I_p} \int_{I_q} \mathbb{E}\left((X - \mu)_p(s_p)(X - \mu)_q(t_q) \mid Z = k\right) \psi_{p,i}(s_p) \psi_{q,j}(t_q) ds_p dt_q \\ &\qquad - \langle\langle \mu_k - \mu, \psi_i \rangle\rangle \langle\langle \mu_k - \mu, \psi_j \rangle\rangle.\end{aligned}$$

By definition, for any $1 \leq p, q \leq P$,

$$\begin{aligned}&\mathbb{E}\left((X - \mu)_p(s_p)(X - \mu)_q(t_q) \mid Z = k\right) \\ &\quad = \sum_{k'=1}^{K} \sum_{k''=1}^{K} \mu_{p,k'}(s_p) \mu_{q,k''}(t_q) \mathbb{E}\left(\mathbf{1}_{\{Z=k'\}} \mid Z = k\right) \mathbb{E}\left(\mathbf{1}_{\{Z=k''\}} \mid Z = k\right) \\ &\qquad + \sum_{k'=1}^{K} \mu_{p,k'}(s_p) \sum_{j \geq 1} \phi_{q,j}(t_q) \mathbb{E}\left(\xi_j \mathbf{1}_{\{Z=k'\}} \mid Z = k\right) \\ &\qquad + \sum_{k''=1}^{K} \mu_{q,k''}(t_q) \sum_{l \geq 1} \phi_{p,l}(s_p) \mathbb{E}\left(\xi_l \mathbf{1}_{\{Z=k''\}} \mid Z = k\right) \\ &\qquad + \sum_{j \geq 1} \sum_{l \geq 1} \phi_{p,j}(s_p) \phi_{q,l}(t_q) \mathbb{E}\left(\xi_j \xi_l \mid Z = k\right)\end{aligned}$$

$$
\begin{aligned}
&- \mu_q(t_q) \sum_{k'=1}^{K} \mu_{p,k'}(s_p) \mathbb{E}\left(\mathbf{1}_{\{Z=k'\}} \mid Z=k\right) \\
&- \mu_p(s_p) \sum_{k''=1}^{K} \mu_{q,k''}(t_q) \mathbb{E}\left(\mathbf{1}_{\{Z=k''\}} \mid Z=k\right) \\
&- \mu_p(s_p) \sum_{l\geq 1} \phi_{q,l}(t_q) \mathbb{E}\left(\xi_l \mid Z=k\right) \\
&- \mu_q(t_q) \sum_{l\geq 1} \phi_{p,l}(s_p) \mathbb{E}\left(\xi_l \mid Z=k\right) \\
&+ \mu_p(s_p)\mu_q(t_q) \\
= \mu_p(s_p)\mu_q(t_q) &+ \mu_{p,k}(s_p)\mu_{q,k}(t_q) - \mu_p(s_p)\mu_{q,k}(t_q) - \mu_{p,k}(s_p)\mu_q(t_q) \\
&+ \sum_{l\geq 1} \sigma_{kl}^2 \phi_{p,l}(s_p)\phi_{q,l}(t_q) \\
= \left(\mu_{p,k}(s_p) - \mu_p(s_p)\right)&\left(\mu_{q,k}(t_q) - \mu_q(t_q)\right) + \sum_{l\geq 1} \sigma_{kl}^2 \phi_{p,l}(s_p)\phi_{q,l}(t_q).
\end{aligned}
$$

Thus,

$$
\begin{aligned}
\mathrm{Cov}\left(c_i, c_j \mid Z=k\right) &= \langle\!\langle \mu_k - \mu, \psi_i \rangle\!\rangle \langle\!\langle \mu_k - \mu, \psi_j \rangle\!\rangle + \sum_{l\geq 1} \sigma_{kl}^2 \langle\!\langle \phi_l, \psi_i \rangle\!\rangle \langle\!\langle \phi_l, \psi_j \rangle\!\rangle \\
&\quad - \langle\!\langle \mu_k - \mu, \psi_i \rangle\!\rangle \langle\!\langle \mu_k - \mu, \psi_j \rangle\!\rangle \\
&= \sum_{l\geq 1} \sigma_{kl}^2 \langle\!\langle \phi_l, \psi_i \rangle\!\rangle \langle\!\langle \phi_l, \psi_j \rangle\!\rangle.
\end{aligned}
$$

Taking $i = j$ in the conditional covariance, we deduce

$$
\tau_{kj}^2 = \mathrm{Var}\left(c_j \mid Z=k\right) = \sum_{l\geq 1} \sigma_{kl}^2 \langle\!\langle \phi_l, \psi_j \rangle\!\rangle^2.
$$

For the marginal distribution of the c_j, the zero-mean is obtained as follows:

$$
\mathbb{E}(c_j) = \sum_{k=1}^{K} \mathbb{P}(Z=k) \mathbb{E}\left(c_j \mid Z=k\right) = \sum_{k=1}^{K} p_k \langle\!\langle \mu_k - \mu, \psi_j \rangle\!\rangle = 0.
$$

For the marginal covariance, we can write

$$
\mathrm{Cov}(c_i, c_j) = \mathbb{E}(c_i c_j)
$$

$$
\begin{aligned}
&= \sum_{k=1}^{K} p_k \mathbb{E}\left(c_i c_j \mid Z = k\right) \\
&= \sum_{k=1}^{K} p_k \left(\mathrm{Cov}\left(c_i, c_j \mid Z = k\right) + \mathbb{E}\left(c_i \mid Z = k\right)\mathbb{E}\left(c_j \mid Z = k\right)\right) \\
&= \sum_{k=1}^{K} p_k \left(\sum_{l\geq 1} \langle\!\langle \phi_l, \psi_i \rangle\!\rangle \langle\!\langle \phi_l, \psi_j \rangle\!\rangle \sigma_{kl}^2 + \langle\!\langle \mu_k - \mu, \psi_i \rangle\!\rangle \langle\!\langle \mu_k - \mu, \psi_j \rangle\!\rangle\right).
\end{aligned}
$$

This concludes the proof. □

Proof of Lemma 10. Given $\{\psi_j\}_{j\geq 1}$ an orthonormal basis of $\mathcal{H}$, X can be written in the form $X(t) = \sum_{j\geq 1} c_j \psi_j(t), t \in \mathcal{T}$, with random variables $c_j = \langle\!\langle X - \mu, \psi_j \rangle\!\rangle$. Then

$$
\begin{aligned}
\left\| X - X_{\lceil J \rceil} \right\|^2 &= \left\| \sum_{j=J+1}^{\infty} c_j \psi_j \right\|^2 \\
&= \sum_{p=1}^{P} \int_{I_p} \left(\sum_{j=J+1}^{\infty} c_j \psi_{p,j}(t) \right) \left(\sum_{j'=J+1}^{\infty} c_{j'} \psi_{p,j'}(t) \right) dt \\
&= \sum_{j=J+1}^{\infty} \sum_{j'=J+1}^{\infty} c_j c_{j'} \langle\!\langle \psi_j, \psi_{j'} \rangle\!\rangle \\
&= \sum_{j=J+1}^{\infty} c_j^2.
\end{aligned}
$$

Moreover,

$$
\mathbb{E}\left(\sum_{j\geq 1} c_j^2 \right) = \mathbb{E}\left(\sum_{j\geq 1} \langle\!\langle X - \mu, \psi_j \rangle\!\rangle^2 \right) = \mathbb{E}\left(\langle\!\langle X - \mu \rangle\!\rangle^2 \right) < \infty.
$$

From this, and the fact that the remainder of a convergent series tends to zero, we have

$$
\mathbb{E}\left(\left\| X - X_{\lceil J \rceil} \right\|^2 \right) = \mathbb{E}\left(\sum_{j=J+1}^{\infty} c_j^2 \right) \xrightarrow[J\to\infty]{} 0.
$$

□

Proof of Lemma 11. First, let us note that, since the bases are orthogonal, the minimization of the truncation error for a given J is equivalent to the maximization of the sum of

the variances $\langle\!\langle \Gamma\psi_j, \psi_j \rangle\!\rangle$, $1 \leq j \leq J$. Moreover, for each $j \geq 1$, we have

$$
\begin{aligned}
&\mathbb{E}\left[\langle\!\langle X-\mu, \varphi_j \rangle\!\rangle^2\right] \\
&\quad= \mathbb{E}\left[\left(\sum_{p=1}^{P}\langle (X-\mu)_p, \varphi_{p,j}\rangle\right)\left(\sum_{q=1}^{P}\langle (X-\mu)_q, \varphi_{q,j}\rangle\right)\right] \\
&\quad= \mathbb{E}\left[\left(\sum_{p=1}^{P}\int_{I_p}(X-\mu)_p(s_p)\varphi_{p,j}(s_p)ds_p\right)\left(\sum_{q=1}^{P}\int_{I_q}(X-\mu)_q(t_q)\varphi_{q,j}(t_q)dt_q\right)\right] \\
&\quad= \sum_{p=1}^{P}\int_{I_p}\sum_{q=1}^{P}\int_{I_q}\mathbb{E}\left[(X-\mu)_p(s_p)\varphi_{p,j}(s_p)(X-\mu)_q(t_q)\varphi_{q,j}(t_q)\right] \\
&\qquad\qquad\qquad\qquad \times \varphi_{p,j}(s_p)\varphi_{q,j}(t_q)ds_p dt_q \\
&\quad= \sum_{p=1}^{P}\int_{I_p}\sum_{q=1}^{P}\int_{I_q} C_{p,q}(s_p,t_q)\varphi_{p,j}(s_p)\varphi_{q,j}(t_q)ds_p dt_q \\
&\quad= \sum_{q=1}^{P}\int_{I_q}\sum_{p=1}^{P}\langle C_{p,q}(\cdot,t_q), \varphi_{p,j}(\cdot)\rangle\varphi_{q,j}(t_q)dt_q \\
&\quad= \sum_{q=1}^{P}\int_{I_q}\langle\!\langle C_{\cdot,q}(\cdot,t_q), \varphi_j(\cdot)\rangle\!\rangle\varphi_{q,j}dt_q \\
&\quad= \sum_{q=1}^{P}\int_{I_q}\left(\Gamma\varphi_j\right)_q(t_j)\varphi_{q,j}(t_q)dt_q \\
&\quad= \sum_{q=1}^{P}\langle\left(\Gamma\varphi_j\right)_q(t_j), \varphi_{q,j}(t_q)\rangle \\
&\quad= \langle\!\langle \Gamma\varphi_j, \varphi_j \rangle\!\rangle
\end{aligned}
$$

Since the MFPCA basis is characterized by the property (4.4), for any orthonormal basis $\{\psi_j\}_{j\geq 1}$, we necessarily have

$$
\sum_{j=1}^{J}\mathrm{Var}(\langle\!\langle X-\mu, \varphi_j \rangle\!\rangle) = \sum_{j=1}^{J}\langle\!\langle \Gamma\varphi_j, \varphi_j \rangle\!\rangle \geq \sum_{j=1}^{J}\langle\!\langle \Gamma\psi_j, \psi_j \rangle\!\rangle.
$$

This concludes the proof. □

Chapter 5

Implementation of the methods

Abstract: *As a practical contribution, we propose the* ***Python*** *package,* `FDApy`*, as an implementation tool for the functional data methods we developed. This package provides several modules for the functional data analysis. It includes classes for different dimensional data as well as irregularly sampled functional data. A simulation toolbox is also provided. It might be used to simulate different clusters of functional data. Some methodologies to handle these data are implemented, such as dimension reduction and clustering. New methods can be easily added. The package is publicly available on the Python Package Index and Github. This article has been submit for publication in the* Journal of Statistical Software.

Contents

5.1 Introduction 190
5.2 Classes of functional data 192
5.3 Data used in the examples 194
5.4 Manipulation of functional data objects 194
5.4.1 Creation of objects 195
5.4.2 Access to the instance variables 197
5.4.3 Plotting 198
5.4.4 Data simulation 198
5.5 Parameters estimation 203
5.5.1 Curves denoising 203
5.5.2 Mean and covariance estimation 204
5.6 MFPCA 206
5.6.1 Methodological background 206
5.6.2 Implementation 207

5.7 fCUBT 210
5.7.1 Methodological background 210
5.7.2 Implementation 211
5.8 Conclusion 214

5.1 Introduction

In order to apply FDA to real datasets, there is a need for appropriate softwares with up-to-date methodological implementation and easy addition of new theoretical developments. Currently, the most widely known software for FDA is the **R** package `fda` [112], based on work cited in [108, 111]. Usually, **R** packages for FDA are specific to one method. For example, one may cite `FDboost` [15] and `refund` [57] for regression and classification, `funFEM` [9], `funHDDC` [119] and `funLBM` [14] for clustering or `fdasrvf` [136] and `fdakma` [103] for functional data registration, *etc.* Most of these packages are built upon `fda`. However, in most packages, the functional data are restricted to univariate ones that are well described by their coefficients in a given basis of functions. The `funData` package [66] has recently been released. It aims to provide a unified framework to handle univariate and multivariate functional data defined on different dimensional domains. Sparse functional data are also considered. The `MFPCA` [67] package is currently the only one built on top of the `funData` package. It implements MFPCA for data defined on different dimensional domains [64].

Concerning the **Python** community, there are only few packages that are related to FDA. One may cite `sktime` [93] and `tslearn` [130] that provide tools for the analysis of time series as a `scikit-learn` compatible API. Thus, they implement specific time series methods such as DTW-based ones or shapelets learning. The only one that develops specific methods for FDA is `scikit-fda` [22]. In particular, it implements diverse registration techniques as well as statistical data depth for functional data. However, most of the methods are for one-dimensional data and they only accept multivariate functional data defined on the same unidimensional domain.

The `FDApy` package implements methods to handle functional data in **Python** based on an object-oriented approach, in the spirit of `funData`. In particular, it provides classes to manipulate dense, irregularly and multivariate functional data defined on one or higher dimensional domains. A large simulation toolbox, based on basis decomposition, is pro-

vided. It allows parameters for different clusters simulation to be configured within the data. An implementation of MFPCA for data defined on different domains, as described in [64], is implemented. Moreover, the fCUBT algorithm [58], used to create partition in the data, is also available. All methods are implemented using the defined classes. The package is publicly available on Github[1] and the Python Package Index[2].

In the general case, the data consist of independent trajectories of a vector-valued stochastic process $X = (X_1, \ldots, X_P)^\top$, $P \geq 1$. For each $1 \leq p \leq P$, let $I_p \subset \mathbb{R}^{d_p}$ with $d_p \geq 1$, as for instance, $I_p = [0,1]^{d_p}$. The realizations of each coordinate $X_p : I_p \to \mathbb{R}$ are assumed to belong to $\mathcal{L}^2(I_p)$, the Hilbert space of squared-integrable, real-valued functions defined on I_p. Thus, X is a stochastic process indexed by $\boldsymbol{t} = (t_1, \ldots, t_P)$ belonging to the $P-$fold Cartesian product $\boldsymbol{I} \coloneqq I_1 \times \cdots \times I_P$ and taking values in the $P-$fold Cartesian product space $\mathcal{H} \coloneqq \mathcal{L}^2(I_1) \times \cdots \times \mathcal{L}^2(I_P)$. In practice, realizations of functional data are only obtained on a finite grid and possibly with noise. Let us consider N curves $X^{(1)}, \ldots, X^{(n)}, \ldots, X^{(N)}$ generated as a random sample of the P-dimensional stochastic process X with continuous trajectories. For each $1 \leq n \leq N$, and given a vector of positive integers $\boldsymbol{M}_n = (M_{n,1}, \ldots, M_{n,P}) \in \mathbb{R}^P$, let $T_{\boldsymbol{m}}^{(n)} = (T_{m_1}^{(n)}, \ldots, T_{m_P}^{(n)}), 1 \leq m_p \leq M_{n,p}, 1 \leq p \leq P$, be the random observation times for the curve $X^{(n)}$. These times are obtained as independent copies of a variable $\boldsymbol{T}$ taking values in $\boldsymbol{I}$. The vectors $\boldsymbol{M}_1, \ldots, \boldsymbol{M}_N$ represent an independent sample of an integer-valued random vector $\boldsymbol{M}$ with expectation $\boldsymbol{\mu}_{\boldsymbol{M}}$ which increases with N. We assume that the realizations of X, $\boldsymbol{M}$ and $\boldsymbol{T}$ are mutually independent. The observations associated with a curve, or trajectory, $X^{(n)}$ consist of the pairs $(Y_{\boldsymbol{m}}^{(n)}, T_{\boldsymbol{m}}^{(n)}) \in \mathbb{R}^P \times \boldsymbol{I}$, where $\boldsymbol{m} = (m_1, \ldots, m_P), 1 \leq m_p \leq M_{n,p}$, $1 \leq p \leq P$, and $Y_{\boldsymbol{m}}^{(n)}$ is defined as

$$Y_{\boldsymbol{m}}^{(n)} = X^{(n)}(T_{\boldsymbol{m}}^{(n)}) + \varepsilon_{\boldsymbol{m}}^{(n)}, \quad 1 \leq n \leq N, \tag{5.1}$$

with the $\varepsilon_{\boldsymbol{m}}^{(n)}$ being independent copies of a centered random vector $\boldsymbol{\varepsilon} \in \mathbb{R}^P$ with finite variance. We use the notation $X^{(n)}(\boldsymbol{t})$ for the value at $\boldsymbol{t}$ of the realization $X^{(n)}$ of X. Univariate functional data refers to the case where $P = 1$.

The remainder of the paper is organized as follows. In Section 5.2, we introduce the classes for an object-oriented implementation of functional data. Section 5.3 describes the data we used as examples. In Section 5.4, we presents the creation and manipulation of functional data objects. Sections 5.5, 5.6 and 4.4 then demonstrate some methods that the

1. https://github.com/StevenGolovkine/FDApy
2. https://pypi.org/project/FDApy/

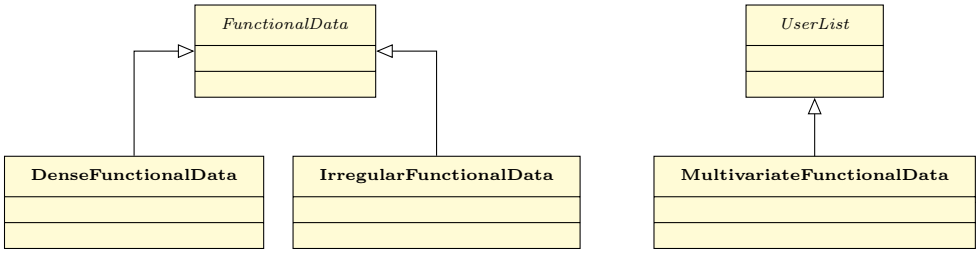


Figure 5.1: Representation of the main classes.

package implements: the estimation of components, MFPCA and the fCUBT algorithm used to find a partition of the sampled data.

5.2 Classes of functional data

The representation of functional data is done using two classes, that both extend an abstract class `FunctionalData`:

1. Class `DenseFunctionalData` represents dense functional data of arbitrary dimension (one for curves, two for images, *etc.*) on a common set of observation points $t_1, \ldots, t_M$ for all observations. It may have missing values within the data.

2. Class `IrregularFunctionalData` represents irregularly sampled data of arbitrary dimension on different sets of observation points. The number and the location of the sampling points vary between observations. It must not have missing values within the data.

Finally, the implementation of the class `MultivariateFunctionalData`, is different because it does not extend the class `FunctionalData` but the `UserList` one. Thus, an instance of `MultivariateFunctionalData` is defined as a list of P elements from the `DenseFunctionalData` and/or `IrregularFunctionalData` classes that may be defined on different dimensional domains (*e.g.* curves and images). A diagram of the classes is given in Figure 5.1.

Remark. In practice, the difference between dense and irregularly sampled functional data can be tricky. By design, dense functional data are assumed to be sampled on the complete grid $\mathcal{T} = \{t_1, \ldots, t_M\}$ and measurement errors may exist. Taking data from sensors as an example, observations are recorded at a given sampling rate and are timestamped but some anomalies may happen during the recording process. While for an irregularly

sampled functional data, we assume that the curves are observed at different sampling points with potentially different numbers of points. This is usually the case in medical studies such as growth curves analysis because one cannot expect that the individuals are measured at the exact same time.

The `DenseFunctionalData` and `IrregularFunctionalData` classes represent the data in a similar way: the instance variable `argvals` contains the sampling points and the instance variable `values` represents the data. In the case of dense functional data, the `argvals` is a dictionary whese each entry contains a `numpy` array that represents the common sampling points for a given dimension, while `values` is a `numpy` array containing the observations. In the case of one-dimensional data sampled on a grid with M points, `argvals` contains only one entry as an array of shape $(M,)$ and `values` is an array of dimension (N, M) where each row is an observation. For two-dimensional observations with $M_1 \times M_2$ sampling points, `argvals` contains two entries, the first being an array of shape $(M_1,)$ and the second an array of shape $(M_2,)$ and `values` is an array of dimension (N, M_1, M_2) where the first coordinate gives the observation. The higher dimensional data are represented by adding an entry in the `argvals` dictionary and a dimension in the `values` array. For irregularly sampled functional data, both `argvals` and `values` are dictionaries. The entries of `argvals` are dictionaries where each entry consists of the sampling points for a particular observation. In a similar way, each entry of the `values` dictionary represents an observation. For one-dimensional irregularly sampled functional data, `argvals` contains one entry which is a dictionary of size N containing the sampling points as array of shape $(M_n,), 1 \leq n \leq N$ and `values` is a dictionary with N entries containing the observations as arrays of shape $(M_n,), 1 \leq n \leq N$. For higher dimensional data, each entry of the `argvals` dictionary represents a dimension of the process and contains another dictionary with N entries for the sampling points. Likewise, the `values` dictionary has N entries and every one of them is an array of shape $(M_{n,1}, M_{n,2}, \dots), 1 \leq n \leq N$.

Finally, the `MultivariateFunctionalData` class inherits from the `UserList` class, and thus gathers P instances of `DenseFunctionalData` and/or `IrregularFunctionalData` as a list. As a result, this class has access to all the methods applicable to lists such as `append`, `extend`, `pop`, *etc.* Given a specific dataset, instances of the different classes are called `DenseFunctionalData`, `IrregularFunctionalData` or `MultivariateFunctionalData` objects. In the following, the generic term, *functional data object*, will refer to instances of all the three classes.

5.3 Data used in the examples

We will consider two datasets in the code examples. The first one will be the **Canadian weather** data, which is presented in the textbook by Ramsay and Silverman [108] and available in their **R** package `fda` [112]. The second dataset is the **CD4 cell count** dataset, used in [56], and available in the **R** package `refund` [57]. As both examples are one-dimensional data, higher dimensional datasets, in particular images ones, will be simulated using the simulation toolbox provided in the package.

The **Canadian weather** dataset contains daily recording of the temperature (in degree Celsius) and the precipitation (in millimeters) for $N = 35$ Canadian cities spread across the country and averaged over the years 1960 to 1994. The daily temperature data will be used as an example of `DenseFunctionalData` defined on a one-dimensional domain. We will add the daily precipitation records to the temperature ones in order to create a `MultivariateFunctionalData` object with elements defined on different one-dimensional domains ($I_1 = [1, 364]$ for the temperature and $I_2 = [1, 363]$ for the precipitation).

From the MACS (Multicenter AIDS Cohort Study), the **CD4 cell count** dataset collects the number of CD4 cells per milliliter of blood of $N = 366$ participants. CD4 cells are a particular type of white blood cell and are key components of the immune system. HIV attacks the CD4 cells in the patient's blood. Thus, the count of CD4 cells can be viewed as a measure of the disease progression. For this dataset, the number of CD4 cells are measured roughly twice a year and centered at the time of seroconversion, which is the time that HIV becomes detectable. For every individual, the number of measurements varies between 1 to 11 over a period of 18 months before and 42 months after seroconversion. The sampling points are different between observations. We will use this dataset as an example of `IrregularFunctionalData`.

5.4 Manipulation of functional data objects

With the help of the two example datasets, this section will present how to create and manipulate a functional data object. In particular, we review the different instance variables used to extract information from the data. We also present methods to modify and plot functional data objects. General methods, such as the computation of the mean or covariance, for `MultivariateFunctionalData` objects usually call the corresponding methods

for each individual and concatenate the results appropriately. For all the code examples, we assume that the correct functions from the `FDApy` package are loaded as well as the packages `numpy` and `pandas` using the following code snippet:

```
import numpy as np
import pandas as pd
```

5.4.1 Creation of objects

Assuming the Canadian temperature data is stored in a *temperature.csv* file and the Canadian precipitation data in a *precipitation.csv* file, the following code loads the data into `pandas` dataframes and creates `DenseFunctionalData` instances from them. We explicitly named the dimension of the observations.

```
temperature = pd.read_csv('temperature.csv', index_col=0)

argvals = pd.factorize(temperature.columns)[0]
values = np.array(temperature)
dailyTemp = DenseFunctionalData({'input_dim_0': argvals}, values)

# Creation of the precipitation object
precipitation = pd.read_csv('precipitation.csv', index_col=0)

argvals = pd.factorize(precipitation.columns)[0]
values = np.array(precipitation)
dailyPrec = DenseFunctionalData({'input_dim_0': argvals}, values)
```

Given multiple functional data objects, the creation of `MultivariateFunctionalData` instances is done by passing a list of objects to the constructor method.

```
canadWeather = MultivariateFunctionalData([dailyTemp, dailyPrec])
```

The construction of an `IrregularFunctionalData` instance is similar, except that the dictionaries for `argvals` and `values` must contain an entry for each observation of the data. We consider that the CD4 cell count data are stored in a *cd4.csv* file containing a matrix representing the CD4 counts for each patient on the common grid

of all sampling points and the missing values are coded as `NA`. Thus, the following code extracts only the non-missing values for each patient and construct an instance of `IrregularFunctionalData`.

```
cd4 = pd.read_csv('cd4.csv', index_col=0)

all_argvals = cd4.columns.astype(np.int64)
argvals = {idx: np.array(all_argvals[~np.isnan(row)])
                for idx, row in enumerate(cd4.values)}
values = {idx: row[~np.isnan(row)] for idx, row in enumerate(cd4.values)}
cd4counts = IrregularFunctionalData({'input_dim_0': argvals}, values)
```

Remark. Two loaders are included within the package: `read_csv` and `read_ts`. These methods can be used to load already well formatted data from csv or ts files. In particular, the ts files are used in the UEA & UCR Time Series Classification Repository[3]. These functions are wrapper functions of the above code snippets. Nonetheless, multivariate functional data cannot be imported this way.

Basic information about the functional data object is printed on the standard output when the object is called in the command line. For example, for the temperature dataset, the output will be:

```
dailyTemp
```

```
Univariate functional data object with 35 observations on a 1-dimensional
support.
```

The outputs are similar for instances of the other types of functional data objects.

For a dense and irregular functional object, a subset of the data can be extracted using the convenient way to substract the object provided in **Python**. For example, in order to get the observations from 5 to 12 from the temperature data, we may write:

```
dailyTemp[5:13]
```

```
Univariate functional data object with 8 observations on a 1-dimensional
support.
```

3. http://bit.ly/ucr_repository

Note that this will not work with `MultivariateFunctionalData` instances. This subsetting method will instead return the univariate functional data in the list. However, an iterator through the observations of the multivariate functional data is provided as the `get_obs` method.

In regards to Remark 5.2, we implement functions to convert `DenseFunctionalData` instances into `IrregularFunctionalData` instances and to do the reverse operation. The missing values are coded with `np.nan`. The code is thus written:

```
dailyTemp.as_irregular()  # dense to irregular
cd4.as_dense()  # irregular to dense
```

5.4.2 Access to the instance variables

The functional data classes come with multiple instance variables. In **Python**, they can usually be accessed using `instance_name.variable_name`. We will present some of them in the following. Note that, some variables cannot be accessed directly for multivariate functional data and have to be retrieved by looping through its univariate elements.

Of course, the `argvals` and `values` are accessible using `dailyTemp.argvals` and `dailyTemp.values` and show what the user gave to the object constructor. Furthermore, we provide a variable `argvals_stand` with the same shape as `argvals` but with normalized sampling points. The instance variables are the following:

- `n_obs` – number of observations in the object.
- `n_points` – number of sampling points in the object for each dimension as a dictionary. For the multivariate functional data object, it should be a list of P entries. In the case of `IrregularFunctionalData`, the returned number is the mean number of sampling points per observation.
- `n_dim` – input dimension of the functional data (one for curves, two for images, *etc.*). For `MultivariateFunctionalData` objects, is expressed as a list.
- `range_obs` – minimum and maximum values of the observations as a tuple.
- `range_points` – minimum and maximum values of the sampling points as a tuple. The calculation is based on the variable `argvals`.

5.4.3 Plotting

Basic plotting methods for functional data objects are provided in the package. They are built upon the `matplotlib` package. We assume the package is loaded with

```
import matplotlib.pyplot as plt
```

The `plot` method returns an instance of `Axes` from the `matplotlib` library. Thus, all the plotting options relative to ticks, frames and so on, are modifiable using this instance of `Axes`. Customization of the graph parameters, such as colors, linetypes or linewidths for example, can be made by passing the arguments as inputs to the function. The following snippet is used to plot all the temperature curves for all the Canadian weather station data (represented as a `DenseFunctionalData` object),

```
_ = plot(dailyTemp)
plt.xlabel('Days')
plt.ylabel('Temperature')
plt.title('Daily Temperature Data')
plt.show()
```

while a plot of the CD4 cell counts for 10 patients on the log-scale (represented as an `IrregularFunctionalData` object) is given by

```
_ = plot(cd4counts[5:15])
plt.xlabel('Month since seroconversion')
plt.ylabel('CD4 cell counts (log-scale)')
plt.title('CD4 counts for individual 5-14')
plt.show()
```

The plots are shown in Figure 5.2.

5.4.4 Data simulation

Simulation functions are implemented in order to test new methodological developments. The data can be simulated using a truncated version of the Karhunen-Loève representation (class `KarhunenLoeve`) as well as diverse Brownian motions (class `Brownian`) that inherits from the `Simulation` class (see Figure 5.3). An element of the `Simulation` class have two

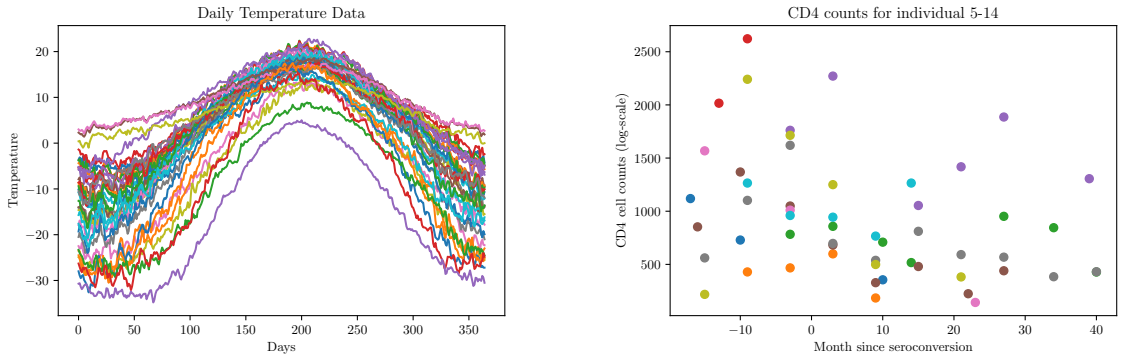


Figure 5.2: Results of the `plot` method for functional data object.

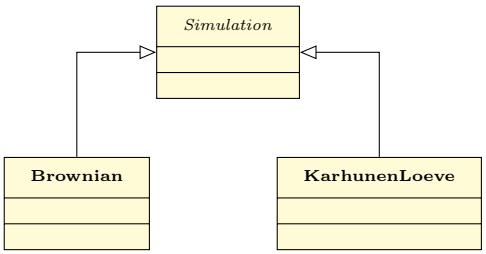


Figure 5.3: Links between classes in the simulation toolbox.

principal instance variables: `basis` that contains the used basis and `data` that contains the simulated observation (after running the fonction `new()`).

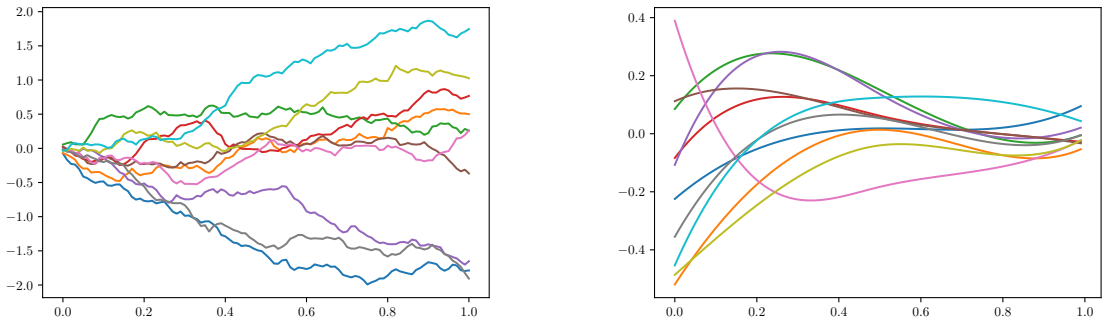
For Brownian motions, three types are implemented: `standard`, `fractional` and `geometric`. For example, we can simulate $N = 10$ realizations of a fractional Brownian motion on the one-dimensional observation grid $\{0, 0.01, \ldots, 1\}$ with a Hurst parameter equal to 0.7 using

```
brownian = Brownian(name='fractional')
brownian.new(n_obs=10, argvals=np.linspace(0, 1, 101), hurst=0.7)
```

The process X has a Karhunen-Loève decomposition. Each of its realizations can be represented using this decomposition, truncated at J coefficients:

$$X^{(n)}(t) = \mu(t) + \sum_{j=1}^{J} \xi_j^{(n)} \phi_j(t), \quad t \in \mathcal{T}, \ n = 1, \ldots, N,$$

with a common mean function μ and an orthonormal basis of functions $\{\phi_j\}_{j=1,\ldots,J}$. The coefficient $\xi_j^{(n)}$ are realizations of random Gaussian variables ξ_j such that $\mathbb{E}(\xi_j) = 0$



(a) (b)

Figure 5.4: Example of simulated data. (a) Brownian motion. (b) Karhunen-Loève expansion

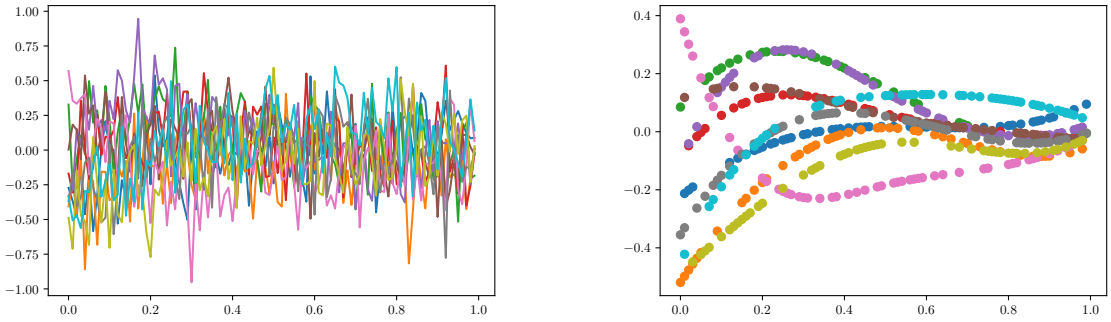
and $\text{Var}(\xi_j) = \lambda_j$ with eigenvalues $\lambda_j \geq 0$ that decrease towards 0. Multiple orthonormal bases are implemented: Legendre polynomials, eigenfunctions of a Wiener process, Fourier series and B-splines basis. The variance of the coefficients can have a linear or exponential decrease or be the eigenvalues of a Wiener process. The user can also set their own. New bases can easily be added.

For example, we can simulate $N = 10$ curves on $I = [0, 1]$, using 5 eigenfunctions from a B-splines basis on I and eigenvalues with exponential decrease:

```
kl = KarhunenLoeve(name='bsplines', n_functions=5)
kl.new(n_obs=10, argvals=np.linspace(0, 1, 101),
       cluster_std='exponential')
```

Figure 5.4 presents a plot of the simulated Brownian motions and those from the Karhunen-Loève decomposition. The simulation of two dimensional data is based on the tensor product of basis functions. Simulation for higher dimensional data is not implemented.

We also added methods to generate noisy observations as well as sparse data. Note that, these functions are only implemented on instances of `Simulation`. The `add_noise` function adds pointwise noise to the observations. Both homoscedastic and heteroscedastic noise are implemented. If a single scalar is given as a parameter to the function, homoscedastic noise will be simulated. For the heteroscedastic case, lambda functions and vector of size `n_points` can be supplied by the user. The noisy data are stored in the



(a) (b)

Figure 5.5: Results for the `add_noise` and `sparsify` functions on the Karhunen-Loève simulated data. (a) Noisy data. (b) Sparse data.

instance variable `noisy_data`. For example, to add random noise with variance $\sigma^2 = 0.05$, we run

```
kl.add_noise(var_noise=0.05)
```

and, for heteroscedastic noise with variance defined by $x \to \sqrt{1 + |x|}$,

```
kl.add_noise(var_noise=lambda x: np.sqrt(1 + np.abs(x)))
```

The function `sparsify` randomly removes sampling points from the observation. Precisely, we randomly generate the number of sampling points to retain for each observation and then randomly select the sampling points to remove from each observation. The sparse data are stored in the instance variable `sparse_data`. For example, to randomly remove 50% of the sampling points (more or less 5%) on the Brownian simulated data, we run

```
brownian.sparsify(percentage=0.5, epsilon=0.05)
```

Figure 5.5 presents a plot of the noisy and sparse verions of the Karhunen-Loève simulated data.

Clusters simulation

Let K be a positive integer, and let Z be a discrete random variable taking values in the range $\{1, \dots, K\}$ such that

$$\mathbb{P}(Z = k) = p_k \quad \text{with} \quad p_k > 0 \quad \text{and} \quad \sum_{k=1}^{K} p_k = 1.$$

The variable Z represents the cluster membership of the realizations of the process. We consider that the stochastic process follows a functional mixture model with K components, that is, it allows for the following decomposition:

$$X(t) = \sum_{k=1}^{K} \mu_k(t) \mathbf{1}_{\{Z=k\}} + \sum_{j \geq 1} \xi_j \phi_j(t), \quad t \in \mathcal{T},$$

where

- $\mu_1, \dots, \mu_K$ are the mean curves per cluster.
- $\{\phi_j\}_{j \geq 1}$ is an orthonormal basis of functions.
- $\xi_j, j \geq 1$ are real-valued random variables which are conditionally independent given Z. For each $1 \leq k \leq K$, $\xi_j | Z = k \sim \mathcal{N}(0, \sigma_{kj}^2)$.

For example, we can generate $N = 10$ realizations of two clusters using 3 eigenfunctions with given coefficients with

```
N = 10
n_features = 3
n_clusters = 2
centers = np.array([[2, -1], [-0.5, 1.5], [0, 0]])
cluster_std = np.array([[2, 1], [0.5, 1], [1, 1]])

simu = KarhunenLoeve('wiener', n_functions=n_features)
simu.new(n_obs=N, n_clusters=n_clusters,
         centers=centers, cluster_std=cluster_std)
```

Figure 5.6 shows the plot of the simulated data corresponding to the previous code snippet.

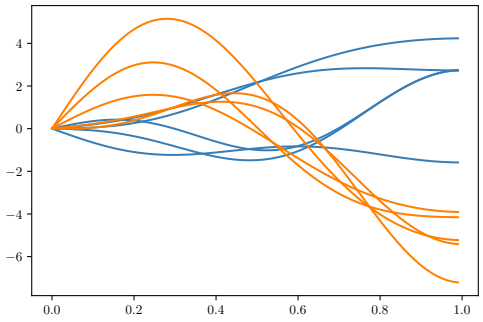


Figure 5.6: Simulation of data with two clusters. Each color represents a cluster.

5.5 Parameters estimation

5.5.1 Curves denoising

Considering the model defined in (5.1), we assume that $P = 1$ and for the sake of readability, we omit the superscript. The objective is to estimate the function $X_n(\cdot)$ using the available sample points. Thus, we consider local polynomial smoothers [45]. This type of estimators crucially depends on a tuning parameter, the bandwidth.

Let $\mathbf{d} \geq 0$ be an integer and $t_0 \in I$ be the evaluation points for the estimation of $X^{(n)}$. For any $u \in \mathbb{R}$, we consider the vector $U(u) = (1, u, \ldots, u^{\mathbf{d}}/\mathbf{d}!)$ and note that $U_h(\cdot) = U(\cdot/h)$. Let $K : \mathbb{R} \to \mathbb{R}$ be a positive kernel and define $K_h(\cdot) = h^{-1}K(\cdot/h)$. Moreover, we define:

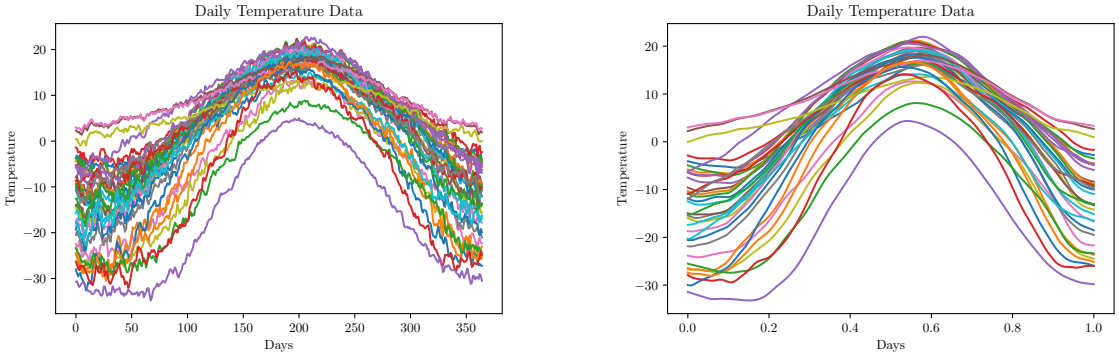
$$\vartheta_{M_n,h} \coloneqq \underset{\vartheta \in \mathbb{R}^{\mathbf{d}+1}}{\arg\min} \sum_{m=1}^{M_n} \left\{ Y_m^{(n)} - \vartheta^\top U_h \left(T_m^{(n)} - t_0 \right) \right\}^2 K_h \left(T_m^{(n)} - t_0 \right),$$

where h is the bandwidth. The vector $\vartheta_{M_n,h}$ satisfies the normal equations $A\vartheta_{M_n,h} = a$ with

$$A = A_{M_n,h} = \frac{1}{M_n} \sum_{m=1}^{M_n} U_h \left(T_m^{(n)} - t_0 \right) U_h^\top \left(T_m^{(n)} - t_0 \right) K_h \left(T_m^{(n)} - t_0 \right), \tag{5.2}$$

$$a = a_{M_n,h} = \frac{1}{M_n} \sum_{m=1}^{M_n} Y_m^{(n)} U_h \left(T_m^{(n)} - t_0 \right) K_h \left(T_m^{(n)} - t_0 \right). \tag{5.3}$$

Let λ be the smallest eigenvalue of the matrix A and note that, whenever $\lambda > 0$, we have $\vartheta_{M_n,h} = A^{-1}a$. With at hand an estimation of the bandwidth $\widehat{h}$, the local polynomial



(a) (b)

Figure 5.7: (a) Curve and (b) smoothed estimation for the Canadian Temperature data.

estimator of $\widehat{X}^{(n)}(t_0)$ of order $\mathbf{d}$ is given by:

$$\widehat{X}^{(n)}(t_0) = U^\top(0)\widehat{\vartheta}, \quad \text{where} \quad \widehat{\vartheta} = \vartheta_{M_n,\widehat{h}}.$$

If $\mathbf{d} = 0$, we are in the particular case of the Nadaraya-Watson estimator. The Gaussian, Epanechnikov, tri-cube and bi-square kernels are implemented and others can be added in a modular way. We propose an estimate of the bandwidth h that is based on the regularity of the underlying function [59].

For example, if we want to smooth the daily temperature curves using a local polynomial smoother with an estimate of bandwidth at $t_0 = 0.5$ and a neighborhood of 2 points, we run:

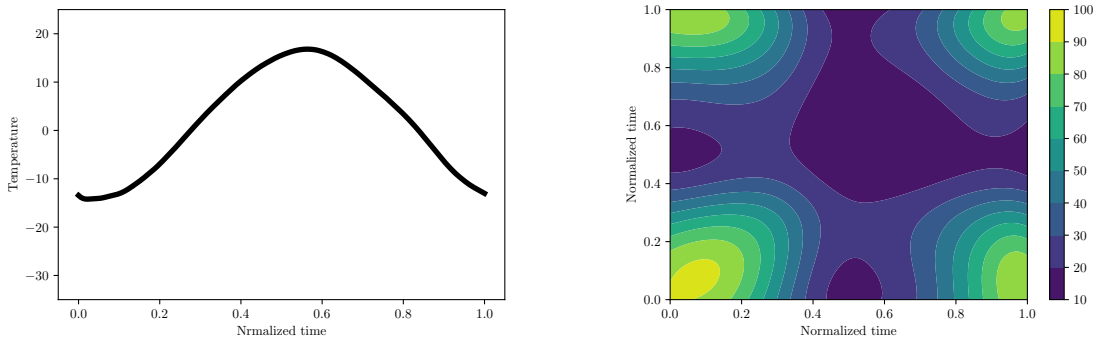
```
dailyTemp_smooth = dailyTemp.smooth(points=0.5, neighborhood=2)
```

Figure 5.7 presents the plot of the smoothed temperature data compared to the original ones.

5.5.2 Mean and covariance estimation

In this section, we develop estimators for the mean and the covariance functions of a component $X_p, 1 \leq p \leq P$ from the process X. These estimators might be used to compute estimators of eigenvalues and eigenfunctions of X_p for the Karhunen-Loève expansion.

Let $\widehat{X}_p^{(n)}$ be a suitable nonparametric estimator of the curve $X_p^{(n)}$ applied with the $M_{n,p}$ pairs $(Y_{m_p}^{(n)}, T_{m_p}^{(n)}), n = 1, \ldots, N$, as for instance a local polynomial estimator such



(a) (b)

Figure 5.8: (a) Mean and (b) covariance estimation for the Canadian Temperature data.

as that presented in the previous subsection. With at hand the $\widehat{X}_n$'s tuned for the mean function estimation, we define

$$\widehat{\mu}_{p,N}(t_p) = \frac{1}{N}\sum_{n=1}^{N} \widehat{X}_p^{(n)}(t_p), \quad t_p \in I_p.$$

For example, the code snippet for the estimation of the mean curve of the daily temperature curves using local linear smoother with bandwidth equal to 0.05 is

```
mean_temp = dailyTemp.mean(smooth='LocalLinear', bandwidth=0.05)
```

For the covariance function, following [146], we distinguish the diagonal from the non-diagonal points. With at hand the $\widehat{X}_p^{(n)}$'s tuned for the covariance function estimation,

$$\widehat{C}_{p,p}(s_p, t_p) = \frac{1}{N}\sum_{n=1}^{N} \widehat{X}_p^{(n)}(s_p)\widehat{X}_p^{(n)}(t_p) - \widehat{\mu}_{p,N}(s_p)\widehat{\mu}_{p,N}(t_p), \quad s_p, t_p \in I_p, \quad s_p \neq t_p. \tag{5.4}$$

The diagonal of the covariance is then estimated using two-dimensional kernel smoothing with $\widehat{C}_{p,p}(s_p, t_p), s_p \neq t_p$ as input data. See [146] for the details.

```
cov_temp = dailyTemp.covariance(smooth='GAM')
```

5.6 MFPCA

The `FDApy` package implements MFPCA for data defined on potentially different domains, developped by Happ and Greven [64]. The implementation of the method is build upon the functional data classes defined in the package. After giving a short review of the methodology in Section 5.6.1, we explain how to effectively use it in Section 5.6.2. For theoretical details, please refer to [64].

5.6.1 Methodological background

Following Happ and Greven [64], the multivariate components for X are computed by plugging in the univariate components computed from each component X_p. These estimations are done as the follows.

1. Perform a univariate fPCA on each of the components of X separately. For a component X_p, the eigenfunctions and eigenvectors are computed as a matrix analysis of the estimated covariance $\widehat{C}_{p,p}$. This results in a set of eigenfunctions $\left(\widehat{\rho}_{p,1}, \dots, \widehat{\rho}_{p,J_p}\right)$ associated with a set of eigenvalues $\left(\widehat{\lambda}_{p,1}, \dots, \widehat{\lambda}_{p,J_p}\right)$ for a given truncation integer J_p. The univariate scores for a realization $X_p^{(n)}$ of X_p are then given by $\widehat{\mathbf{c}}_{p,j}^{(n)} = \langle \widehat{X}_p^{(n)}, \widehat{\rho}_{p,j} \rangle,\ 1 \leq j \leq J_p$.

2. Define the matrix $\mathcal{Z} \in \mathbb{R}^{N \times J_+}, J_+ = \sum_{p=1}^{P} J_p$, where each row stacks the scores for each components for a unique observation $\left(\widehat{\mathbf{c}}_{1,1}^{(n)}, \dots, \widehat{\mathbf{c}}_{1,J_1}^{(n)}, \dots, \widehat{\mathbf{c}}_{P,1}^{(n)}, \dots, \widehat{\mathbf{c}}_{P,J_P}^{(n)}\right)$. Define $\mathbf{Z} \in \mathbb{R}^{J_+ \times J_+}$ such that $\mathbf{Z} = (N-1)^{-1} \mathcal{Z}^\top \mathcal{Z}$.

3. An eigenanalysis of the matrix $\mathbf{Z}$ is performed and leads to the eigenvectors $\widehat{\boldsymbol{v}}_j$ and eigenvalues $\widehat{\lambda}_j$.

4. Finally, the multivariate eigenfunctions are estimated with

$$\widehat{\varphi}_{p,j}(t_p) = \sum_{j'=1}^{J_p} [\widehat{\boldsymbol{v}}_j]_{p,j'} \widehat{\rho}_{p,j'}(t_p), \quad t_p \in I_p,\ 1 \leq j \leq J_+,\ 1 \leq p \leq P.$$

   and the multivariate scores with

$$\widehat{\mathbf{c}}_j^{(n)} = \mathcal{Z}_{n,\cdot} \widehat{\boldsymbol{v}}_j, \quad 1 \leq n \leq N_0, \quad 1 \leq j \leq J_+.$$

The multivariate Karhunen-Loève expansion of the process X is thus

$$\widehat{X}^{(n)}(\boldsymbol{t}) = \widehat{\mu}_N(\boldsymbol{t}) + \sum_{j=1}^{J_+} \widehat{\mathfrak{c}}_j^{(n)} \widehat{\varphi}_j(\boldsymbol{t}), \quad \boldsymbol{t} \in \boldsymbol{I}.$$

where $\widehat{\mu}_N(\cdot) = (\widehat{\mu}_{1,N}(\cdot), \ldots, \widehat{\mu}_{P,N}(\cdot))$ is the vector of the estimated mean functions.

5.6.2 Implementation

The implementation of the MFPCA is based on the `MFPCA` class. Hence, we construct an object of class `MFPCA` specifying the number of eigencomponents that we want. The computation of the eigenelements is performed using the `fit` method, and the scores are then calculted using the `transform` method. Given scores, the inverse transformation to the functional space is done using the `inverse_transform` method. The triptych `fit`, `transform` and `fit_transform` is based on the implementation choice of the package `sklearn`.

MFPCA for the Canadian Weather data

In this example, we perform an MFPCA for the bivariate Canadian Weather data. We expand each univariate element using an univariate fPCA with a number of components that explain 99% of the variance within the data. The number of components are specified in a list in the `MFPCA` constructor. The `method` parameter in the `fit` method indicates how the univariate scores are computed. Here, we use numerical integration to derive them.

```
fpca = MFPCA(n_components=[0.99, 0.99])
fpca.fit(canadWeather, method='NumInt')
```

The scores are computed using the `transform` function:

```
scores = fpca.transform(data=canadWeather)
```

The eigenvalues are stored as instance variables. We remark the rapid decrease of the eigenvalues. Hence, we only need a few eigencomponents to explain most of the variance within the data.

```
fpca.eigenvalues
```

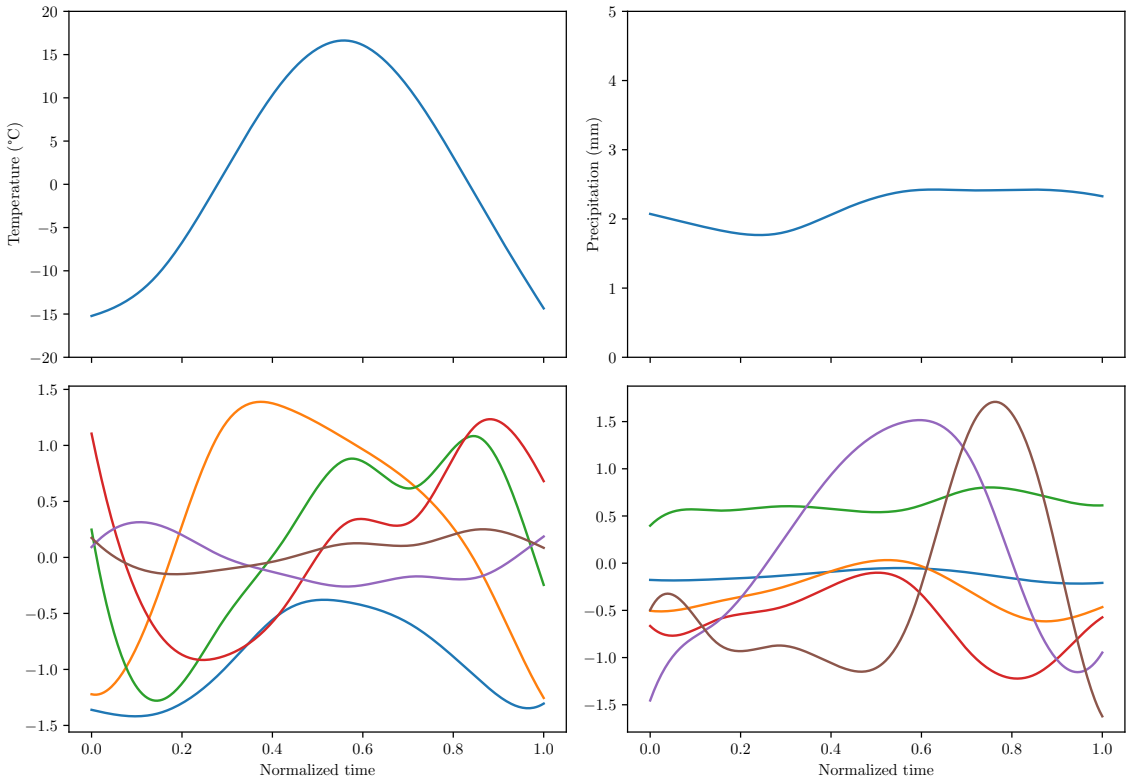


Figure 5.9: Results of the MFPCA for the Canadian Weather data. The first row represents the mean functions of the temperature (left) and precipitation (right) data. The second row corresponds to the bivariate eigenfunctions found for 99% of explained variance.

```
array([4.36e+01, 4.62e+00, 1.20e+00, 5.76e-01, 1.14e-01, 2.61e-02])
```

Implemented univariate basis expansion

`MFPCA` is based on the univariate basis expansion of each of the components of the process. Currently, only two basis expansions are implemented. New bases can easily be added to the package. All univariate basis expansions should implemented the methods: `fit`, used to compute the elements of the basis, `transform`, to compute the scores of the observations within the basis, and `fit_transform`, to return the observations in the functional space given their scores. The implemented bases are:

- `UFPCA` – Univariate Functional Principal Components Analysis for data on one dimensional domains. This basis was used in the Canadian Weather example. Multiple smoothing methods are implemented for the estimation of the mean and the covariance (see Section 5.5), such as local polynomial estimation or Generalized Additive Models (GAM) with penalized B-splines. The scores are computed using numerical integration. Considering sparse functional data, one may also used the PACE algorithm [146]. The main argument to build an instance of the class `UFPCA` is `n_comp` which can be the proportion of variance explained by the principal components, if `n_comp` < 1, or the number of principal components to computed, if `n_comp` ≥ 1.

- `FCP-TPA` – FCP-TPA for data on two dimensional domains. This algorithm is used to find a basis decomposition of image data. Consider N realizations of a stochastic process X defined on $S_x \times S_y$, the data can be represented as a tensor $\mathbf{X}$ in $\mathbb{R}^{N \times S_x \times S_y}$. A Candecomp/Parafac representation of the data is assumed:

$$\mathbf{X} = \sum_{j=1}^{J} \lambda_j u_j \otimes v_j \otimes w_j,$$

  where λ_j is scalar, $u_j \in \mathbb{R}^N, v_j \in \mathbb{R}^{S_x}$ and $w_j \in \mathbb{R}^{S_y}$ are vectors and $\otimes$ denotes the outer product. In addition, the outer product $v_j \otimes w_j$ can be interpreted as the jth eigenimage evaluated on the same grid points as the original data. Moreover, the vector $\lambda_j v_j$ is the score vector gathering the observations projected onto the eigenimage $v_j \otimes w_j$. Our implementation is adapted from the function `FCP_TPA` of the **R** package `MFPCA` [67]. The main argument to build an instance of the class `FCPTPA`, is `n_comp` which is the number of principal components to computed.

5.7 `fCUBT`

The `FDApy` package implements the fCUBT for clustering of functional data objects defined on potentially different domains, developed by [58]. The implementation of the method is build upon the functional data classes defined in the package. After giving a short review of the methodology in Section 5.7.1, we explain how to effectively use it in Section 5.7.2. For a detailed description, please refer to [58].

5.7.1 Methodological background

Let $\mathcal{S}$ be a sample of realizations of the process X. We consider the problem of learning a partition $\mathcal{U}$ such that every element U of $\mathcal{U}$ gathers similar elements of $\mathcal{S}$. The partition $\mathcal{U}$ is built as a tree $\mathfrak{T}$ defined using a top-down procedure by recursive splitting. Each node of the tree $\mathfrak{T}$ is denoted by $\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}$.

Growing

At each stage, a node $(\mathfrak{d}, \mathfrak{j})$ is possibly split into two subnodes in a four step procedure:

1. A MFPCA, with $\mathtt{n_{comp}}$ components, is conducted on the elements of $\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}$. It results in a set of eigenvalues $\Lambda_{\mathfrak{d},\mathfrak{j}}$ associated with a set of eigenfunctions $\Phi_{\mathfrak{d},\mathfrak{j}}$.

2. The matrix of scores $C_{\mathfrak{d},\mathfrak{j}}$ is then defined with the columns built with the projections of the elements of $\mathcal{S}_{\mathfrak{d},\mathfrak{j}}$ onto the elements of $\Phi_{\mathfrak{d},\mathfrak{j}}$.

3. For each $K = 1, \ldots, K_{max}$, we fit a GMM to the columns of the matrix $C_{\mathfrak{d},\mathfrak{j}}$. The resulting models are denoted as $\{\mathcal{M}_1, \ldots, \mathcal{M}_{K_{max}}\}$. Considering the BIC, we determine
$$\widehat{K}_{\mathfrak{d},\mathfrak{j}} = \underset{K=1,\ldots,K_{max}}{\arg\max} \ \mathrm{BIC}(\mathcal{M}_K) \tag{5.5}$$

4. If $\widehat{K}_{\mathfrak{d},\mathfrak{j}} > 1$, we split $\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}$ using the model $\mathcal{M}_2$, which is a mixture of two Gaussian vectors. Otherwise, the node is considered to be a terminal node and the construction of the tree is stopped for this node.

The recursive procedure continues downwards until one of the following stopping rules are satisfied: there are less than `minsize` observations in the node or the estimation $\widehat{K}_{\mathfrak{d},\mathfrak{j}}$ of the number of clusters in the mode is equal to 1. When the algorithm ends, a label

is assigned to each leaf (terminal node). The resulting tree is referred to as the maximal binary tree.

Joining

In this step, the idea is to join terminal nodes which do not necessarily share the same direct ancestor.

1. Build the graph $\mathcal{G} = (V, E)$ where

$$V = \{\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}, 0 \leq j < 2^{\mathfrak{d}}, 0 \leq \mathfrak{d} < \mathfrak{D} \mid \mathfrak{S}_{\mathfrak{d},\mathfrak{j}} \text{ is a terminal node}\}, \quad \text{and}$$

$$E = \left\{(\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}, \mathfrak{S}_{\mathfrak{d}',\mathfrak{j}'}) \mid \mathfrak{S}_{\mathfrak{d},\mathfrak{j}}, \mathfrak{S}_{\mathfrak{d}',\mathfrak{j}'} \in V,\ \mathfrak{S}_{\mathfrak{d},\mathfrak{j}} \neq \mathfrak{S}_{\mathfrak{d}',\mathfrak{j}'} \text{ and } \widehat{K}_{(\mathfrak{d},\mathfrak{j})\cup(\mathfrak{d}',\mathfrak{j}')} = 1\right\}. \tag{5.6}$$

2. Let $(\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}, \mathfrak{S}_{\mathfrak{d}',\mathfrak{j}'})$ be the edge with the maximum BIC value. Remove this edge then and replace the asssociated vertex by $\mathfrak{S}_{\mathfrak{d},\mathfrak{j}} \cup \mathfrak{S}_{\mathfrak{d}',\mathfrak{j}'}$.

3. Continue the procedure by applying the step **1.** with $\{V \setminus \{\mathfrak{S}_{\mathfrak{d},\mathfrak{j}}, \mathfrak{S}_{\mathfrak{d}',\mathfrak{j}'}\}\} \cup \{\mathfrak{S}_{\mathfrak{d},\mathfrak{j}} \cup \mathfrak{S}_{\mathfrak{d}',\mathfrak{j}'}\}$.

The procedure continues until the set V is reduced to a unique element or the set E is the empty set.

5.7.2 Implementation

The implementation of the fCUBT is based on the `FCUBT` class. Hence, we construct an object of `FCUBT` class specifyng the root node of the tree which contains a sample of data. The growth of the tree is performed using the `grow` function with the number of eigencomponents to keep at each node as parameters. Once the tree has grown, the joining step is made using the `join` function. The prediction of the class of a new observation is possible through the `predict` function (or `predict_proba` for the probabilities to belong to each class).

Example on the Canadian Weather data

In this example, we perform a clustering of the univariate Canadian Temperature data extracted from the bivariate Canadian Weather data. We build the root node containing all the observations within the dataset. The `FCUBT` constructor is called.

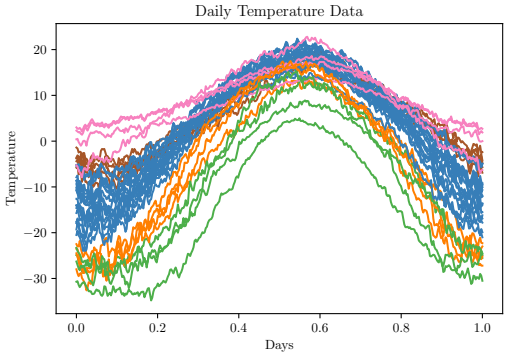


Figure 5.10: Plot of the Canadian Temperature dataset. Each color represents a different cluster.

```
root_node = Node(dailyTemp, is_root=True)
fcubt = FCUBT(root_node=root_node)
```

To grow the tree, we choose to consider a number of components that explain 95% of the variance of the remaining observations at each node of the tree. Moreover, the construction of the branch is stopped if there are less than 5 observations in a node. Figure 5.10 presents the results of clustering. The `plot` function from the `FCUBT` class allows us to show the maximum tree once the data has been fitted (currently, only for univariate data objects). This representation is particularly useful for the understanding of the clustering results. One might also cut the tree at a given height. For example, considering Figure 5.11,

```
fcubt.grow(n_components=0.95, min_size=5)
```

The joining step is performed using the `join` function. We choose to consider 95% of the explained variance of the observations to join two nodes.

```
fcubt.join(n_components=0.95)
```

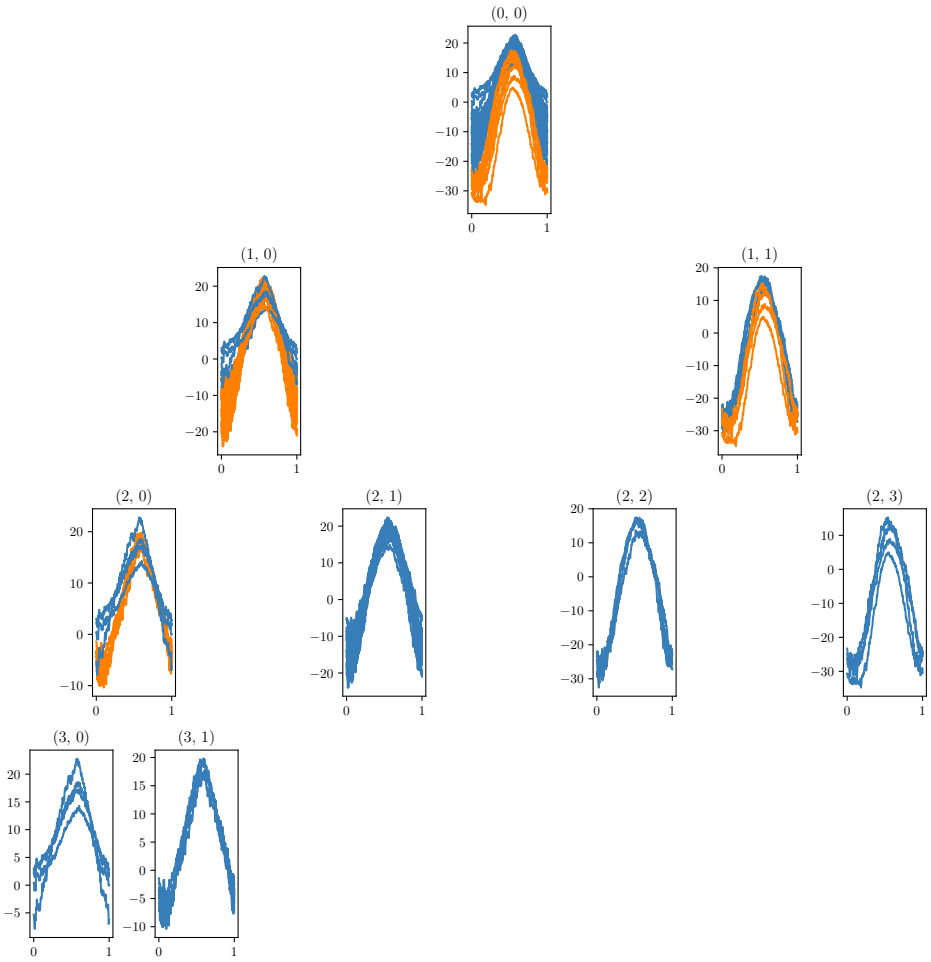


Figure 5.11: Grown tree $\mathfrak{T}$ illustration for the Canadian Temperature dataset.

5.8 Conclusion

The package is publicly available on Github[4] and the Python Package Index[5]. A documentation, including examples, is available with the package. Some tests are also implemented using `unittest` which is the unit testing framework provided with **Python**.

4. https://github.com/StevenGolovkine/FDApy
5. https://pypi.org/project/FDApy/

Conclusion

Being able to analyze data with different structures and dimensions becomes critical today, and especially, in the automotive industry where more and more data are recorded from embarked sensors.

In the second chapter of the thesis, an estimate of the local regularity of the process generating our data has been introduced as a guideline for our statistical methodology contributions. This new point of view allows us to have less stringent assumptions than the standard ones in the functional data analysis framework. Standard assumptions usually include continuously observed errorless smooth curves (by smooth, we mean at least twice differentiable). So, in particular, we can relax the twice differentiable assumption in our approach.

Based on a thorough theoretical foundation, the proposed estimator for the local regularity of the process was shown to concentrate to its true value under general mild assumptions without imposing a specific distribution to the stochastic process. Morover, this theoretical result is non-asymptotic in the sense that it hold true for a number of realizations of the process and sampling points satisfyting our assumptions. The results of the simulation study show the accuracy of the estimator in a large number of scenarios. The estimate is easily computable and updatable if new data arrives.

From a methodological point of view, the estimate of the local regularity should be used to derive estimators of different quantities from the data. Thus, with at hand this estimate, we build a nearly optimal local polynomial smoother for noisy trajectories and nonparametric estimators for their mean and covariance functions. Future research might aim at deriving estimators for other quantities of interest, such as the eigencomponents and the scores. It could also be interesting to use it to estimate coefficient functions in the functional linear model for example.

Given an estimation of the local regularity of the process computed on some realizations, bounds for the pointwise risk of the local polynomial smoother for noisy trajectories are derived uniformly on a new set of realizations of the process. Simulation results indicate good performance to recover the true curves in a variety of case, as well as fast computation, especially compare to the cross-validation method. Concerning the mean

and covariance functions, the estimators have nearly optimal rates of convergence in the minimax sense.

The results of the automotive applications in Chapter 2 and Chapter 3 indicate that the use of the local regularity can lead to accurate estimation of the curves, as well as the mean and covariance functions. The method is general and applications are of course not restricted to the automotive industry. For example, one may cite electric power consumption data for which the local regularity is likely to be small.

Clustering is a statistical model class that aims at finding meaninful partitions of a sample of data. The `fCUBT` procedure has been proposed as a model-based clustering algorithm for a general class of functional data for which the components could be curves or images. The idea is to build a set of binary trees by recursive splitting of the observations. The number of groups are determined in a data-driven way. It provides easily interpretable and fast prediction for online data sets. The results of the analysis of vehicle trajectories on a German roundabout show the ability of the algorithm to find driving scenarios that are representatives of the driver behaviors in a specific situation (a roundabout). The current implementation of the algorithm in the `FDApy` package is very flexible and new methods can easily be added. However, there is still room for improvements and extensions. For example, up to now, the algorithm is not robust to outliers. We could imaging replace the Gaussian Mixture Model by some robust clustering algorithm (see *e.g.* [51]). Furthermore, our model does not include clusters that are defined by phase variation. Future research might also integrate warping functions in the model to consider such cases (see [102]).

In the thesis, we focused our applications on trajectories data. However, our contributions are very general and are not limited to the Engineering department in the automotive industry. Functional data analysis can also be applied to driving laws used by the Quality department to cite an example. The last contribution was to make known and to show the usefulness of functional data analysis within the company.

Bibliography

[1] Allen, G. I. Multi-way functional principal components analysis. In *2013 5th IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 220–223, Dec. 2013.

[2] Belloni, A., Chernozhukov, V., Chetverikov, D., and Kato, K. Some new asymptotic theory for least squares series: Pointwise and uniform results. *Journal of Econometrics*, 186(2):345–366, June 2015.

[3] Ben Slimen, Y., Allio, S., and Jacques, J. Model-based co-clustering for functional data. *Neurocomputing*, 291:97–108, May 2018.

[4] Berrendero, J., Justel, A., and Svarc, M. Principal components for multivariate functional data. *Computational Statistics & Data Analysis*, 55:2619–2634, Sept. 2011.

[5] Bhattacharyya, A., Hanselmann, M., Fritz, M., Schiele, B., and Straehle, C.-N. Conditional Flow Variational Autoencoders for Structured Sequence Prediction. *arXiv:1908.09008 [cs, stat]*, Aug. 2020.

[6] Biernacki, C., Celeux, G., and Govaert, G. 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.

[7] Blanke, D. and Vial, C. Global smoothness estimation of a Gaussian process from general sequence designs. *Electronic Journal of Statistics*, 8(1):1152–1187, 2014.

[8] Bock, J., Krajewski, R., Moers, T., Runde, S., Vater, L., and Eckstein, L. The inD Dataset: A Drone Dataset of Naturalistic Road User Trajectories at German Intersections. *arXiv:1911.07602 [cs, eess]*, Nov. 2019.

[9] Bouveyron, C. *funFEM: Clustering in the Discriminative Functional Subspace*, 2015. R package version 1.1.

[10] Bouveyron, C. and Jacques, J. Model-based Clustering of Time Series in Group-specific Functional Subspaces. *Advances in Data Analysis and Classification*, 5(4):281–300, 2011.

[11] Bouveyron, C., Côme, E., and Jacques, J. The discriminative functional mixture model for a comparative analysis of bike sharing systems. *The Annals of Applied Statistics*, 9(4):1726–1760, Dec. 2015.

[12] Bouveyron, C., Bozzi, L., Jacques, J., and Jollois, F.-X. 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, 2018.

[13] Bouveyron, C., Celeux, G., Murphy, T. B., and Raftery, A. E. *Model-Based Clustering and Classification for Data Science: With Applications in R*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2019.

[14] Bouveyron, C., Jacques, J., and Schmutz, A. *funLBM: Model-Based Co-Clustering of Functional Data*, 2020. R package version 2.1.

[15] Brockhaus, S., Rügamer, D., and Greven, S. Boosting functional regression models with FDboost. *Journal of Statistical Software*, 94(10):1–50, 2020.

[16] Butterworth, S. On the theory of filter amplifiers. *Wireless Engineer*, 7(6): 536–541, 1930.

[17] Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., and Beijbom, O. nuScenes: A multimodal dataset for autonomous driving. *arXiv:1903.11027 [cs, stat]*, May 2020.

[18] Cai, T. T. and Yuan, M. Nonparametric Covariance Function Estimation for Functional and Longitudinal Data. *University of Pennsylvania and Georgia institute of technology*, page 36, 2010.

[19] Cai, T. T. and Yuan, M. Optimal estimation of the mean function based on discretely sampled functional data: Phase transition. *Annals of Statistics*, 39(5): 2330–2355, Oct. 2011.

[20] Cai, T. T. and Yuan, M. Minimax and Adaptive Estimation of Covariance Operator for Random Variables Observed on a Lattice Graph. *Journal of the American Statistical Association*, 111(513):253–265, Jan. 2016.

[21] Caliński, T. and Harabasz, J. A Dendrite Method for Cluster Analysis. *Communications in Statistics - Theory and Methods*, 3:1–27, Jan. 1974.

[22] Carreño, C. R. Gaa-uam/scikit-fda: Version 0.4, July 2020.

[23] Carroll, C., Müller, H.-G., and Kneip, A. Cross-component registration for multivariate functional data, with application to growth curves. *Biometrics*, July 2020.

[24] Carroll, C., Gajardo, A., Chen, Y., Dai, X., Fan, J., Hadjipantelis, P. Z., Han, K., Ji, H., Mueller, H.-G., and Wang, J.-L. *fdapace: Functional Data Analysis and Empirical Dynamics*, 2021. R package version 0.5.6.

[25] Carroll, R. J., Delaigle, A., and Hall, P. Unexpected properties of bandwidth choice when smoothing discrete data for constructing a functional data classifier. *Annals of Statistics*, 41(6), Dec. 2013.

[26] Cattell, R. B. The Scree Test For The Number Of Factors. *Multivariate Behavioral Research*, 1(2):245–276, Apr. 1966.

[27] Celeux, G. and Govaert, G. A classification EM algorithm for clustering and two stochastic versions. report, INRIA, 1991.

[28] Celeux, G., Chauveau, D., and Diebolt, J. On Stochastic Versions of the EM Algorithm. report, INRIA, 1995.

[29] Chamroukhi, F. and Nguyen, H. D. Model-based clustering and classification of functional data. *WIREs Data Mining and Knowledge Discovery*, 9(4):e1298, 2019.

[30] Chan, G. and Wood, A. T. A. Estimation of fractal dimension for a class of non-Gaussian stationary processes and fields. *Annals of Statistics*, 32(3):1222–1260, June 2004.

[31] Cheng, S.-S., Wang, H.-M., and Fu, H.-C. A model-selection-based self-splitting Gaussian mixture learning with application to speaker identification. *EURASIP Journal on Advances in Signal Processing*, 2004:2626–2639, Jan. 2004.

[32] Cherfi, A., Arbaretier, E., and Zhao, L. Sécurité-innocuité des véhicules autonomes : enjeux et verrous. In *Congrès Lambda Mu 20 de Maîtrise des Risques et de Sûreté de Fonctionnement*. IMdR, Dec. 2016.

[33] Constantine, A. G. and Hall, P. Characterizing Surface Smoothness via Estimation of Effective Fractal Dimension. *Journal of the Royal Statistical Society. Series B (Methodological)*, 56(1):97–113, 1994.

[34] Cox, D. D. and Savoy, R. L. Functional magnetic resonance imaging (fMRI) "brain reading": detecting and classifying distributed patterns of fMRI activity in human visual cortex. *NeuroImage*, 19(2 Pt 1):261–270, June 2003.

[35] Dai, W. and Genton, M. Multivariate Functional Data Visualization and Outlier Detection. *Journal of Computational and Graphical Statistics*, Mar. 2017.

[36] Davies, D. and Bouldin, D. A Cluster Separation Measure. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-1:224–227, May 1979.

[37] Delaigle, A. and Hall, P. Defining probability density for a distribution of random functions. *The Annals of Statistics*, 38(2):1171–1193, Apr. 2010.

[38] Delaigle, A. and Hall, P. Achieving near perfect classification for functional data. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 74(2):267–286, 2012.

[39] Delaigle, A. and Hall, P. Classification using censored functional data. *Journal of the American Statistical Association*, 108(504):1269–1283, 2013.

[40] Dempster, A. P., Laird, N. M., and Rubin, D. B. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[41] Di, C.-Z., Crainiceanu, C. M., Caffo, B. S., and Punjabi, N. M. Multilevel functional principal component analysis. *The Annals of Applied Statistics*, 3(1): 458–488, Mar. 2009.

[42] Diehl, F., Brunner, T., Le, M. T., and Knoll, A. Graph Neural Networks for Modelling Traffic Participant Interaction. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 695–701, June 2019.

[43] Dong, C., Dolan, J. M., and Litkouhi, B. Intention estimation for ramp merging control in autonomous driving. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pages 1584–1589, June 2017.

[44] European Commission. Roadmap to a Single European Transport Area – Towards a competitive and resource efficient transport system, 2011. COM(2011) - 144 final.

[45] Fan, J. and Gijbels, I. *Local polynomial modelling and its applications*. Number 66 in Monographs on statistics and applied probability , ISSN 0960-6696 ; ZDB-ID: 22968-4. Chapman & Hall, London, 1996.

[46] Ferraty, F. and Vieu, P. *Nonparametric Functional Data Analysis: Theory and Practice*. Springer Series in Statistics. Springer-Verlag, New York, 2006.

[47] FHWA, U.S. Department of Transportation. NGSIM–Next Generation SIMulation, 2006.

[48] Fraiman, R., Ghattas, B., and Svarc, M. Interpretable clustering using unsupervised binary trees. *Advances in Data Analysis and Classification*, 7(2), June 2013.

[49] Fraley, C. and Raftery, A. E. Model-Based Clustering, Discriminant Analysis, and Density Estimation. *Journal of the American Statistical Association*, 97(458): 611–631, June 2002.

[50] Gaïffas, S. On pointwise adaptive curve estimation based on inhomogeneous data. *ESAIM: Probability and Statistics*, 11:344–364, 2007.

[51] García-Escudero, L., Gordaliza, A., Matrán, C., and Mayo, A. A review of robust clustering methods. *Advances in Data Analysis and Classification*, 4: 89–109, Sept. 2010.

[52] Geiger, A., Lenz, P., Stiller, C., and Urtasun, R. Vision meets robotics: the KITTI dataset. *The International Journal of Robotics Research*, 32:1231–1237, Sept. 2013.

[53] Ghattas, B., Michel, P., and Boyer, L. Clustering nominal data using unsupervised binary decision trees: Comparisons with the state of the art methods. *Pattern Recognition*, 67:177–185, July 2017.

[54] Gneiting, T., Ševčíková, H., and Percival, D. B. Estimators of Fractal Dimension: Assessing the Roughness of Time Series and Spatial Data. *Statistical Science*, 27(2):247–277, May 2012.

[55] Goldenshluger, A. and Lepski, O. Bandwidth selection in kernel density estimation: Oracle inequalities and adaptive minimax optimality. *The Annals of Statistics*, 39(3):1608–1632, June 2011.

[56] Goldsmith, J., Greven, S., and Crainiceanu, C. Corrected Confidence Bands for Functional Data Using Principal Components. *Biometrics*, 69(1):41–51, 2013.

[57] Goldsmith, J., Scheipl, F., Huang, L., Wrobel, J., Di, C., Gellar, J., Harezlak, J., McLean, M. W., Swihart, B., Xiao, L., Crainiceanu, C., and Reiss, P. T. *refund: Regression with Functional Data*, 2020. R package version 0.1-22.

[58] Golovkine, S., Klutchnikoff, N., and Patilea, V. Clustering multivariate functional data using unsupervised binary trees. *arXiv:2012.05973 [cs, stat]*, Dec. 2020.

[59] Golovkine, S., Klutchnikoff, N., and Patilea, V. Learning the smoothness of noisy curves with application to online curve estimation. *arXiv:2009.03652 [math, stat]*, Sept. 2020.

[60] Govaert, G. and Nadif, M. *Co-Clustering: Models, Algorithms and Applications*. John Wiley & Sons, Dec. 2013.

[61] Guillou, A. and Klutchnikoff, N. Minimax pointwise estimation of an anisotropic regression function with unknown density of the design. *Math. Methods Statist.*, 20(1):30–57, 2011.

[62] Halkias, J. and Colyar, J. NGSIM interstate 80 freeway dataset. Technical report, US Federal Highway Administration, Washington, DC, USA, 2006.

[63] Hall, P., Müller, H.-G., and Wang, J.-L. Properties of principal component methods for functional and longitudinal data analysis. *The Annals of Statistics*, 34 (3):1493–1517, June 2006.

[64] Happ, C. and Greven, S. Multivariate Functional Principal Component Analysis for Data Observed on Different (Dimensional) Domains. *Journal of the American Statistical Association*, 113(522):649–659, Apr. 2018.

[65] Happ, C. M. *Statistical methods for data with different dimensions*. Text.PhDThesis, Ludwig-Maximilians-Universität München, Sept. 2017.

[66] Happ-Kurz, C. Object-Oriented Software for Functional Data. *Journal of Statistical Software*, 93(1):1–38, Apr. 2020.

[67] Happ-Kurz, C. *MFPCA: Multivariate Functional Principal Component Analysis for Data Observed on Different Dimensional Domains*, 2020. R package version 1.3-6.

[68] Hastie, T., Tibshirani, R., and Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. Springer Series in Statistics. Springer-Verlag, New York, 2 edition, 2009.

[69] Hayfield, T. and Racine, J. S. Nonparametric Econometrics: The np Package. *Journal of Statistical Software*, 27(1):1–32, July 2008.

[70] He, K., Gkioxari, G., Dollár, P., and Girshick, R. Mask R-CNN. *arXiv:1703.06870 [cs]*, Jan. 2018.

[71] Henaff, M., Canziani, A., and LeCun, Y. Model-Predictive Policy Learning with Uncertainty Regularization for Driving in Dense Traffic. *arXiv:1901.02705 [cs, stat]*, Jan. 2019.

[72] Hernando Bernabé, A. Development of a Python package for Functional Data Analysis. Depth measures, applications and clustering. June 2019.

[73] Horvàth, L. and Kokoszka, P. *Inference for Functional Data with Applications*. Springer Series in Statistics. Springer-Verlag, New York, 2012.

[74] Hsing, T. and Eubank, R. *Theoretical Foundations of Functional Data Analysis, with an Introduction to Linear Operators*. John Wiley & Sons, May 2015.

[75] Hu, X. and Yao, F. Sparse Functional Principal Component Analysis in High Dimensions. *arXiv:2011.00959 [stat]*, Nov. 2020.

[76] Hu, Y., Zhan, W., and Tomizuka, M. A Framework for Probabilistic Generic Traffic Scene Prediction. *arXiv:1810.12506 [cs, stat]*, Oct. 2018.

[77] Hubert, L. and Arabie, P. Comparing partitions. *Journal of Classification*, 2 (1):193–218, Dec. 1985.

[78] Ieva, F., Paganoni, A. M., Pigoli, D., and Vitelli, V. 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.

[79] Inland Transport Commitee. Report of the sixty-eighth session of the Working Party on Road Traffic Safety, Apr. 2014. Economic Commission for Europe.

[80] ISO. 26262: Road vehicles-Functional safety. *International Standard ISO/FDIS*, 26262, 2011.

[81] Izquierdo, R., Quintanar, A., Parra, I., Fernandez-Llorca, D., and Sotelo, M. A. Vehicle Trajectory Prediction in Crowded Highway Scenarios Using Bird Eye View Representations and CNNs. *arXiv:2008.11493 [cs]*, Aug. 2020.

[82] Jacques, J. and Preda, C. Funclust: A curves clustering method using functional random variables density approximation. *Neurocomputing*, 112:164–171, July 2013.

[83] Jacques, J. and Preda, C. Functional data clustering: a survey. *Advances in Data Analysis and Classification*, 8(3):24, Jan. 2014.

[84] Jacques, J. and Preda, C. Model-based clustering for multivariate functional data. *Computational Statistics and Data Analysis*, 71:92–106, June 2014.

[85] Johnson, W. B., Schechtman, G., and Zinn, J. Best Constants in Moment Inequalities for Linear Combinations of Independent and Exchangeable Random Variables. *The Annals of Probability*, 13(1):234–253, 1985.

[86] Karhunen, K. *Über lineare Methoden in der Wahrscheinlichkeitsrechnung.* PhD thesis, (Sana), Helsinki, 1947.

[87] Kayano, M., Dozono, K., and Konishi, S. Functional Cluster Analysis via Orthonormalized Gaussian Basis Expansions and Its Application. *Journal of Classification*, 27(2):211–230, Sept. 2010.

[88] Krajewski, R., Moers, T., Bock, J., Vater, L., and Eckstein, L. The rounD Dataset: A Drone Dataset of Road User Trajectories at Roundabouts in Germany. submitted.

[89] Krajewski, R., Bock, J., Kloeker, L., and Eckstein, L. The highD Dataset: A Drone Dataset of Naturalistic Vehicle Trajectories on German Highways for Validation of Highly Automated Driving Systems. *arXiv:1810.05642 [cs, stat]*, Oct. 2018.

[90] LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541–551, Dec. 1989.

[91] Li, Y. and Hsing, T. Uniform convergence rates for nonparametric regression and principal component analysis in functional/longitudinal data. *The Annals of Statistics*, 38(6):3321–3351, Dec. 2010.

[92] Loève, M. Sur les fonctions aléatoires stationnaires de second ordre. *La Revue Scientifique, 5. Série*, 83:297–303, 1945.

[93] Löning, M., Bagnall, A., Ganesh, S., Kazakov, V., Lines, J., and Király, F. J. sktime: A Unified Interface for Machine Learning with Time Series. *arXiv:1909.07872 [cs, stat]*, Sept. 2019.

[94] Marron, J. S., Ramsay, J. O., Sangalli, L. M., and Srivastava, A. Functional data analysis of amplitude and phase variation. *Statist. Sci.*, 30(4):468–484, 11 2015.

[95] Mercat, J., Zoghby, N. E., Sandou, G., Beauvois, D., and Gil, G. P. Inertial Single Vehicle Trajectory Prediction Baselines and Applications with the NGSIM Dataset. *arXiv:1908.11472 [cs]*, Aug. 2019.

[96] Mercer, J. 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.

[97] Messaoud, K., Yahiaoui, I., Verroust-Blondet, A., and Nashashibi, F. Non-local Social Pooling for Vehicle Trajectory Prediction. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 975–980, June 2019.

[98] Messaoud, K., Yahiaoui, I., Verroust, A., and Nashashibi, F. Attention Based Vehicle Trajectory Prediction. *IEEE Transactions on Intelligent Vehicles*, 2020.

[99] Montanino, M. and Punzo, V. Making NGSIM Data Usable for Studies on Traffic Flow Theory. *Transportation Research Record: Journal of the Transportation Research Board*, 2390:99–111, Dec. 2013.

[100] Nadaraya, E. A. On Estimating Regression. *Theory of Probability & Its Applications*, 9(1):141–142, Jan. 1964.

[101] Oishi, K., Faria, A., Jiang, H., Li, X., Akhter, K., Zhang, J., Hsu, J. T., Miller, M. I., van Zijl, P. C. M., Albert, M., Lyketsos, C. G., Woods, R., Toga, A. W., Pike, G. B., Rosa-Neto, P., Evans, A., Mazziotta, J., and Mori, S. Atlas-based whole brain white matter analysis using large deformation diffeomorphic metric mapping: application to normal elderly and Alzheimer's disease participants. *NeuroImage*, 46(2):486–499, June 2009.

[102] Park, J. and Ahn, J. Clustering multivariate functional data with phase variation. *Biometrics*, 73(1):324–333, 2017.

[103] Parodi, A., Patriarca, M., Sangalli, L., Secchi, P., Vantini, S., and Vitelli, V. *fdakma: Functional Data Analysis: K-Mean Alignment*, 2015. R package version 1.2.1.

[104] Pelleg, D. and Moore, A. X-means: Extending K-means with Efficient Estimation of the Number of Clusters. In *In Proceedings of the 17th International Conf. on Machine Learning*, pages 727–734. Morgan Kaufmann, 2000.

[105] Punzo, V., Borzacchiello, M. T., and Ciuffo, B. F. Estimation of Vehicle Trajectories from Observed Discrete Positions and Next-Generation Simulation Program (NGSIM) Data. 2009.

[106] Punzo, V., Borzacchiello, M. T., and Ciuffo, B. On the assessment of vehicle trajectory data accuracy and application to the Next Generation SIMulation

(NGSIM) program data. *Transportation Research Part C Emerging Technologies*, 19:1243–1262, Dec. 2011.

[107] Raffaelli, L., Fayolle, G., and Vallée, F. ADAS Reliability and Safety. page 10, Oct. 2016.

[108] Ramsay, J. and Silverman, B. W. *Functional Data Analysis*. Springer Series in Statistics. Springer-Verlag, New York, 2 edition, 2005.

[109] Ramsay, J. O. When the data are functions. *Psychometrika*, 47(4):379–396, Dec. 1982.

[110] Ramsay, J. O. and Silverman, B. W. *Applied Functional Data Analysis: Methods and Case Studies*. Springer Series in Statistics. Springer-Verlag, New York, 2002.

[111] Ramsay, J. O., Hooker, G., and Graves, S. *Functional Data Analysis with R and MATLAB*. Use R! Springer-Verlag, New York, 2009.

[112] Ramsay, J. O., Graves, S., and Hooker, G. *fda: Functional Data Analysis*, 2020. R package version 5.1.5.1.

[113] Rand, W. M. Objective Criteria for the Evaluation of Clustering Methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

[114] Reed, M. and Simon, B. *Methods of Modern Mathematical Physics: Functional analysis*. Academic Press, 1980.

[115] Reiss, A., Indlekofer, I., Schmidt, P., and Van Laerhoven, K. Deep PPG: Large-Scale Heart Rate Estimation with Convolutional Neural Networks. *Sensors (Basel, Switzerland)*, 19(14), July 2019.

[116] Revuz, D. and Yor, M. *Continuous Martingales and Brownian Motion*. Springer Science & Business Media, Mar. 2013.

[117] Ross, H.-L. Why Functional Safety in Road Vehicles? In Ross, H.-L., editor, *Functional Safety for Road Vehicles: New Challenges and Solutions for E-mobility and Automated Driving*, pages 7–39. Springer International Publishing, Cham, 2016.

[118] Ruppert, D., Sheather, S. J., and Wand, M. P. An Effective Bandwidth Selector for Local Least Squares Regression. *Journal of the American Statistical Association*, 90(432):1257–1270, 1995.

[119] Schmutz, A., Jacques, J., and Bouveyron, C. *funHDDC: Univariate and Multivariate Model-Based Clustering in Group-Specific Functional Subspaces*, 2019. R package version 2.3.0.

[120] Schmutz, A., Jacques, J., Bouveyron, C., Cheze, L., and Martin, P. Clustering multivariate functional data in group-specific functional subspaces. *Computational Statistics*, Feb. 2020.

[121] Schwarz, G. Estimating the Dimension of a Model. *The Annals of Statistics*, 6 (2):461–464, 1978.

[122] Scrucca, L., Fop, M., Murphy, T. B., and Raftery, A. E. mclust 5: clustering, classification and density estimation using Gaussian finite mixture models. *The R Journal*, 8(1):289–317, 2016.

[123] Serfling, R. J. *Approximation Theorems of Mathematical Statistics*. John Wiley & Sons, Sept. 2009.

[124] Singhal, A. and Seborg, D. Clustering multivariate time-series data. *Journal of Chemometrics*, 19:427–438, Aug. 2005.

[125] Soueidatt, M., Preda, C., Jacques, J., and Kubicki, V. *Funclustering: A package for functional data clustering.*, 2014. R package version 1.0.1.

[126] Staerman, G., Mozharovskyi, P., Clémençon, S., and d'Alché Buc, F. Functional Isolation Forest. In *Asian Conference on Machine Learning*, pages 332–347. PMLR, Oct. 2019.

[127] Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., Vasudevan, V., Han, W., Ngiam, J., Zhao, H., Timofeev, A., Ettinger, S., Krivokon, M., Gao, A., Joshi, A., Zhao, S., Cheng, S., Zhang, Y., Shlens, J., Chen, Z., and Anguelov, D. Scalability in Perception for Autonomous Driving: Waymo Open Dataset. *arXiv:1912.04838 [cs, stat]*, Mar. 2020.

[128] Tan, C. W., Bergmeir, C., Petitjean, F., and Webb, G. I. Monash University, UEA, UCR Time Series Regression Archive. *arXiv:2006.10996 [cs, stat]*, June 2020.

[129] Tarpey, T. and Kinateder, K. K. J. Clustering Functional Data. *Journal of Classification*, 20(1):093–114, May 2003.

[130] Tavenard, R., Faouzi, J., Vandewiele, G., Divo, F., Androz, G., Holtz, C., Payne, M., Yurchak, R., Russwurm, M., Kolar, K., and Woods, E. Tslearn, A Machine Learning Toolkit for Time Series Data. *Journal of Machine Learning Research*, 21(118):1–6, 2020.

[131] The United Nations. Convention on Road Traffic, Nov. 1968.

[132] Tibshirani, R., Walther, G., and Hastie, T. Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(2):411–423, 2001.

[133] Tokushige, S., Yadohisa, H., and Inada, K. Crisp and fuzzy k-means clustering algorithms for multivariate functional data. *Computational Statistics*, 22(1): 1–16, Apr. 2007.

[134] Traore, O. I., Cristini, P., Favretto-Cristini, N., Pantera, L., Vieu, P., and Viguier-Pla, S. Clustering acoustic emission signals by mixing two stages dimension reduction and nonparametric approaches. *Computational Statistics*, 34 (2):631–652, June 2019.

[135] Tsybakov, A. B. *Introduction to Nonparametric Estimation*. Springer Series in Statistics. Springer New York, New York, NY, 2009.

[136] Tucker, J. D. *fdasrvf: Elastic Functional Data Analysis*, 2020. R package version 1.9.4.

[137] von Luxburg, U. *Clustering Stability: An Overview*. Now Publishers Inc, 2010.

[138] Wang, J.-L., Chiou, J.-M., and Müller, H.-G. Functional Data Analysis. *Annual Review of Statistics and Its Application*, 3(1):257–295, 2016.

[139] Watson, G. S. Smooth Regression Analysis. *Sankhyā: The Indian Journal of Statistics; Series A*, 26, 1964.

[140] Wendel, J. G. Note on the Gamma Function. *The American Mathematical Monthly*, 55(9):563–564, 1948.

[141] Wong, R. K. W. and Zhang, X. Nonparametric operator-regularized covariance function estimation for functional data. *Computational Statistics & Data Analysis*, 131:131–144, Mar. 2019.

[142] Wong, R. K. W., Li, Y., and Zhu, Z. Partially Linear Functional Additive Models for Multivariate Functional Data. *Journal of the American Statistical Association*, 114(525):406–418, Jan. 2019.

[143] Wood, S. N. Low-rank scale-invariant tensor product smooths for generalized additive mixed models. *Biometrics*, 62(4):1025–1036, 2006.

[144] Wu, Y., Hou, J., Chen, G., and Knoll, A. Trajectory Prediction Based on Planning Method Considering Collision Risk. In *2020 5th International Conference on Advanced Robotics and Mechatronics (ICARM)*, pages 466–470, Dec. 2020.

[145] Wulfe, B., Chintakindi, S., Choi, S.-C. T., Hartong-Redden, R., Kodali, A., and Kochenderfer, M. J. Real-time Prediction of Intermediate-Horizon Automotive Collision Risk. *arXiv:1802.01532 [cs]*, Feb. 2018.

[146] Yao, F., Müller, H.-G., and Wang, J.-L. Functional Data Analysis for Sparse Longitudinal Data. *Journal of the American Statistical Association*, 100(470):577–590, June 2005.

[147] Young, N. *An Introduction to Hilbert Space*. Cambridge University Press, Cambridge, 1988.

[148] Zambom, A. Z., Collazos, J. A., and Dias, R. Selection of the Number of Clusters in Functional Data Analysis. *arXiv:1905.00977 [stat]*, May 2019.

[149] Zambom, A. Z., Collazos, J. A. A., and Dias, R. Functional data clustering via hypothesis testing k-means. *Computational Statistics*, 34(2):527–549, June 2019.

[150] Zhang, J., Siegle, G. J., D'Andrea, W., and Krafty, R. T. Interpretable Principal Components Analysis for Multilevel Multivariate Functional Data, with Application to EEG Experiments. *arXiv:1909.08024 [stat]*, Sept. 2019.

[151] Zhang, J.-T. and Chen, J. Statistical inferences for functional data. *The Annals of Statistics*, 35(3):1052–1079, July 2007.

[152] Zhang, X. and Wang, J.-L. From sparse to dense functional data and beyond. *The Annals of Statistics*, 44(5):2281–2321, Oct. 2016.

[153] Zhang, X. and Wang, J.-L. Optimal weighting schemes for longitudinal and functional data. *Statistics & Probability Letters*, 138:165–170, July 2018.

UNIVERSITE BRETAGNE LOIRE / MATHSTIC

ENSAI École nationale de la statistique et de l'analyse de l'information

Titre : Méthodes statistiques pour données fonctionnelles multivariées

Mot clés : analyse de données fonctionnelles ; analyse en composantes principales fonctionnelles ; lissage optimal ; groupement *model-based* ; régularité

Résumé : Le sujet de cette thèse est lié à l'analyse de données fonctionnelles et est motivé par l'analyse de données provenant de l'industrie automobile. Les méthodes standards concernant les données fonctionnelles sont basées sur l'hypothèse que les courbes sont observées de façon continue et sans erreur. Or, en pratique, c'est rarement le cas. Pour cette raison, une étape cruciale est de reconstruire les trajectoires à partir de mesures bruitées ayant des instants d'observations discrets et alatoires. Pour cela, nous proposons une approche originale : l'utilisation de la régularité locale du processus générant les courbes. Ainsi, utilisant le grand nombre de trajectoires, ainsi que leur variabilité intrinsèque, nous proposons un estimateur simple de cette régularité locale. Munis de cet estimateur, nous construisons un estimateur par polynômes locaux, quasiment optimal, des courbes à partir d'un échantillon de courbes bruitées. Des estimateurs non-paramétriques des fonctions moyenne et covariance pour données fonctionnelles, basés sur la régularité locale du processus, sont développés. De plus, un algorithme de groupement, de type *model-based*, pour une classe générale de données fonctionnelles pour laquelle les composantes peuvent être des courbes ou des images est présenté. Les résultats sur des données réelles et simulées montrent les bonnes performances de ces méthodes. Un package Python, implementant celles-ci et disponible publiquement, a été développé.

Title: Statistical methods for multivariate functional data

Keywords: functional data analysis; functional principal component analysis; model-based clustering; optimal smoothing; regularity

Abstract: The topic of this thesis is related to functional data analysis and is motivated by modern data from automobile industry. The standard functional data methods rely on the assumption that the curves are continuously observed, without error. However, in general, the real data is neither continuously nor exactly observed. Therefore, a crucial step is to recover the trajectories from noisy measurements at discrete random points. For that, we propose an original point of view: the local regularity of the process generating the curves. Thus, combining information both within and across trajectories, we propose a simple estimator for this local regularity. Given this estimate, we build a nearly optimal local polynomial smoother of the curves from a sample of noisy trajectories. Nonparametric estimators for the mean and the covariance functions of functional data, using the local regularity of the process, are derived. Moreover, we propose a model-based clustering algorithm for a general class of functional data for which the components could be curves or images. Results of both simulated and real data show the good performances of this methods. A Python package, implementing the methods and publicly available, has been developed.