Université Grenoble Alpes

THÈSE

Pour obtenir le grade de

DOCTEUR DE L' UNIVERSITÉ GRENOBLE ALPES

Spécialité : **Mathématiques Appliquées**

Arrêté ministériel : 25 mai 2016

Présentée par

Alexandre CONSTANTIN

Thèse dirigée par **Stéphane GIRARD**, Directeur de recherche, Inria et co-encadrée par **Mathieu FAUVEL**, Chargé de recherche, INRAe Toulouse

préparée au sein du **Laboratoire Jean Kuntzmann**
dans **l'École Doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique**

Time-series Analysis of Massive Satellite Images: Application to Earth Observation

Analyse de Séries Temporelles Massives d'Images Satellitaires : Applications à la Cartographie des Écosystèmes

Thèse soutenue publiquement le **13 décembre 2021**,
devant le jury composé de :

Mr, Stéphane GIRARD
Directeur de recherche, Inria Grenoble Rhône-Alpes, Directeur de thèse
Mr, Mathieu FAUVEL
Chargé de recherche, INRAe Toulouse, Co-Encadrant de thèse
Mr, Julien JACQUES
Professeur des universités, Université Lumière Lyon 2, Rapporteur
Mr, Gabriele MOSER
Professeur, University of Genoa, Rapporteur
Mme, Marie CHABERT
Professeure des universités, INP-ENSEEIHT Toulouse, Examinatrice, Présidente
Mr, Lionel BOMBRUN
Maître de conférence, Bordeaux Sciences Agro, Examinateur
Mr, Yannick TANGUY
Ingénieur, Centre National d'Études Spatiales Toulouse, Invité
Mr, Serge IOVLEFF
Maître de conférence, Université Lille 1, Invité

RÉSUMÉ

Cette thèse s'inscrit dans le contexte de l'exploitation des données issues de la mission Sentinel-2. Cette mission, initiée par l'Agence spatiale européenne et lancée en 2015, produit massivement des séries temporelles d'images satellite (SITS). Parmi les axes d'analyse de ces images, cette thèse se concentre plus particulièrement sur la classification, c'est-à-dire la production de cartes d'occupation ou d'utilisation des sols en utilisant l'aspect spectro-temporel des SITS issues de Sentinel-2.

Les deux principales difficultés auxquelles ces travaux de thèse se confrontent sont les suivantes. Tout d'abord, la quantité sans précédent de données nécessite à la fois la mise en œuvre de classifieurs capables de passer à l'échelle et l'utilisation de techniques d'optimisation de code (telles que le traitement parallèle). Deuxièmement, le bruit d'acquisition (nuages, ombres) combiné à l'aspect temporel des données résulte en un échantillonnage irrégulier des séries temporelles. Les approches conventionnelles ré-échantillonnent les séries temporelles sur une grille unique, puis elles utilisent des méthodes d'apprentissage vectorielles pour classer à grande échelle (échelle nationale). Cette démarche en deux étapes a pour inconvénient principal d'alourdir le nombre de traitements appliqués aux SITS, rendant les traitements plus complexes. Dans une moindre mesure, l'étape de ré-échantillonnage peut légèrement altérer les caractéristiques temporelles de la donnée.

Les contributions présentées dans cette thèse sont les suivantes. Nous introduisons une nouvelle approche statistique ayant la capacité de classer des séries temporelles avec un échantillonnage irrégulier basée sur un mélange de processus gaussiens multivariés. Une approche en deux étapes a été proposée, en définissant dans un premier temps un modèle sur des séries temporelles uni-dimensionnelles et indépendantes au sens de l'indépendance spectrale, puis en considérant dans un second temps conjointement les informations spectrales et temporelles des SITS. Ces modèles permettent, de surcroît, une reconstruction de données à des instants non observés ou bruités. L'estimation des deux modèles repose sur un code python parallélisé afin d'être exploitable sur les données de très grande taille. Les deux modèles sont évalués numériquement sur les SITS issues de Sentinel-2 en terme de classification et de reconstruction et sont comparés aux approches conventionnelles. L'analyse des résultats illustre la pertinence des deux modèles et le bénéfice de disposer de modèles paramétriques interprétables.

Mots clés : Processus gaussiens, Analyse de séries temporelles, Optimisation, Données massives, Observation de la Terre.

ABSTRACT

This thesis takes place in the context of the processing of the data from Sentinel-2 mission. This mission, initiated by the European Space Agency and launched in 2015, produces an unprecedented amount of Satellite Image Time-Series (SITS). Among the key analyses of these images, this thesis focuses on the classification task, *i.e.* land use or land cover maps that can be produced using spectro-temporal aspect of the Sentinel-2 SITS.

Two main difficulties are identified in this thesis for the process of Sentinel-2 SITS. First, the unprecedented amount of data requires both scalable classifiers and code optimization techniques (such as parallel processing). Second, the acquisition noise (clouds, shadows) combined with the temporal aspect results in irregular and unevenly sampled time-series. Conventional approaches re-sample time-series to a set of time stamps, then they use machine learning techniques to classify vectors at a large-scale (national scale). The main disadvantage of this two-step processing approach is that it increases the number of operations applied to the SITS, implying a more difficult transition to massive amount of data. To a lower extent, the re-sampling step may slightly alter the temporal characteristics of the data.

This thesis contributions are the following. We introduce a novel model-based approach with the ability to classify irregularly sampled time-series based on a mixture of multivariate Gaussian processes. A two-step approach has been used, by defining on one hand a model of uni-variate time-series, independent from the spectral wavelength point of view, then by considering on the second hand both spectral and temporal information from SITS. These models allow jointly a reconstruction of unobserved or noisy data. Estimation of both models has been implemented using a parallelized python code to be scalable to large-scale data-sets. The two models are evaluated numerically on Sentinel-2 SITS in terms of classification and reconstruction accuracy and are compared with conventional approaches. Analyses of the results illustrate the relevance of the two models and the benefit of using interpretable parametric models.

Index terms: Gaussian Processes, Time Series analysis, Optimization, Big Data, Earth Observation.

Acknowledgements / Remerciements

To start in English, I would like to thank all the members of the jury. Thank you Julien and Gabriele for accepting to review my manuscript, I really enjoyed reading your comments and feedback on my work. Thanks also to Marie and Lionel, it was a pleasure to answer your questions during my oral defence. Finally, thanks to Serge and Yannick, I enjoyed talking to you both during the three years of my thesis. The following is written in French.

Je tiens tout d'abord à remercier Stéphane et Mathieu pour m'avoir offert l'opportunité de faire cette thèse. C'était un plaisir d'avoir pu collaborer avec vous pendant trois ans et je pense que toutes vos exigences pour le moindre détail (chaque graphe, chaque mot, jusqu'au contenu du code), m'ont vraiment permis de clôturer sur ce manuscrit que j'espère être clair, concis et bien illustré. J'esp'ere que nous continuerons à l'avenir d'échanger sur les problématiques soulevées dans cette thèse.

Je tiens également à remercier l'équipe Mistis (maintenant Statify) pour son accueil, en commençant par Florence la chef, JB, Sophie, Julyan et Pedro. Merci aussi aux plus anciens doctorants: Alexis, Brice, Thibault, Clément, Karina, Veronica et Fabien. Une mention spéciale aux doctorants avec qui j'ai partagé ces trois années: Meryem pour les randonnées, Masha pour les anniversaires, toutes les petites attentions et d'avoir proposé le café à 10h.. Enfin Benoit, merci pour les discussions toujours intéressantes ainsi que ton aide sur la rigueur mathématique ! Je souhaite maintenant bon courage aux doctorants qui continuent: Dasha (et ses animaux), Théo, Lucrezia, Hana, MinhTri, Louise et Yuchen. Je salue aussi les post-doctorants actuels et anciens de l'équipe: Antoine, Fei, Hongliang, Pascal, Pierre et Argheesh.

Je tiens à remercier l'équipe enseignante de Grenoble-INP, merci Olivier et Florent pour votre confiance lorsque vous m'avez confié des BEs, TPs, TDs et cours. J'ai vraiment apprécié ce travail d'enseignement au cours de ma thèse. Merci aussi pour vos retours sur les divers éléments constituants le métier d'enseignant-chercheur.

Pour terminer, je souhaite remercier ma famille pour tout leur soutien. Cela n'a pas été simple au début d'expliquer pourquoi je souhaitais continuer encore trois années d'études, finalement je pense qu'il ne subsiste plus aucun doute sur ma motivation. Merci aussi à Qi qui m'a soutenu tous les jours pendant ces trois années.

CONTENTS

General Introduction

The context of this PhD thesis is part of a multi-disciplinary research project between remote sensing and applied statistics. This PhD has been done in the Statify team, a joint team between LJK (Jean-Kuntzmann laboratory) and Inria Grenoble and in collaboration with the CESBIO lab, a joint unit with the French national space center (CNES), the French National Centre for Scientific Research (CNRS), the French National Research Institute for Sustainable Development (IRD), Toulouse University (UT2) and the National Research Institute for Agriculture, Food and Environment (INRAe), Toulouse. It has been done under a grant from the CNES and the Grenoble institute of technology (Grenoble INP) in the context of an IRS project (ANR-15-IDEX-02). CNES has also provided data and computational resources.

Finally, in parallel to this thesis, a so-called *label* has been also obtained with focuses on teaching. It consists of a reflection on the teaching practices and applications on concrete cases in the classroom (experimentation of formats or modes of evaluation for example).

Earth observation using satellite image time-series

With the increasing number of successful launches of orbital satellites, earth observations are more and more used for environmental and climatic monitoring such as land use or land cover. The context of this work focuses on Sentinel-2 constellation, initiated by the European Space Agency with the Copernicus mission. Launched in 2015, it produces an unprecedented amount of time-series thanks to its revisit cycle. These data are used for military and civilian applications. The very high ground resolution allows ecosystems mapping, forests and agricultural plots monitoring. Furthermore, the Sentinel-2 mission covers the complete European area which makes the data usable at a country scale for a country like France. At such spatial and temporal scale, the data contains numerous clouds, shadows, snows (without taking into account perpetual snow) or any other interference with the ground information.

In order to produce a clean map, the time-series at large scale is conventionally done in two steps:

1. **Temporal re-sampling**: the data are sampled onto a regular temporal grid, discarding clouds and shadows dates.

2. **Classification**: assigning labels to pixels from the satellite image time-series.

The pre-processing step to reconstruct irregularly sampled time-series may be costly and adds computational run-time and memory usage which is critical on large data-sets as Sentinel-2 images.

Contributions

This work aims at combining statistical modelling with applications to remote sensing using Sentinel-2 images. It proposes a one-step approach by defining a new classifier for irregularly and unevenly sampled multi-dimensional time-series. The contributions are based on Gaussian processes.

The contributions of this thesis can be summarized in two points corresponding to the definition of two models:

1. The *Mixture of Multi-variate Independent Gaussian Processes* (MIMGP or MIGP) model. This model assumes independence between the spectral bands of the satellite images and classify irregularly sampled time-series. This work has been published [41]. In practice, it is known that the independence assumption is not realistic, this leads us to the second contribution.

2. The *Mixture of Multi-variate Gaussian Processes* (M2GP) model. This model assumes a linear dependency with respect to latent processes which improves the classification results of MIMGP. It defines a generative model for multi-output classification. This work has been submitted [42].

Both models are optimized numerically to scale to Sentinel-2 data-sets. In particular, the implementation of the first model uses the "just-in-time" python compiler and pre-compiled code.

Communications

This work has been presented in:

National conference:

Alexandre Constantin, Mathieu Fauvel, Stéphane Girard, Serge Iovleff. Classification de Signaux Multidimensionnels Irrégulièrement Échantillonnés. *GRETSI 2019 - 27e Colloque francophone de traitement du signal et des images*, Aug 2019, Lille, France.

National workshops:

Alexandre Constantin, Mathieu Fauvel, Stéphane Girard, Serge Iovleff, Yannick Tanguy. Classification de Signaux Multidimensionnels Irrégulièrement Echantillonnés. 2019 - *Journée Jeunes Chercheurs MACLEAN du GDR MADICS*, Dec 2019, Paris, France.

Alexandre Constantin, Mathieu Fauvel, Stéphane Girard, Serge Iovleff. Supervised classification of multidimensional and irregularly sampled signals. *Statlearn 2019 - Workshop on Challenging problems in Statistical Learning*, Apr 2019, Grenoble, France.

Outline of this thesis

The remainder of this thesis is organized as follow:

- In Chapter 1, an in-depth review of SITS classification is done. It describes the Sentinel-2 data-sets to produce land use or land cover maps. It also reviews state-of-the-art classifiers in the remote sensing literature.
- In Chapter 2, we review state-of-the-art in statistical modelling, for supervised model-based classification which is the framework of our work. These models are described for different type of inputs, in particular distributions on real-input vectors for time-series classification and extensions to real-input matrices. Then it introduces Gaussian processes for regression and classification, firstly by modelling one dimensional processes (as continuous time-series) and secondly by modelling multi-output processes.
- Chapter 3 summarizes the contributions of this thesis. It introduces the contributions on the representation of one-dimensional irregularly sampled time-series. Starting from the general form of the proposed models and the assumptions done for each contribution. An emphasis is done on how statistics are used to perform classification and reconstruction of irregularly sampled multi-dimensional time-series.
- Chapter 4 and Chapter 5 present the two contributions. They consist of, respectively, a first article (*to appear*) in IEEE Transactions on Geoscience and Remote Sensing and a second article (*in review*) to a statistical journal. They compare our models to state-of-the-art methods with both scores on classification and reconstruction using Sentinel-2 satellite images.

A general conclusion and a discussion are provided in the final chapter.

At last, Appendix A discusses numerical issues encountered with the estimation of the models parameters. Particularly on difficult cases where the time-series is not observed within a large temporal window. Appendix B includes the supplementary materials of [41]. Finally Appendix C highlights some technical optimizations of the code.

Chapter 1

Satellite image time-series analysis and classification

Outline

French introduction

Ce chapitre présente les séries temporelles d'images satellitaires (SITS), issues de la mission Sentinel-2. Ces données sont bruitées (bruits atmosphériques et bruit d'acquisition) et hétérogènes (diverses résolutions spatiales selon la longueur d'onde du capteur multi-spectral embarqué). Ce chapitre présente donc les données qui vont être utilisées (après correction de la réflectance atmosphérique brute en réflectance de surface, création du masque de détection de nuages et d'ombres).

Par la suite une revue de l'état de l'art est faite sur les méthodes de classification de ces séries temporelles. En particulier les machines à vecteur de support (SVM) et les forêts aléatoires (RF) par lesquelles des cartes d'occupation ou d'utilisation des sols à grande échelle sont produites. Cela met en évidence la problématique de classification liée au caractère massif des données et l'étape supplémentaire de ré-échantillonnage obligatoire pour obtenir des vecteurs de dimension fixe pour traiter les SITS par ces méthodes de classification. Les méthodes d'apprentissage profond (réseaux de neurones) sont aussi introduites et montrent qu'il est très compliqué de passer à l'échelle avec ces méthodes dites plus complexes.

Ce chapitre se termine en ouvrant la discussion sur les méthodes statistiques, réputées plus légères en nombre de paramètres, pour tenter de résoudre le problème d'échantillonnage et de production de cartes des sols à grande échelle.

Passive remote sensing satellite images are images taken by an *optical sensor* known as *Multi-* or *Hyper-*Spectral Instrument (MSI/HSI) sensor, onboarded on the satellite's payload. They differ from digital photographies taken by a camera, as aerial ortho-photos, from both mechanical and data point of views. More specifically, an optical sensor measures the reflected light at different wavelengths from the sun by a surface, see Figure 1.1, and divides it by the solar illumination. This quantity is known as *reflectance*, the reflectance is a function of the wavelength and is a property of the surface from which the light has been reflected, it takes its values between 0 and 1 (see [83]). A *wavelength*, often denoted by λ_0, is the characteristic length of a wave (from visible spectrum to infra-red in the order of magnitude of a micrometer for optical satellites, see Figure 1.2). The wavelength is inversely proportional to the wave's frequency f_0: $\lambda_0 = c/f_0$ where c is the speed of light in vacuum.

Physically, the image sensor (MSI/HSI) of a satellite is different from a camera on an electronic point of view [52, Section 3.1.2] as the materials are adapted to space constraints. Indeed, a satellite has a higher orbit than a plane (or helicopter) so an embedded telescope can be found to complete the MSI and additional protections to space constraints (electro-magnetic waves or extreme variations of temperature for example) are also embedded. An other aspect is the reception of images which is more complex for a satellite. A satellite is not able to come back to earth to copy the data. All satellites are completed with a complete array of stations on earth (see [52, Fig. 6] for Sentinel-2 mission) to download images from the satellite.

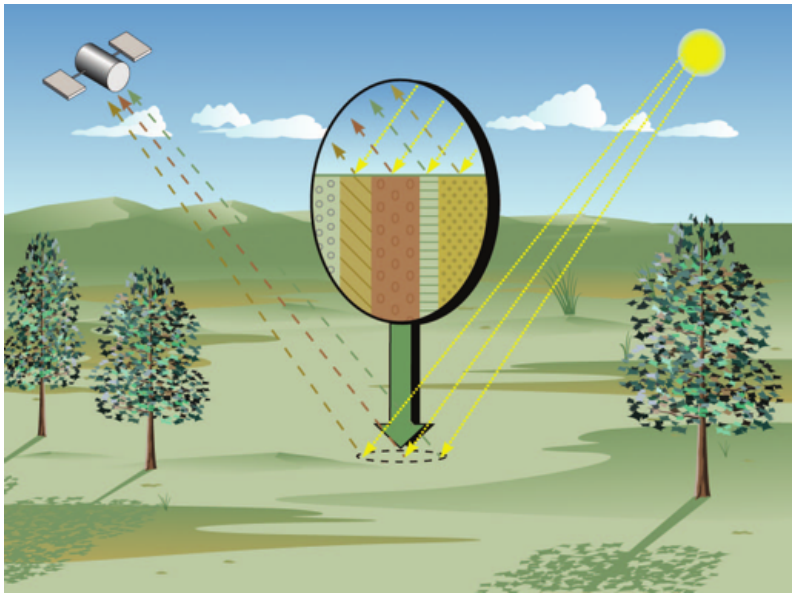


Figure 1.1: (Image from [98]). Surface reflectance measured thanks to an embedded multi-spectral instrument.

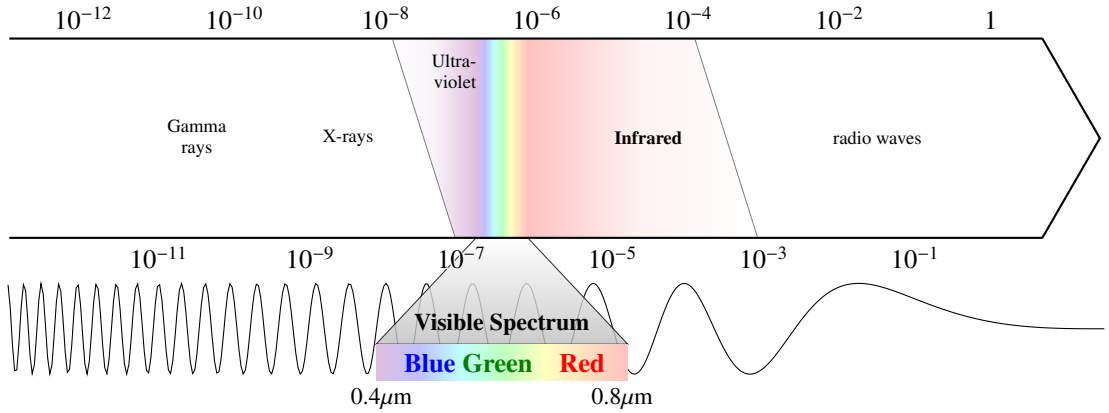


Figure 1.2: Electromagnetic spectrum (wavelengths in m). Optical satellites measure reflectance in the visible spectrum and in the infrared. The concerned wavelengths values range from 400 to 2400nm.

Satellite images are quantized in three dimensions: *spatial*, as any image has a finite number of pixels; *spectral*, as MSI/HSI have a finite number of spectral bands; and *temporal*, satellite images are taken at different times. The spatial characteristic is denoted by the *ground resolution*. It is the shape of the pixel at ground level, for the considered satellites, the pixel is usually a square and ground resolution corresponds to its side-length. Then, the main difference between *multi-* and *hyper* spectral instruments is the number of *spectral bands* acquired. Indeed MSI captures between three and a dozen spectral bands whereas HSI captures a hundred of them (higher spectral resolution). The choice of the ground resolution will differ with respect to the targeted application, for example high resolution satellites (as Pléiades family with 50cm ground resolution) are used for civil or military applications [143] and coarser resolution satellites (as Sentinel-3 with 300m ground resolution) with applications to oceanography [51]. Finally, the temporal quantization is characterized by the *revisit cycle*. It is the time between two satellite images acquisitions of the same spatial areas. A complete set of images acquired in a given time window is called a *Satellite Image Time-Series* (or SITS).

Among the Sentinel missions[1], Sentinel-2 satellites carry a MSI and are used for environmental monitoring, Land Use or Land Cover (LULC) applications among others. Sentinel-2 SITS are a set of images with the same views (same incidence angle, same altitude, ...). We refer to [13, Table 1] for an overview of the Sentinel family including optical and non-optical (Synthetic Aperture Radar) images. Sentinel-2 satellites have a 10 to 60m ground

[1]European Copernicus program, `https://www.copernicus.eu/en`

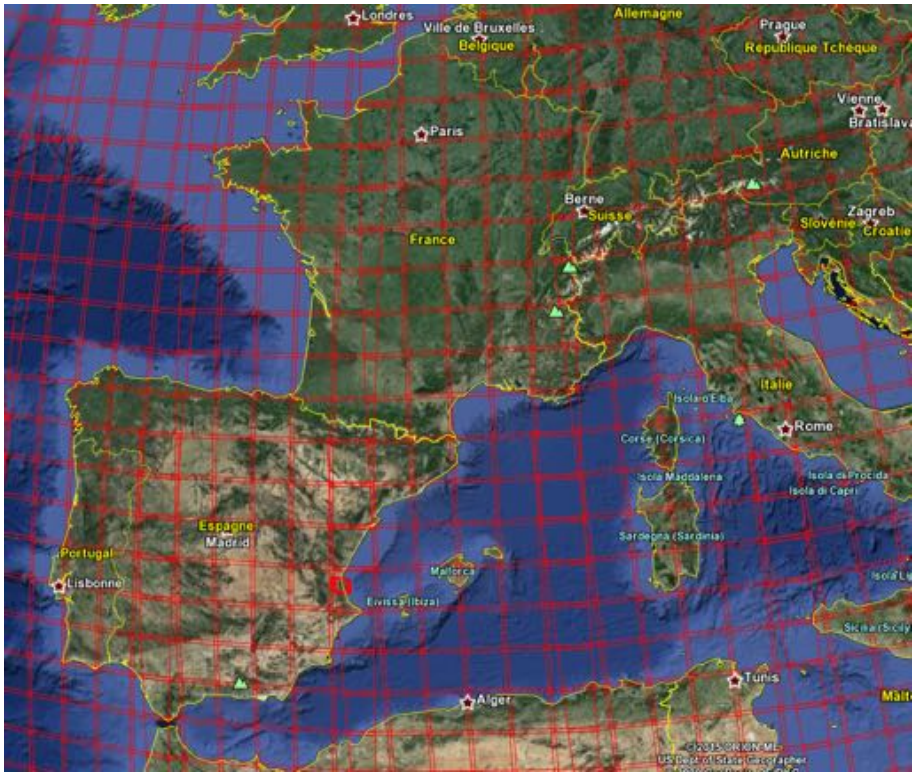


Figure 1.3: Sentinel-2X tiles, represented by red squares, over South-east Europe (image downloaded from [82]). The doubled-lines represent a 10km overlap between two consecutive tiles.

resolution sensor including 13 spectral bands and have a revisit cycle of 5 days per satellite.

Sentinel-2 multispectral satellite image time-series are detailed in Section 1.1, then a review of Satellite image time-series classification is presented in Section 1.2. Section 1.3 concludes this chapter.

1.1 Sentinel-2 multi-spectral satellite image time-series

This Section presents the SITS produced by Sentinel-2 constellation. It illustrates the time-series aspect and some basic processing done to these data for Land Use or Land Cover (LULC) applications. Section 1.1.1 presents the Sentinel-2 mission in depth. Section 1.1.2 presents the production of the reference data and the problem of mislabeled data in large scale.

1.1.1 Sentinel-2 data

Sentinel-2 [52], also called Sentinel-2X (or simply S2), is a constellation. Two optical satellites, Sentinel-2A and Sentinel-2B, are currently in orbit (polar orbit, altitude of 786 kilo-meters) and take high resolution images from earth. As an optical satellite, it mainly captures visible light bands and near infra-red spectral bands (from 400 to 2400nm approximatively). The data providers (European Space Agency, ESA and Centre National d'Études Spatiales, CNES for the French territory) provide Sentinel-2 data by *tiles*. A tile is a 100km side-length square [82]. One tile is revisited every five days thanks to low orbital path of Sentinel-2. The tiles over south-east of Europe are presented in Figure 1.3.

Sentinel-2 ground resolution and spectral wavelengths

To provide sufficient information for classification purpose, Sentinel-2 has a wide spectral covering from the visible range to short wave infrared wavelengths with a total of thirteen spectral bands. The choice of a 400 to 2400nm spectral range is the ability to focus on agricultural particularities for different crops [188]. All spectral bands, their central wavelength and the ground resolution are reported in Table 1.1.

Band name	Ground resolution (m)	Wavelength (nm)
B2 (Blue)	10	490
B3 (Green)	10	560
B4 (Red)	10	665
B5	20	705
B6	20	740
B7	20	783
B8 (near infra-red)	10	842
B8A	20	865
B11	20	1610
B12	20	2190
B1	60	443
B9	60	945
B10	60	1380

Table 1.1: Sentinel-2 bands specifications from [52]. Reported wavelengths are central wavelengths only; bandwidths and Signal to noise ratios are detailed in [52]. Figure 1.4 highlights the wavelengths positions on the spectrum (400 to 2400 nm).

Among the thirteen spectral bands, four wavelengths, including blue, green and red visible light and one infrared band, have a ground resolution of 10m. These bands are sufficient to detect vegetations or non-vegetation areas [178] as they allow the computation of vegetation indexes as NDVI, the *Normalized Difference Vegetation Index*. This index is defined as the normalized difference between the infrared (IR) and the red (R) spectral band: NDVI $= (IR - R)/(IR + R)$. Figure 1.4 shows the reflectance w.r.t. to the wavelength for grass and dried grass repeated three times, one for each ground resolution and the spectral filters for all bands are highlighted within the coloured areas. In the third graph of Figure 1.4 (10m ground resolution), it is easy to see that the difference between the reflectance in the grey zone (IR) and the red zone (R) is larger for grass than dried grass. [11] provides an application to vegetation's monitoring.

Six wavelengths in the near infrared and short-wave infrared have a ground resolution of 20m. Together with the 10m ground resolution spectral bands, they cover the complete range of S2 optical abilities. The infrared spectral bands around 800nm may be used to focus on vegetation and higher values of wavelengths focus on other behaviours.

Finally three wavelengths have a ground resolution of 60m. In general they are used to calibrate atmospheric correction.

Sentinel-2 revisit cycle

As the vegetation is changing with the seasons (for example leafs are falling from deciduous trees in Autumn), the previous spectra (Grass or Dried Grass) depend on the acquisition day in a year: the spectral signature may also change with time [19]. The following focuses on the temporal aspect thanks to the high revisit cycle of S2.

Despite the intrisinc noise of electronic components which is not of interest here, the noise considered in the following concerns environmental effects which makes the acquisition unusable. As an optical satellite, the noise sources we have is the presence of clouds and shadows. Presented in [187] and in [188], an average revisit cycle with less than 8 days allows a 70% of noise-free data, also called *clear* data. Thus, about 30% of the data are corrupted by clouds or shadows. The Figure 1.5 represents an extract of one SITS (true color composition) from France for a complete year of satellite images acquisition. We can clearly see variations of colors (which means a variation of the reflectance with respect to the time) with seasonal effects, clouds or shadows at some locations. An other tile from the city center of Toulouse, France, is represented in Figure 5.1. In the following, $\mathcal{T} = [t_{\min}, t_{\max}]$ denotes the time window for a SITS: $t_{\min}$ is close to January the 1st and $t_{\max}$ around December the 31st for a one-year SITS.

Sentinel-2 level 2A SITS and noise detection

Now that the Sentinel-2 SITS are described, what follows describes the final product, *i.e.* level 2A SITS reflectance, and the construction of the mask of data (noise detection).

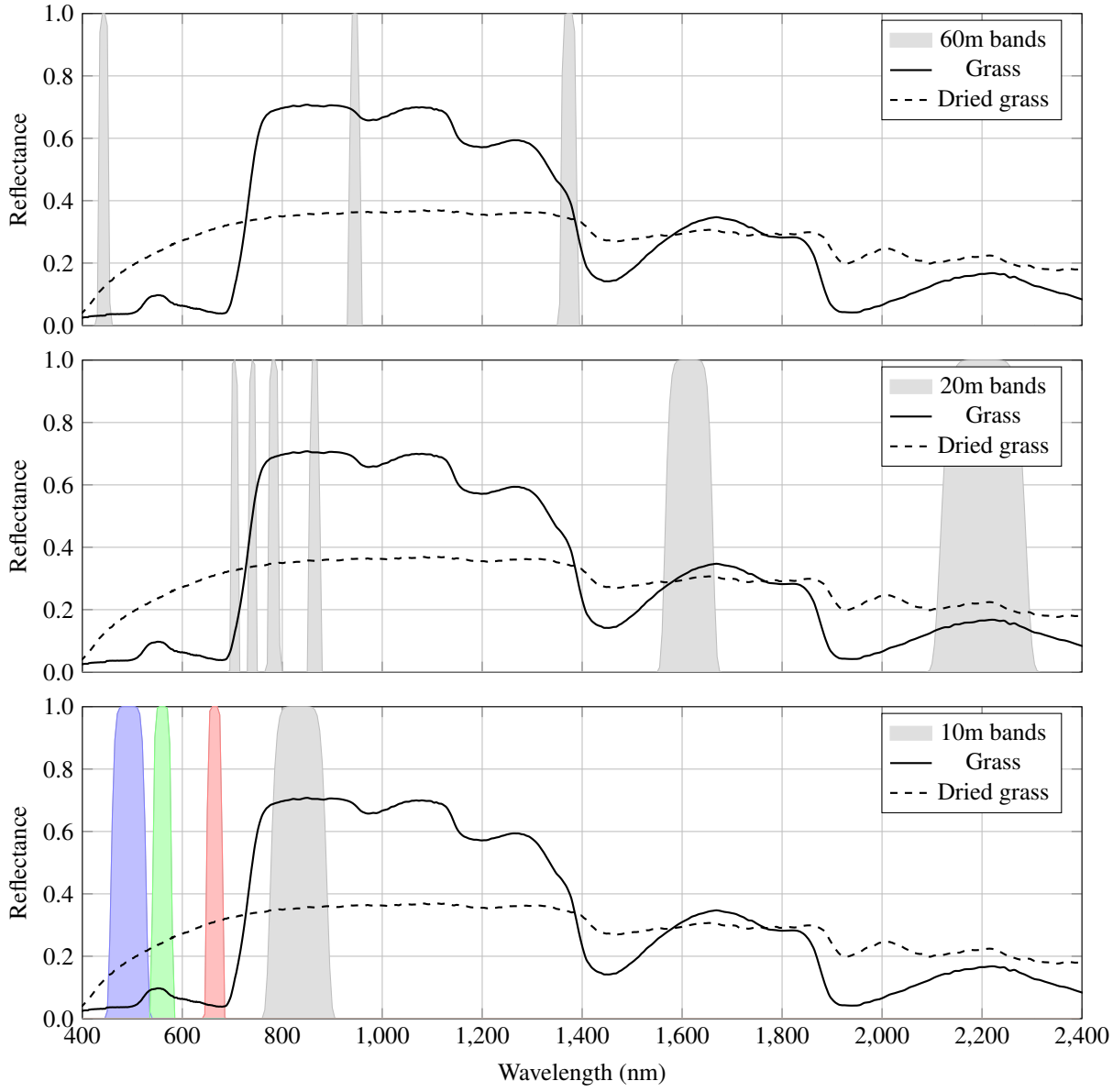


Figure 1.4: Sentinel-2 spectral band filters (coloured areas) centered around the wavelength provided in Table 1.1 and with associated bandwidths [52]. Visible light starts from 380nm and ends around 780nm: blue, green and red bands are highlighted in the 10m ground resolution bands. The spectral reflectance signature of healthy grass is reported on each grid (full line) and the signature of dried grass (dashed line).

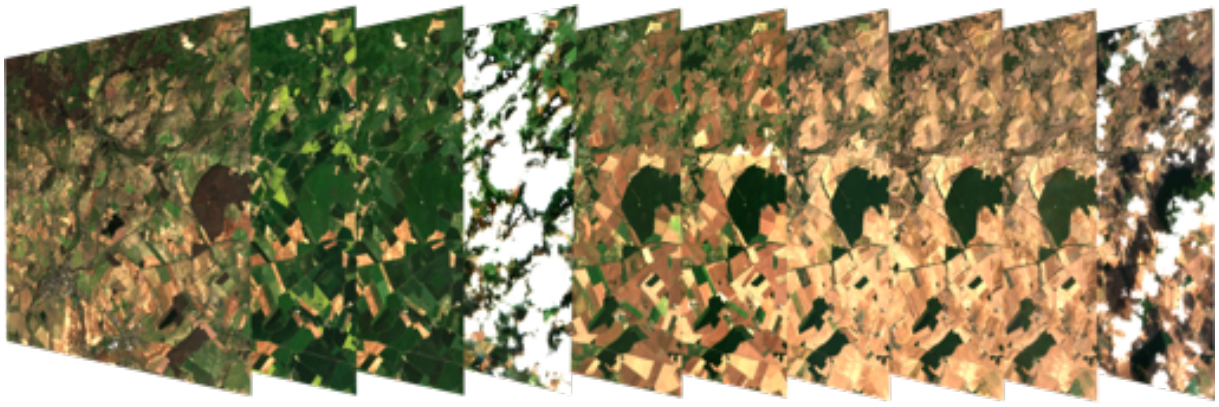


Figure 1.5: Satellite Image Time-Series true colors. The data come from year 2018 over the center of French territory. Images were downloaded from Theia Land Data Center (`https://www.theia-land.fr/en/products/`).

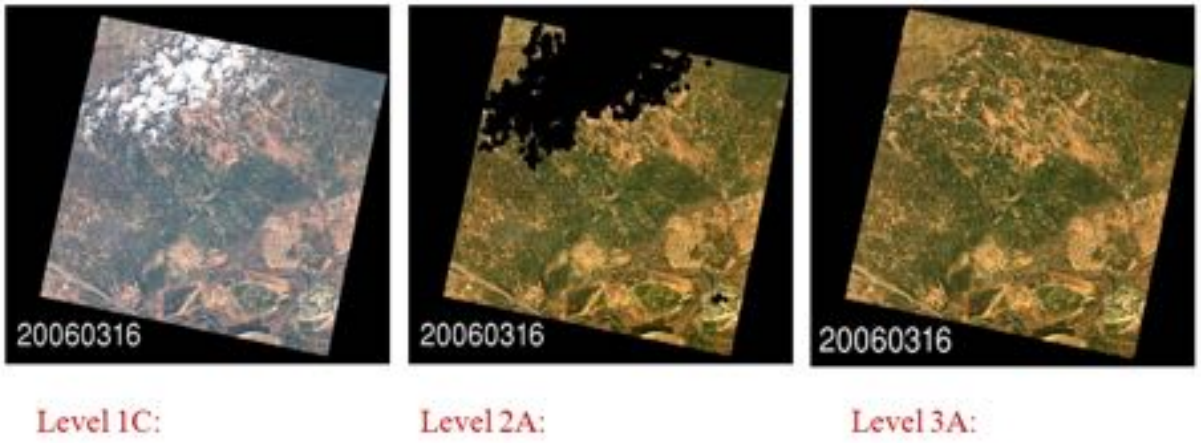


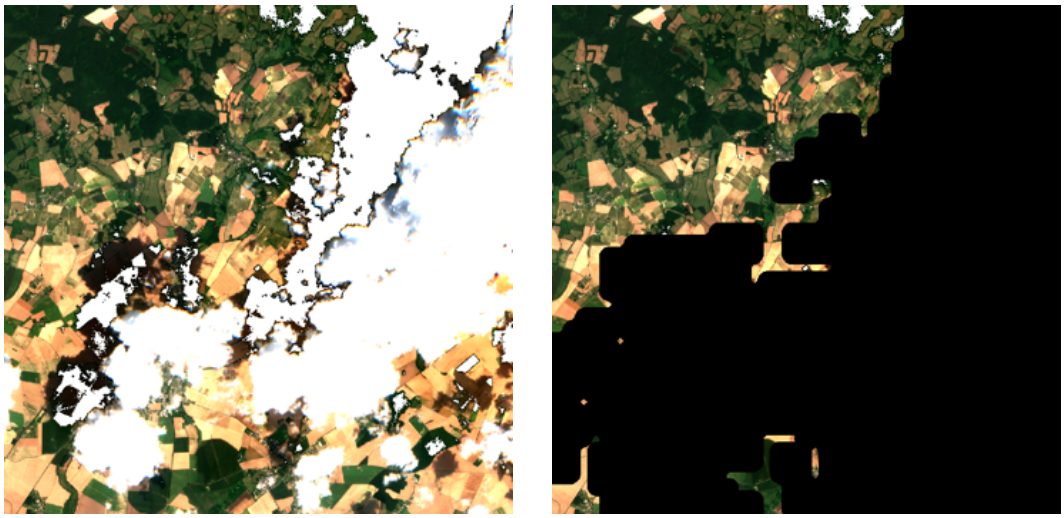
Figure 1.6: (Image from [80]). Sentinel-2 product levels: level 1C (with atmospheric interference and clouds) and level 2A (reflectance and clouds removal). Level 3A shows reconstructed satellite image.

The SITS are proposed by the data provider at different level of processing ranging from the raw data to clean an reconstructed surface reflectance. These levels are called *product levels*. Product levels available for Sentinel-2 are the ones derived from the NASA (National Aeronautics and Space Administration) in 1996, some of the product levels are presented in Figure 1.6. The multi-spectral instrument from the satellite captures a raw reflectance (level 0) which is completed with metadata (level 1A: acquisition time, geo-references, ...) and then ortho-rectified (level 1C). At levels 0 and 1, reflectances are still interfered by the atmosphere and have to be corrected to be used as surface reflectances (see Figure 1.1).

Image products are downloaded[2] at level 2A where the data are composed by surface reflectance and a mask [79]. At time $t \in \mathcal{T}$, we denote by $\mathbf{Y}(t) \in E$ the surface reflectance at multiple spectral bands and $M(t) \in \mathbb{R}^+$ the associated mask. $M(t) > 0$ means that clouds or shadows are detected. Figure 1.7 shows a true color satellite image (left image) and the same image where noisy data are masked (right image) thanks to the associated mask. All product's levels for Sentinel-2 are presented and discussed in [80]. These pre-processing steps are based on the MACCS (Multi-sensor Atmospheric Correction and Cloud Screening) / MAJA (MACCS-ATCOR Joint Algorithm) a processing chain [81] from CNES (French space agency), CESBIO (National center for space and biosphere studies) and the German Aerospace Center (DLR, [8]).

Finally, some data are in the edge of a tile or, for some reason, the data are not exploitable. In that case there is no data. The data provider indicates it using "reflectance" value -10000. No data "values" are automatically removed from the time-series.

[2] All products are available in the Theia Land Data Center: `http://www.theia-land.fr/en/presentation/products`.



(a) True color satellite image (b) Masked image

Figure 1.7: True color noisy image (left side) with clouds, shadows and saturations. The left image presents the data detected as noisy by [79]. Most of the noise has been captured, however some clouds on the top have not been detected. Images were downloaded from Theia Land Data Center (`https://www.theia-land.fr/en/products/`).

1.1.2 Land use or land cover maps

The construction of LULC maps, especially at large scale (country scale or larger), is often computationally expensive and the reference data are partially mislabelled. Mainly two reasons are identified, firstly the labels change over the years (constructions of urban areas, clear-cutting, *etc*) and yields to a heavy and expensive update at large scale. Secondly the change of ground resolution between two satellites missions imply a re-labelling of reference data. For example Formosat SITS are well-known for LULC maps and have now corrected the reference data throughout their use but has a different ground resolution to be used directly with Sentinel-2 SITS. The following focuses on the construction of the sets of Sentinel-2 reference data.

SITS reference data

The classes in this manuscript are extracted from [93, Section 4.2] and grouped by polygons. Each polygon refers to a unique class designated by CORINE Land Cover (CLC) dataset [20], French National Geographic Institute (IGN) maps [125] and other more specialized providers (grasslands and glaciers). The classification problem aims to separate around twenty classes. In a tile-based approach, the number of classes to predict may change from one tile to another, for example Glaciers polygons only appear in high altitudes. An example of resulting polygons is reported in Figure 1.8 (Other classes are presented in Figure 4.7).

As the authors mentioned in [93], they took into account multiple reference datasets to obtain large-scale maps but the proportion of mislabeled data is still significant. To overcome this issue, the authors used a Random Forests classifier (discussed later) known as robust for mislabeled data.

The diversity of land use or land cover maps

The production of Land-Use or Land-Cover (LULC) maps in the literature is wide. Depending on the context, the data source, the geographical area and the land cover may vary.

From [36, 73, 192], global land cover maps of high resolution (Landsat 30m ground resolution images) are discussed and reached classification scores from 64% to 70% for a total of 9 to 10 classes including agricultural areas or artificial areas among others. Only recently, [96] presented a LULC map at 10m using Sentinel-2 images with 85% classification score for 11 classes. The LULC has also been studied for smaller maps as continents or countries. [69] presented the case of South America with 5 classes with an overall accuracy of 89%; [89] for United State of America from 2001 to 2016 with 16 classes and 80% to 83% of estimated accuracy. France Metropolitan area has also been mapped [93] with 17 classes and about 90% accuracy score using Sentinel-2 10m ground resolution. China was mapped in [107] with 19 classes and about 75% accuracy score.

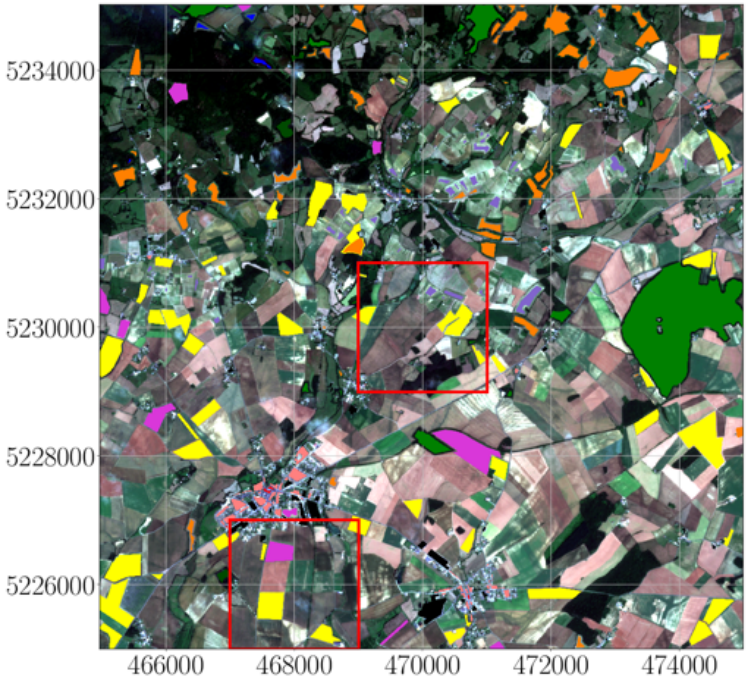


Figure 1.8: (Image from the supplementary materials of [41]). Polygons classes over a true colors tile in the center of the metropolitan France. Red squares are zoomed areas presented in [41].

The definition of the LULC problem will not be discussed further. Next Section only focuses on the classification techniques and comparison for a given LULC problem. This part highlights that a direct comparison between different studies is difficult.

1.2 Review of satellite image time-series classification

In the following E denotes the space where the SITS objects $\mathbf{y}$ live, *i.e.* $\mathbf{y} \in E$, and C is the total number of classes in the classification problem. In the following we consider non-noisy data, $\mathbf{y}$ can be represented by a vector of fixed length, and we consider supervised classification problems, each vector $\mathbf{y}$ is associated with a class membership. A common example of a vector input space is $E = \mathbb{R}^{pq}$ where pq is the dimension of the vector. q represents the number of temporal features within SITS and p the number of spectral bands. Other cases of E are considered in the next Chapter when reviewing state of the art in statistical modelling. The supervised framework implies that the models' parameters are learned from a training set including the class membership. The problem of supervised classification is recalled in Section 2.1.

1.2.1 Production of land use or land cover maps

As presented previously, SITS are complex for multiple reasons: they have spatio-spectro temporal information, they are noisy (optical satellite images) and they are produced in very large scale. The processing and classification tasks are challenging as the use of some classifiers is not possible at large scale or some additional assumptions must be done such as spatial or temporal independence, *etc*. A recent review can be found in [170] for different LULC purpose and scales. The "*conventional*" classifiers for LULC applications are presented, they were intensively used in the past decade and were issued from the Machine Learning community. Then more complex classifiers are presented, they are gaining in interest and are part of the deep neural network family.

Conventional classifiers

Among the conventional classifiers, this chapter will not review model-based classifiers as Gaussian Mixture models [102]. These models are described in-depth in the Chapter 2. Two major classifiers are reviewed: Random

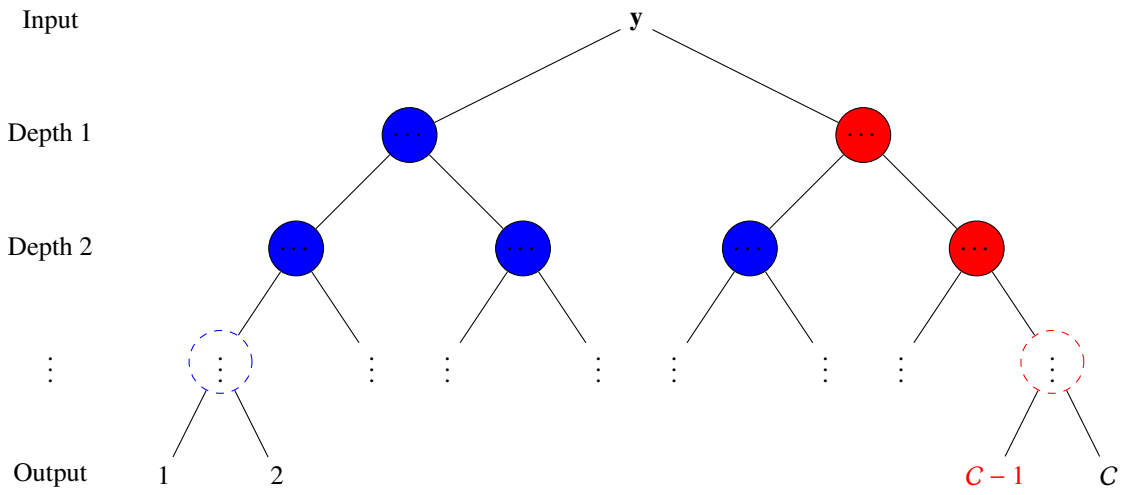


Figure 1.9: Schematic view of a decision tree which returns as output a class membership for a given input $\mathbf{y} \in E$. The class membership associated with $\mathbf{y}$ is given by the red path ($c = C - 1$) in this example.

Forests (RF) and Support Vector Machines (SVM).

Random forests (RF, [26]) are probably the most known classifier in the remote sensing literature. RF is a tree-based methods, it is a collection of numerous decision trees. A decision tree is a binary tree which can take as input a finite-dimensional vector, $\mathbf{y} \in E$. At each node, a decision rule gives a direction to the decision (left or right) and results into a set of labelled nodes by $c \in \{1, \dots, C\}$, a schematic view is reported in Figure 1.9. The data is classified thanks to its final node, for example the red path in Figure 1.9 leads to class $c = C - 1$. We refer to [84, Chapter 9] for additional properties. Each decision tree within the RF classifier returns a class, the class membership returned by RF is the majority voted class from decision trees.

Considering a time-series observed at q time-stamps with p spectral bands and a spatial independence (as done in the most cases [103]), then the input space E is often a vector of size pq and $E = \mathbb{R}^{pq}$ where all the p time-series are stacked. In [93], the authors presented promising results on Landsat 8 spectra-temporal SITS before the launch of Sentinel-2. They trained multiple RF classifiers, one per tile over the France territory and one per climatic area. They used linear interpolation to re-sample time-series. Spatio-temporal information can also be exploited, in [66] they have shown good results on Sentinel-2 at regions level. Finally [181] used RF to produce crops maps over Europe on Sentinel-1 time-series.

In addition to the good classification results, [93] highlighted that RF is robust when some target classes within the training set are wrong. RF is known to scale well w.r.t. the number of samples.

Support Vector Machines (SVM, [43]) were investigated for LULC applications before RF [160]. In binary classification ($C = 2$), it aims to find an optimal hyperplane to separate the two classes, see Figure 1.10. The hyperplane is optimal if it maximizes the margin between the two classes and minimizes the classification error in the training set. From there, two versions of SVM are found in the literature, linear SVM with an optimum hyperplane within the data space and non-linear SVM which transforms the data space into a feature space where the data are more likely linearly separable.

In linear SVM, the hyperplane is linear w.r.t. the input $\mathbf{y} \in E$, it corresponds to:

$$\langle \boldsymbol{\omega}, \mathbf{y} \rangle_E + b = 0, \tag{1.1}$$

where $\{\boldsymbol{\omega}, b\}$ are parameters of the model. Non-linear SVM can be defined by replacing the dot product in (1.1) by a kernel function (*kernel trick*), we refer to Section 2.2.1 for more details on kernels or to [72] for kernel methods in remote sensing. However, despite higher abilities to separate classes (polynomial or RBF hyperplanes among others, see Section 2.2.1), it is harder to scale to large data-sets. Indeed, only linear SVM are considered as "light". Finally the problem is extended to multi-class ($C > 2$) using the so-called *one-vs-all* or *one-vs-one* strategies. The latter one is often preferred [90].

Multiple comparisons between SVM and RF have been done on LULC applications. [91] compared SVM with decision trees and two other classifiers (model-based and Neural network based approaches) on a 6 classe problem. [173] provided a more recent comparison between SVM, RF and k-Nearest Neighbors classifiers on Sentinel-2 images. However, despite good results, the choice of non-linear SVM with more complex kernels

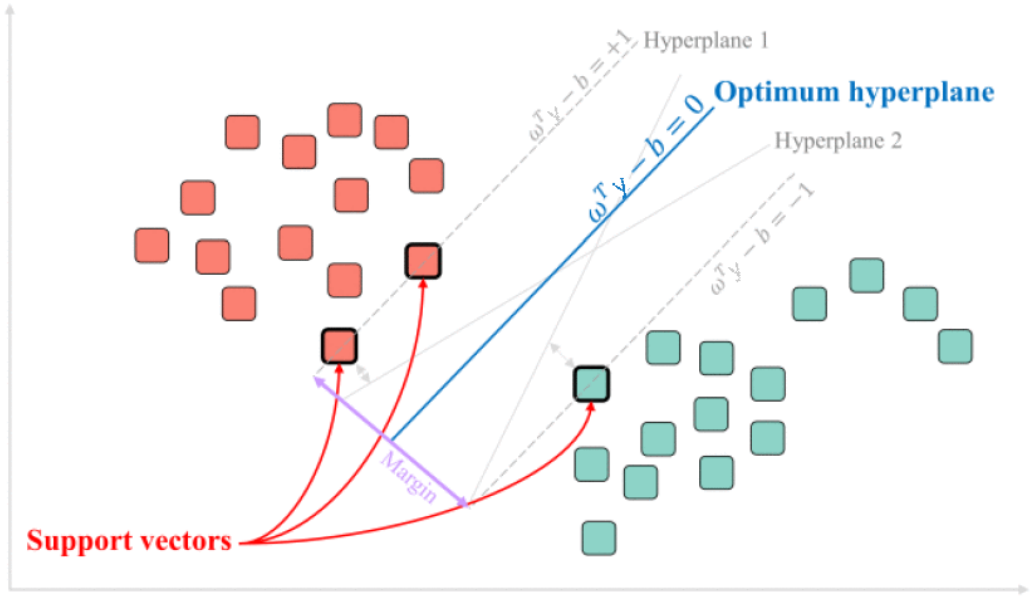


Figure 1.10: (Image from [160]). Two classes (red and green) are separated thanks to the optimal hyperplane (blue line), *i.e.* the hyperplane which maximizes the margin between the two groups. In this example the data $\mathbf{y}$ are real-input vectors.

scales quadraticly or cubicly with respect to the number of samples within the training set an large data-set remains a problem.

Applications in LULC are also wide, in addition to the above comparisons: [124] used SVM on spectral features for crop classification and [32] used time-series for data change detection. The use of both SVM and RF techniques is still very important. As presented in [160], RF is becoming more and more important than SVM in the remote sensing literature over the last four years. Both methods are still of interest as the number of parameters is reasonable to reach high accuracy on the produced map and the interpretability of some of the model parameters can be done. The following focuses on more complex techniques with a larger number of trainable parameters.

Artificial neural network classifiers

Artificial deep neural networks are widespread in LULC applications on smaller area as they have millions of parameters to learn. The number of parameters induces naturally a huge amount of computation time and resources.

A Neural Network (NN) is a structure based on the multi-layer perceptron (MLP) from Rosenblatt [151]. A perceptron is a parametric function with parameters $\{\boldsymbol{\omega}, b\}$, $\boldsymbol{\omega} \in E$ and $b \in \mathbb{R}$ that has been transformed through the Heaviside step function $h : \mathbb{R} \mapsto \{0, 1\}$ as in (1.2). A perceptron is also known as an *artificial neuron* as the output is activated (Output of h equal to 1) or inactivated otherwise:

$$z = h\left(\langle \boldsymbol{\omega}, \mathbf{y} \rangle_E + b\right). \tag{1.2}$$

The MLP has also been studied in remote sensing as in [91] which compared RF/SVM with the MLP structure. However, the advent of new kind of layers within neural networks (derived from MLP) and the big data era brought the intensive study of Deep Neural Network (DNN) with a large number of intermediate layers, also called *hidden* layers, over the past ten years. Every layer, including *hidden* layers, are composed by weights with different shapes and each layer is terminated by a non-linear function (more general than heaviside step function, like relu, elu, *etc*). The dimensions of input, output and hidden layers and the parameters (or weights) assigned to each layer with the associated activation function, together, is called a network *architecture*. A DNN architecture with a large number of hidden layers easily reaches millions of parameters. Inference of DNN is complex [84, Section 11.5] and done by back propagation of the gradient, issues related to a large number of parameters arise but also over-fitting problems which have to be taken into account within the training of such network (dropout techniques, data augmentation, *etc*).

In the context of remote sensing, mainly two architecture's families of DNN are studied in the literature: the first family is composed by *convolutional layers* and is known as Convolutional NN (CNN), they are specialized in

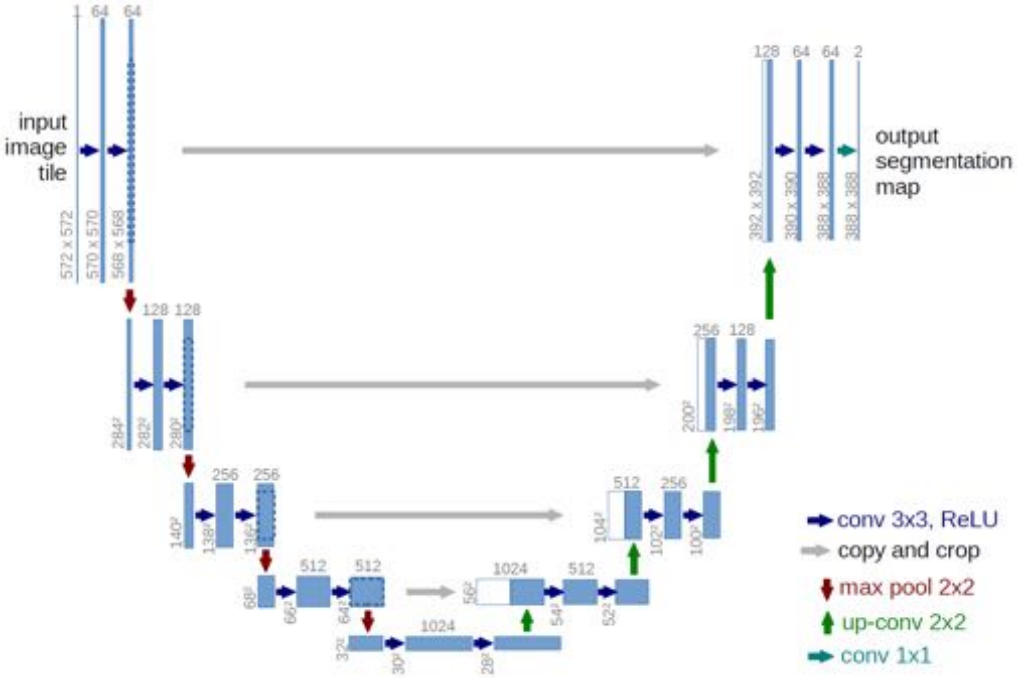


Figure 1.11: (Image from [150]). U-net architecture for pixel-wise classification.

image analysis (or spatial features). The second family is composed by *recurrent* layers and is known as Recurrent NN (RNN). They are specialized in time-series (temporal features).

A Convolution layer is a convolution product between a so-called *convolutional filter* and a tensor input. From there two structures are derived from the CNN, the first architecture takes as input an image (real-input matrix, or higher order tensor) and has C outputs (real-valued vector) where C is the number of classes in the classification problem. The number of parameters is correlated with the size of the convolutional filter and the complexity decreases significantly compared to a fully-connected network (for all hidden layers, each output is a linear combination of all inputs). The convolutions are mostly used for application to images. In remote sensing they are used to exploit spatial information of satellite images. For example [168] uses CNN to produce maps from Sentinel-2 images by upsampling the final layer to the size of the image. An other architecture derived from CNN is the auto-encoder architecture. The auto-encoder shares some dimensions of the output with the input of the network. The most known architecture is the U-net [150], it has the spatial dimensions shared between inputs and outputs. U-net is used for pixel-wise classification as illustrated in Figure 1.11 (outputs are tensors of order 3, the last dimension can be increased for different classification tasks. An application to aerial images is presented in [99].

The second family, Recurrent Neural Network (RNN), uses the temporal features of SITS. This family contains LSTM (Long-Short Term Memory, [87]) networks which are interesting for multiple views of the same scene at different times to predict, in this context, pixel-wise class membership. It has been used in [144, 152] on Sentinel-2 images and shows promising results.

Some recent works also focus on combining recurrent networks with convolutions for images [65, 138] on Sentinel-2 images. For two in-depth reviews, we refer to [193] for deep NN models applied to high spatial resolution satellite images and to [5] for application to hyper-spectral images. At this time, [96] presents a large scale map with 10m ground resolution but without temporal information from SITS. When dealing with large data set at a country scale with large time domain (one year of images) at high resolution, the conventional machine learning techniques are preferred when DNN architecture reached millions and millions of parameters to train.

Conventional and neural network classifiers, take as input a vector of fixed length, a situation that can be hardly reached with SITS. The next Section discusses on problems induced by missing data.

1.2.2 Classification with missing data

When dealing with missing data, the previous classifiers are not able to perform without re-sampling to a finite and fixed grid of time stamps. This grid must be the same within the training and the validation sets of data. For Sentinel-2 spatio- spectro- temporal images, the reconstruction of missing data to a fixed temporal grid may differ

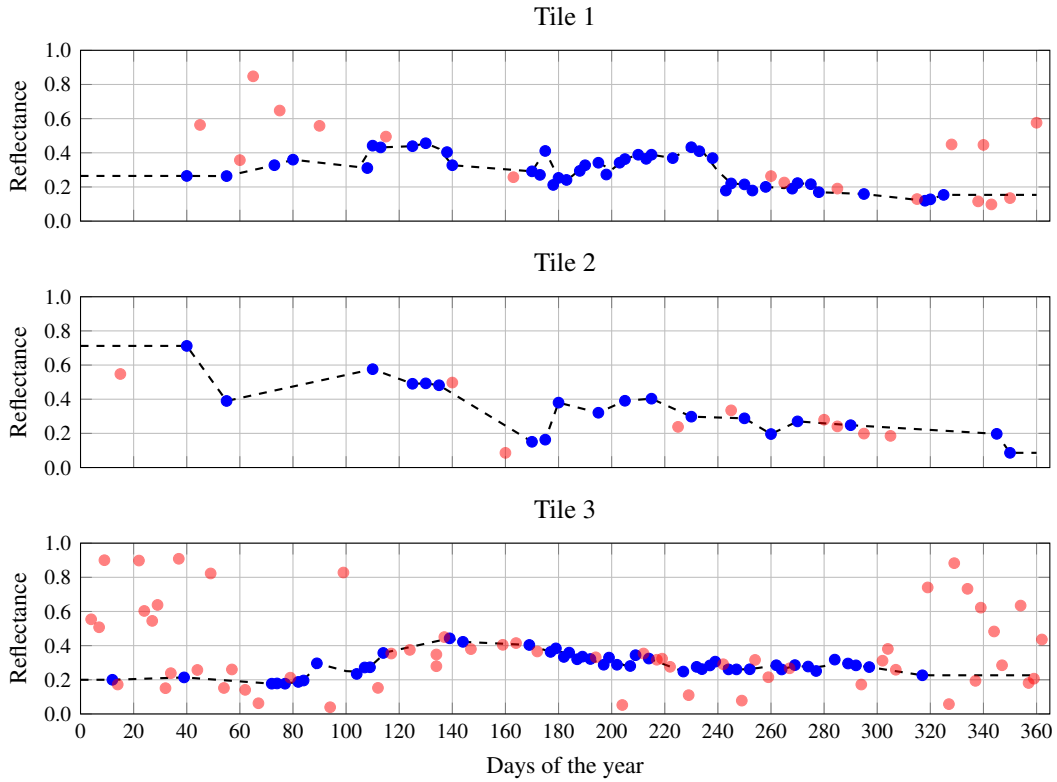


Figure 1.12: Sentinel-2 pixel time series for the near infra-red band (B8) for three different tiles over the French territory. The clear dates are marked as blue whereas red marks indicate a detection of noise from the mask. Dashed lines represent the gap-filled data.

for different studies. We refer to [158] for a review of these techniques and how to combine them to reconstruct missing data. Before going further, let us recall that S2 SITS are processed at level 2A (Section 1.1.1) and are brought with a mask with noise detection. As the 60m ground resolution are used to process the data from 1A to 2A, the 60m spectral bands will not be used in this manuscript.

Spatial re-sampling

Let us recall that the spatial location of each pixel has not been taken into account in this manuscript, the time-series will be classified thanks to their spectral and temporal signature (using non-noisy time stamps). We refer to [158, 195] for missing data reconstruction using spatial information.

The process will only concern 20m ground resolution spectral bands (Table 1.1). A simple up-sampling from 20m to 10m ground resolution will be done. It uses the "bicubic" algorithm from orfeo toolbox [171], in particular *superimpose*; this toolbox is an open-source project for remote-sensing applications.

Temporal re-sampling

Sentinel-2 SITS have different temporal samplings for each time series. The reasons for this two-fold: firstly the Sentinel-2 tiles have different time stamps from one tile to another (provided by the data provider) thanks to the orbital path of the satellites, an example is provided in Figure 1.12 for three different tiles by the blue and red dots. The red dots correspond to masked data (noisy acquisitions). Secondly the noise sources as clouds or shadows appear locally, *i.e.* the clear data (or noise-free data) from one time-series to another appear at different times in a year. To overcome these issues, [93] defined a unique temporal grid over a complete territory (metropolitan France in their study). The authors removed noisy acquisition thanks to the mask values (red dots in Figure 1.12) and linearly re-sampled time-series to the new fixed grid. This technique, simply mentioned as *gap-filling*, is implemented in orfeo toolbox [171]. It consists in taking the linear interpolation between two clear dates, an illustration is provided in Figure 1.12 where dashed lines represent the gap-filled time-series.

Other techniques are available but are not used to process Sentinel-2 SITS. In Chapter 5 we review statistical methods for missing data reconstruction.

1.3 Challenges of satellite image time-series classification

This chapter presented the Satellite image time-series and the Sentinel-2 constellation which operates to produce high resolution images. We finished the presentation of Sentinel-2 SITS with the emphasis on mislabelled data sets, particularly in large scale areas. These data will be used throughout this manuscript in order to assess the performances of the proposed classifiers thanks to the spectro-temporal aspect of the data (a spatial independence assumption will be done). We also presented the state-of-the-art classifiers with finite-sized input vectors, with promising classification scores and working on national scale to produce Land Use or Land Cover (LULC) maps. Application to Sentinel-2 SITS requires an additional pre-processing step to re-sample time-series to a fixed set of time stamps. We illustrated this processing step on one time-series using the linear gap-filling technique which is usually done to process national LULC map [93].

The main challenges are the classification of these SITS using classifiers that scales to large data-sets and to remove the processing step of re-sampling the time-series to a known set of time-stamps.

Recent statistical approaches are of interest thanks to their interpretability (unlike more complex classifiers as Neural Networks) and the wide variety of applications. An example is the use of the Gaussian processes [31] for the analysis of the three dimensions of SITS. Supervised model-based classification is, more generally, already used in time-series classification. The following Chapter reviews statistical modelling for classification of multi-dimensional time-series.

Chapter 2

Statistical modelling for time-series classification

Outline

French introduction

Ce chapitre présente la classification de séries temporelles par modélisation statistique. La modélisation statistique peut être séparée en deux grande catégories : l'apprentissage non supervisé et l'apprentissage supervisé. Ce chapitre se concentre plus particulièrement sur ce dernier.

La première partie présente la classification supervisée. Pour cela la règle de décision (assignation d'un élément à une classe) est présentée pour différents types d'objets à classer : des vecteurs, des matrices ou bien des objets plus complexes (données textuelles, *etc*). Différents modèles génératifs sont alors présentés, en particulier les modèles supposant une distribution gaussienne. Plus particulièrement la loi gaussienne multivariée est étudiée puis étendue au cas matriciel.

La seconde partie présente la statistique fonctionnelle en détaillant les processus gaussiens (Gaussian Processes, GP). Deux cas sont abordés : l'utilisation des processus gaussiens dans le cadre de problèmes de régression et de classification. La régression repose sur le principe que les échantillons sont des observations d'une fonction unique puis, par la loi des probabilités conditionnelles pour des vecteurs gaussiens, permet la prédiction de valeurs à des instants non observés. La classification repose sur l'hypothèse que la probabilité d'appartenance à une classe est composée d'une fonction à valeurs dans $[0, 1]$ et d'une fonction latente modélisée par un GP. Le cas de la classification binaire est présenté puis étendu au cas multi-classes. Enfin, cette partie se termine par un bref état de l'art sur les GP multi-dimensionnels, *i.e.* des fonctions aléatoires à valeurs dans $\mathbb{R}^p$ avec $p > 1$. Ces méthodes de classification ont, pour l'essentiel, une complexité cubique par rapport au nombre d'échantillons. Ces notions sont nécessaires pour la compréhension des contributions.

Ces méthodes ne permettent pas, à notre connaissance, de classer les séries temporelles à échantillonnage irrégulier, notamment les SITS Sentinel-2. Ces méthodes reposent sur des objets de dimensions finie et fixe, elles impliquent un ré-échantillonnage temporel des SITS, ce qui n'est pas compatible avec notre problématique.

Let E be an arbitrary space containing all objects to classify (vectors of dimension p, time-series, *etc*) and let $\{1, \ldots, C\}$ be a set of C classes. Let $\mathcal{S} = \{(\mathbf{y}_i, z_i)\}_{i=1}^n$ be a set of n independent and identically distributed (*i.i.d.*) samples, or realizations, from a random pair $(\mathbf{Y}, z) \in E \times \{1, \ldots, C\}$. For any given class $c \in \{1, \ldots, C\}$, the model-based classification problem is to assume that the conditional distribution of $\mathbf{y}|z = c$, $\mathbf{y} \in E$, belongs to some parametric family. For any sample i, $\mathbf{y}_i|z_i = c$ means that $\mathbf{y}_i$ belongs to the class c with associated density $p(\mathbf{y}_i|z_i = c)$. This approach differs from the previous approaches where the data mechanism is not modelled by a statistical distribution [27] as presented in Chapter 1. The statistical approaches are interesting for their interpretability and, often, reduced numerical complexity thanks to a reduced number of model parameters.

In this chapter, we focus on statistical modelling by describing supervised model-based classification at first in Section 2.1 with a focus on Gaussian distributions. Afterwards an overview of Gaussian processes for regression and classification is given in Section 2.2. Finally a brief conclusion is provided in Section 2.3.

2.1 Supervised model-based classification

Model-based classification is based on the definition of a *decision rule* δ which returns a class c considered as the best candidate from an observation $\mathbf{y} \in E$. We say that $\hat{c}$ is the class membership of $\mathbf{y}$:

$$\begin{aligned} \delta \ : \ E \ &\rightarrow \ \{1, \dots, C\}, \\ \mathbf{y} \ &\mapsto \ \hat{c}. \end{aligned}$$

In a supervised framework, all classes are known and represented in the set of data. The probabilistic model is inferred on the complete set $\{(\mathbf{y}_i, z_i)\}_{i=1}^n$. On the opposite, an unsupervised framework supposes that the classes are unknown and are inferred from the data inputs $\{\mathbf{y}_i\}_{i=1}^n$.

Firstly the decision rule is presented Section 2.1.1, then the discrimination problem with a Gaussian assumption is presented in Section 2.1.2 and finally extensions to matrix-variate Gaussian distributions are presented in Section 2.1.3.

2.1.1 Decision theory and supervised framework

***Maximum a posteriori* rule**

In the Bayesian decision theory, the optimal decision rule is called the *maximum a posteriori* (MAP) rule. The Bayes rule gives us the *a posteriori* probability by:

$$\mathbb{P}(z = c|\mathbf{y}) = \frac{\pi_c p(\mathbf{y}|z = c)}{p(\mathbf{y})} \propto \pi_c p(\mathbf{y}|z = c),$$

where $p(\mathbf{y}|z = c)$ is the *likelihood* density and $\pi_c = \mathbb{P}(z = c)$ the *a priori* probability on class c. $p(\mathbf{y})$ is the marginal distribution of $\mathbf{y}$ which can be written as a finite mixture:

$$p(\mathbf{y}) = \sum_{c=1}^{C} \pi_c p(\mathbf{y}|Z = c).$$

The MAP rule assigns to a non-labeled object the class $\hat{c}$ such as:

$$\delta(\mathbf{y}) = \hat{c} = \arg\max_c \mathbb{P}(z = c|\mathbf{y}) = \arg\max_c \pi_c p(\mathbf{y}|z = c). \tag{2.1}$$

From (2.1), two approaches are found in the literature: the discriminative approach models the *a posteriori* probability and the generative approach which models the *a priori* probability and the *likelihood*. The latter approach is the one considered in the following.

In a supervised framework, π_c is usually estimated by its empirical counterpart and the generative approach is to model the *likelihood* $p(\mathbf{y}|z = c)$. A model is a distribution which can be parametrized using a finite number of parameters denoted by $\boldsymbol{\theta}$. If we suppose that the problem is decoupled w.r.t. each class, we define C models where each model in class $c \in \{1, \dots, C\}$ is parametrized by its own parameters $\boldsymbol{\theta}_c$. We write the parametrized *likelihood* density in class c by $p(\mathbf{y}|z = c; \boldsymbol{\theta}_c)$. In the following we introduce the subset $\mathcal{S}_c$ of $\mathcal{S}$ such that $\mathcal{S}_c = \{(\mathbf{y}_i, z_i)\}_{i|z_i=c}$. We write $n_c = |\mathcal{S}_c|$, then $n_c \leq n$ and $\sum_{c=1}^{C} n_c = n$.

Remark. Let us highlight that the optimization problem (2.1) is equivalent to:

$$\hat{c} = \arg\min_c k \log\left(\pi_c p(\mathbf{y}|z = c; \boldsymbol{\theta}_c)\right),$$

where k is a non-zero negative constant ($k = -2$ for Gaussian distributions as log is strictly increasing on $\mathbb{R}^+$.

Maximum likelihood estimation

Maximum likelihood estimator (or MLE) is an estimator which maximizes the *likelihood* function w.r.t. parameters $\boldsymbol{\theta}$. Optimal parameters are denoted by $\hat{\boldsymbol{\theta}}_{MLE}$ and defined by:

$$\hat{\boldsymbol{\theta}}_{MLE} = \arg\max_{\boldsymbol{\theta}} L(\boldsymbol{\theta}, \mathbf{y}). \tag{2.2}$$

$L(\boldsymbol{\theta}, \mathbf{y}) = \prod_{c=1}^{C} p(\mathbf{y}|z = c, \boldsymbol{\theta}_c)$ is the complete likelihood function of the model with parameters $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_C\}$ on data $\mathbf{y} \in E$. As all samples from $\mathcal{S}$ are independent and identically distributed, the complete likelihood is the product of density $p(\mathbf{y}_i|z_i = c; \boldsymbol{\theta}_c)$, also called *marginals*. Additionally, when the problem is decoupled w.r.t. the class, then (2.2) is computed for each class, *i.e.* for each sample in subset $\mathcal{S}_c$:

$$\hat{\boldsymbol{\theta}}_{c,MLE} = \arg\max_{\boldsymbol{\theta}_c} \prod_{i|z_i=c} p(\mathbf{y} = \mathbf{y}_i|z_i = c; \boldsymbol{\theta}_c). \tag{2.3}$$

For the sake of clarity, the realization $\mathbf{y} = \mathbf{y}_i$ given $z_i = c$ will be directly written $\mathbf{y}_i|z_i = c$. Generally, π_c is a parameter of the model and is encompassed in $\boldsymbol{\theta}_c$, for example in unsupervised classification they have to be estimated through the EM algorithm (Expectation Maximization) [22, Eq. (2.10), p.25]. In our supervised framework, π_c are estimated by their empirical counterpart: $\hat{\pi}_c = n_c/n$ and are removed from parameters $\boldsymbol{\theta}_c$.

MLE has interesting properties like convergence (when $n \to \infty$) to a normal distribution centered on the true value of $\boldsymbol{\theta}$ [84, Chapter 8, p.266].

Probabilistic models and distributions

As mentioned previously, the data $\mathbf{y}$ live in an arbitrary space E. In the following we discuss different examples of space E and associated distributions found in the literature.

Let q be the number of features of $\mathbf{y}$. If $E = \mathbb{R}^q$ then the samples are real-valued vectors, for example regularly sampled time-series. In this case, the multivariate (q-variate) Gaussian distribution is often adopted, leading to the well-known linear and quadratic discriminant analysis classifiers (see Section 2.1.2). The probability density function (pdf) of the multivariate normal distribution is given in (2.4).

Recent works focus on non-Gaussian distributions. One example is the skew-normal distribution which has been introduced in the past century [6]. Considering the one-dimensional case ($E = \mathbb{R}$), the pdf of the skew-normal distribution, denoted by p_{s-n}, is defined with a parameter γ which controls the skewness ($\gamma = 0$ yields the standard Gaussian distribution):

$$p_{\text{s-n}}(y) = 2p(y)\Phi(\gamma y),$$

where $p(.)$ denotes the standard normal pdf and $\Phi(.)$ is the standard Gaussian cumulative density function ($\Phi(y) = \int_{-\infty}^{y} p(t)dt$). This distribution is used to deal with asymmetric data [35, 175], the multivariate skew-normal distribution is described in [7].

An other example is the t-distribution with heavier tails. The distribution depends on a shape parameter ζ ($\zeta \to \infty$ yields the standard Gaussian distribution), the multivariate t-distribution pdf is given by:

$$p(\mathbf{y}) = \frac{\Gamma\left((\zeta + q)/2\right)}{\Gamma(\zeta/2)(\zeta\pi)^{q/2}|\boldsymbol{\Sigma}|^{1/2}} \left(1 + \frac{1}{\zeta}(\mathbf{y} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{y} - \boldsymbol{\mu})\right)^{-(\zeta+q)/2}, \qquad \mathbf{y} \in \mathbb{R}^q,$$

where Γ is the gamma function and $|\boldsymbol{\Sigma}|$ is the determinant of $\boldsymbol{\Sigma}$. This distribution is used to deal with outliers [4, 132]. The distributions are illustrated in Figure 2.1. It presents a right skewed pdf ($\gamma > 0$) and a t-distribution with heavy tails ($\zeta = 1$) with comparison to the standard Gaussian pdf. More details about these two parameters are provided in [22, Chapter 9]. When $\mathbf{y}$ are categorical data, for example nominal data as species analysis [55] (presence or absence of species in an environment), the space E is discrete. In that context, some studies focus on multinomial distributions [34] or Dirichlet distributions [21]. If E is ordered, then dedicated distributions are proposed as in [15] with ordinal data. Other data-sets, or spaces E, are presented in [22, Chapter 6].

Lastly, when E is finite and high dimensional or infinite dimensional (*i.e.* when $E = \mathbb{R}^q$ and $q \to \infty$), specific parametric and non-parametric models are preferably used. Section 2.1.2 presents the High Dimensional Discriminant Analysis (HDDA) and Section 2.2.1 discusses the kernel functions that can be used in non-linear classifiers for infinite dimensional data.

2.1.2 General discriminative problem

As introduced before, when $E = \mathbb{R}^q$ we can use linear and quadratic classifiers. The following presents the Linear and Quadratic Discriminant Analysis (known as LDA and QDA) with Gaussian assumption on the likelihood density.

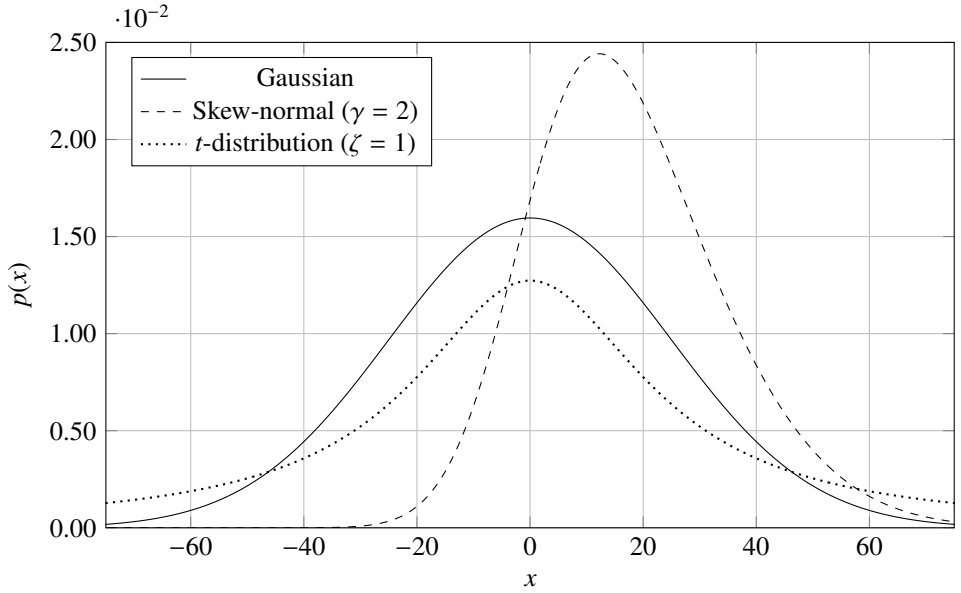


Figure 2.1: Probability density functions (p) for 1-dimensional Gaussian distribution (solid line), skew-normal distribution (dashed line) with $\gamma = 2$ and the t-distribution (dotted line) with $\zeta = 1$.

Linear and Quadratic Discriminant Analysis (LDA/QDA)

Let $\mathbf{y}|z = c \sim \mathcal{N}_q(\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$ be a multivariate Gaussian model with mean $\boldsymbol{\mu}_c$ and covariance matrix $\boldsymbol{\Sigma}_c$. The model parameters are $\boldsymbol{\theta}_c = \{\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c\}$. Then the associated q-variate Gaussian likelihood density (or pdf) for the i-th sample, or realization from the Gaussian, $\mathbf{y}_i|z_i = c$ is given by:

$$p(\mathbf{y}_i|z_i = c; \boldsymbol{\theta}_c) = (2\pi)^{-q/2}|\boldsymbol{\Sigma}_c|^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{y}_i - \boldsymbol{\mu}_c)^\top \boldsymbol{\Sigma}_c^{-1}(\mathbf{y}_i - \boldsymbol{\mu}_c)\right). \tag{2.4}$$

From (2.3) in supervised framework (recall that *a-priori* probabilities are estimated by their empirical counterpart) with Gaussian density (2.4), the minimization of the negative log-likelihood is:

$$\hat{c} = \arg\min_c \prod_{i|z_i=c} \ell(\mathbf{y}_i; \boldsymbol{\theta}_c).$$

with:

$$\ell(\mathbf{y}_i; \boldsymbol{\theta}_c) = -\log(p(\mathbf{y}_i; \boldsymbol{\theta}_c)) \propto D^2_{\boldsymbol{\Sigma}_c}(\mathbf{y}_i, \boldsymbol{\mu}_c) + \log|\boldsymbol{\Sigma}_c| + q\log(2\pi).$$

$D_{\boldsymbol{\Sigma}}(\mathbf{y}, \boldsymbol{\mu}) = \sqrt{(\mathbf{y} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\mathbf{y} - \boldsymbol{\mu})}$ stands for the Mahalanobis distance with covariance $\boldsymbol{\Sigma}$. The optimization problem minimizes both the quadratic term ($D_{\boldsymbol{\Sigma}}(\mathbf{y}, \boldsymbol{\mu})$) and the complexity ($|\boldsymbol{\Sigma}_c|$). Then the MAP rule $\delta(\mathbf{y}_i) = \arg\max_c \mathbb{P}(z_i = c|\mathbf{y}_i)$ returns the class with the highest probability, this method is known as the *Quadratic Discriminant Analysis* (or QDA).

If all covariance matrices $\boldsymbol{\Sigma}_c$, $c \in \{1, \ldots, C\}$, are assumed to be equal to $\boldsymbol{\Sigma}$, *i.e.* $\boldsymbol{\theta}_c = \{\boldsymbol{\mu}_c, \boldsymbol{\Sigma}\}$, then the covariance matrix is shared for all classes. This particular case is known as the *Linear Discriminant Analysis* (LDA) as explained below.

Decision boundaries

A decision boundary describes a delimitation between groups where the probabilities to belong to all classes are equal. Let us consider the case $C = 2$ with $E = \mathbb{R}^q$, then the decision boundary delineates two groups. Let $s(\mathbf{y})$ be the quantity defined by:

$$s(\mathbf{y}) = \log\left(\frac{\mathbb{P}(z = 2|\mathbf{y})}{\mathbb{P}(z = 1|\mathbf{y})}\right) = \log(p(\mathbf{y}|z = 2)) - \log(p(\mathbf{y}|z = 1)), \qquad \forall\, \mathbf{y} \in E. \tag{2.5}$$

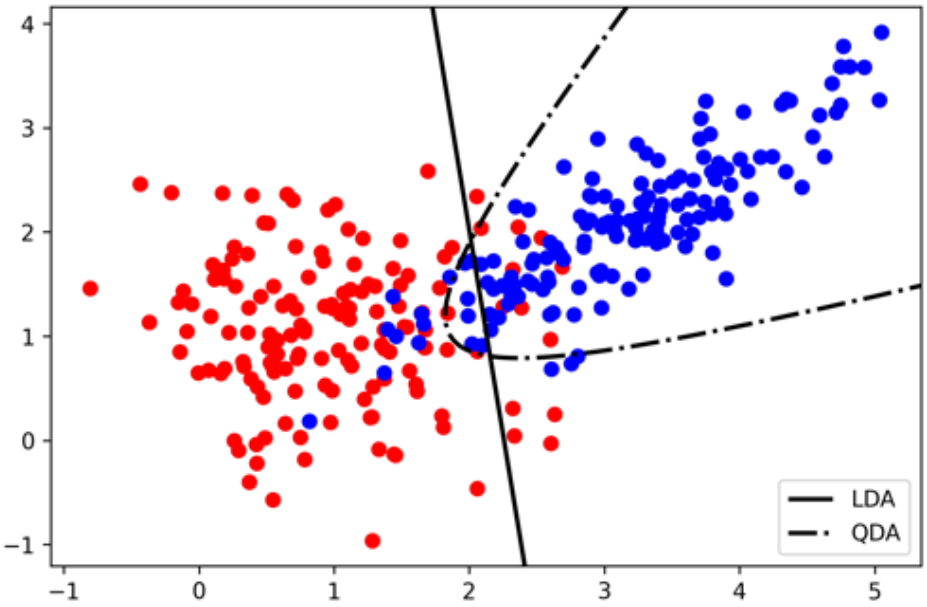


Figure 2.2: LDA and QDA decision boundaries between two classes ($c = 1$ in red and $c = 2$ in blue).

If the probabilities are equal, then $s(\mathbf{y}) = 0, \forall\, \mathbf{y} \in E = \mathbb{R}^q$. In particular, the space $\{\mathbf{y} \in \mathbb{R}^q | s(\mathbf{y}) = 0\}$ defines the decision boundary.

If the distribution is Gaussian, $s(\mathbf{y}) = 0$ is equivalent to:

$$\begin{aligned} D^2_{\boldsymbol{\Sigma}_2}(\mathbf{y}, \boldsymbol{\mu}_2) + \log|\boldsymbol{\Sigma}_2| - D^2_{\boldsymbol{\Sigma}_1}(\mathbf{y}, \boldsymbol{\mu}_1) + \log|\boldsymbol{\Sigma}_1| &= 0, \\ \Leftrightarrow (\mathbf{y} - \boldsymbol{\mu}_2)^\top \boldsymbol{\Sigma}_2^{-1}(\mathbf{y} - \boldsymbol{\mu}_2) - (\mathbf{y} - \boldsymbol{\mu}_1)^\top \boldsymbol{\Sigma}_1^{-1}(\mathbf{y} - \boldsymbol{\mu}_1) + \log\frac{|\boldsymbol{\Sigma}_2|}{|\boldsymbol{\Sigma}_1|} &= 0, \\ \Leftrightarrow \mathbf{y}^\top \boldsymbol{\Sigma}_2^{-1}\mathbf{y} - \mathbf{y}^\top \boldsymbol{\Sigma}_1^{-1}\mathbf{y} - 2\boldsymbol{\mu}_2^\top \boldsymbol{\Sigma}_2^{-1}\mathbf{y} + 2\boldsymbol{\mu}_1^\top \boldsymbol{\Sigma}_1^{-1}\mathbf{y} + \boldsymbol{\mu}_2^\top \boldsymbol{\Sigma}_2^{-1}\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1^\top \boldsymbol{\Sigma}_1^{-1}\boldsymbol{\mu}_1 + \log\frac{|\boldsymbol{\Sigma}_2|}{|\boldsymbol{\Sigma}_1|} &= 0. \end{aligned} \tag{2.6}$$

The boundary equation in (2.6) is quadratic w.r.t. $\mathbf{y}$, explaining why it refers to the *Quadratic* discriminant analysis. Taking $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}$ in (2.6) yields:

$$2(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^\top \boldsymbol{\Sigma}^{-1}\mathbf{y} + \boldsymbol{\mu}_2^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1^\top \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_1 = 0,$$

which is linear w.r.t. $\mathbf{y}$, referring to the *linear* discriminant analysis. Figure 2.2 illustrates the linear and quadratic decision boundaries between two classes (blue and red dots) for samples $\mathbf{y}_i \in E = \mathbb{R}^2, \forall i \in \{1, \ldots, n\}$.

Estimation of parameters

Recall that estimation of parameters is done by *maximum likelihood* in each class c independently. Inference in class c is done on the set $\mathcal{S}_c$ with cardinal n_c, the number of samples in class c. $\hat{\boldsymbol{\theta}}_{c,MLE}$ is solution of (2.3) and $\hat{\pi}_c = n_c/n$ where $n = \sum_c n_c$. In QDA, the optimal parameters are well known [84, Chapter 4] and are given by:

$$\begin{aligned} \hat{\boldsymbol{\mu}}_c &= \sum_{i|z_i=c} \mathbf{y}_i/n_c, \\ \hat{\boldsymbol{\Sigma}}_c &= \sum_{i|z_i=c} (\mathbf{y}_i - \hat{\boldsymbol{\mu}}_c)^\top (\mathbf{y}_i - \hat{\boldsymbol{\mu}}_c)/n_c. \end{aligned}$$

Penalized and regularized discriminant analysis

In the Gaussian discriminative analysis, some additional hypothesis are usually made to give the model more robustness to real world data-sets especially in high dimension. Most of them consist of modifying the estimator of the covariance matrix.

The penalized approach is done by adding *a-priori* information on the covariance matrix $\mathbf{\Sigma}$ as:

$$\tilde{\mathbf{\Sigma}} = \hat{\mathbf{\Sigma}} + \eta\mathbf{P}.$$

$\mathbf{P}$ is the penalizer which contains *a-priori* behavior and η is the strength of the penalization. It also increases the stability of convergence (in case of singularity of matrix $\hat{\mathbf{\Sigma}}$).

The regularized discriminant analysis (RDA) is more complex, it combines both LDA and QDA behavior. We refer to [63] for a complete description of the technique.

High dimensional data analysis

The problem of high-dimensional data arises when the cardinal of the set n is small compared to the dimension of the samples q. Although regularized discriminant analysis increases the stability, other methods are found in the literature as dimension reduction and parsimonious methods.

We just mention here the so-called High Dimensional Discriminant Analysis (HDDA) [24] which is part of the parsimonious models. It assumes that the covariance matrix $\mathbf{\Sigma}$ is approximated by:

$$\tilde{\mathbf{\Sigma}} = \mathbf{Q}\mathbf{\Delta}\mathbf{Q}^\top,$$

where $\mathbf{Q}$ is the matrix of eigenvectors from $\hat{\mathbf{\Sigma}}$ and more specifically, $\mathbf{\Delta}$ is the diagonal matrix which contains eigenvalues with some simplification. An example is a diagonal matrix which contains only two eigenvalues (λ_0 and λ_1): $\mathbf{\Delta} = \text{diag}(\lambda_0, \ldots, \lambda_0, \lambda_1, \ldots, \lambda_1)$ with the same size as $\mathbf{\Sigma}$. We refer to [22, Chapter 8] for a wide overview of the dimensionality problem and applications.

The dimensionality issue in this manuscript does not refer to high dimensional data but to large data-sets (*i.e.* n large enough to encounter numerical issues as the processing of Terabytes of data in satellite images).

2.1.3 Matrix-variate Gaussian distribution

Previously, we illustrated distributions and examples on vectors ($E = \mathbb{R}^q$). In the context of satellite image time-series, both spectral, temporal and spatial information are provided from the satellites and it is interesting to combine them in order to increase the precision of the analysis.

In the following we decide to illustrate the case where two dimensions are used jointly. Let p be the number of spectral bands and q the number of observations (time-stamps), then the observation $\mathbf{Y}$ is a $p \times q$ real matrix, we write $\mathbf{Y} \in E$ with $E = \mathcal{M}_{p,q}(\mathbb{R})$. $\mathcal{M}_{p,q}(\mathbb{R})$ denotes the set of real matrices sized $p \times q$. One may stack all vectors (see Definition 2.1) in order to retrieve the vector case (with $E = \mathbb{R}^{pq}$) but the dimensions could become too large and numerical issues may appear.

The matrix-variate normal distribution has been used in different contexts, for example in electro-encephalography [74, 165] where the study combines time samples and sensors (or spatial and spectral parts [198]), remote sensing [71], where the authors combine time samples and spectral information, or dimension reduction [76] with time samples and spatial locations. [121] shows that it may be extended to higher order tensor distributions.

Definitions and properties

For the sake of self-containedness, the *vec* operator and the *Kronecker product* are defined (Definitions 2.1 and 2.2). Then the matrix-variate Gaussian distribution is formerly introduced in Definition 2.3.

Definition 2.1 (*vec* operator)**.** Let $\mathbf{C}$ a $m \times n$ matrix. $\text{vec}(\mathbf{C})$ is a vector of size mn such that:

$$\text{vec}(\mathbf{C}) = (c_{11}, \ldots, c_{m1}, c_{12}, \ldots, c_{m2}, \ldots, \ldots, c_{1n}, \ldots, c_{mn}).$$

Definition 2.2 (Kronecker product)**.** Let $\mathbf{C}$ a $m_1 \times n_1$ matrix and $\mathbf{D}$ an other matrix of size $m_2 \times n_2$. The left Kronecker product, or **Kronecker product**, $\mathbf{C} \otimes \mathbf{D}$ is a matrix of size $m_1 m_2 \times n_1 n_2$ such that:

$$\mathbf{C} \otimes \mathbf{D} = \begin{pmatrix} c_{11}\mathbf{D} & \ldots & c_{1n}\mathbf{D} \\ \vdots & \ddots & \vdots \\ c_{m1}\mathbf{D} & \ldots & c_{mn}\mathbf{D} \end{pmatrix}. \tag{2.7}$$

Remark. Let us highlight that, from (2.7), the matrix $\mathbf{C}\otimes\mathbf{D}$ does not allow us to retrieve $\mathbf{C}$ and $\mathbf{D}$ uniquely. Indeed, let $\eta > 0$, then $\mathbf{C}\otimes\mathbf{D} = (\eta)\mathbf{C}\otimes(1/\eta)\mathbf{D}$. With higher-order tensor distribution as in [121], the identifiability issue is increasing.
Besides, we refer to [155] for several properties of these two operators.

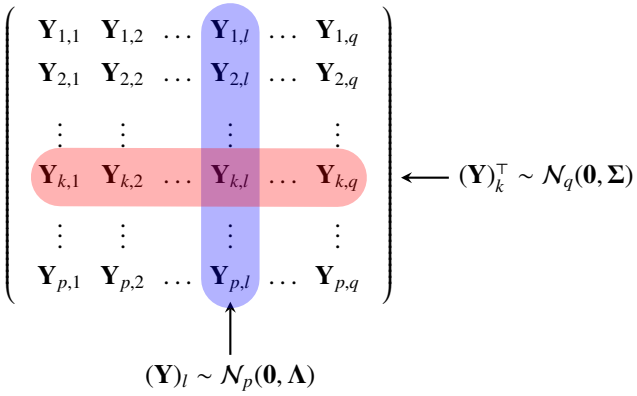


Figure 2.3: Separability between lines and columns for a centered random matrix $\mathbf{Y}$. The k-th line of $\mathbf{Y}$ does not depend on covariance matrix $\mathbf{\Lambda}$ and the l-th line does not depend on $\mathbf{\Sigma}$.

From [48], a popular approach to reduce the dimension of the problem is to consider uncorrelated lines and columns of $\mathbf{Y} \in \mathcal{M}_{p,q}(\mathbb{R})$. This is the hypothesis of **separability** between lines and columns, illustrated in Figure 2.3. With a Gaussian assumption [48, 166], it implies the definition of two covariance matrices: $\mathbf{\Sigma}$ of size $q\times q$ which defines the covariance between two columns of $\mathbf{Y}$ and $\mathbf{\Lambda}$ of size $p\times p$ between two lines. This illustration does not show any mean, however it is easy to shift the matrix using a mean matrix $\mathbf{M}$.

Definition 2.3 (Matrix-variate Gaussian distribution, [48])**.** Let $\mathbf{Y}$ be a random matrix of size $p\times q$. $\mathbf{Y}$ follows a matrix-variate Gaussian distribution, denoted by $\mathcal{MN}_{p,q}(\mathbf{M},\mathbf{\Sigma},\mathbf{\Lambda})$ with mean $\mathbf{M}$ and covariance matrices $\mathbf{\Lambda}$ (sized $p\times p$) between lines and $\mathbf{\Sigma}$ (sized $q\times q$) between columns if and only if $\mathbf{y} = \text{vec}(\mathbf{Y})$, the vector of size pq, follows a pq-variate Gaussian distribution with mean $\boldsymbol{\mu} = \text{vec}(\mathbf{M})$ and covariance matrix $\mathbf{\Sigma}\otimes\mathbf{\Lambda}$. We write:

$$\mathbf{Y} \sim \mathcal{MN}_{p,q}(\mathbf{M},\mathbf{\Sigma},\mathbf{\Lambda}) \text{ if and only if } \mathbf{y} \sim \mathcal{N}_{pq}(\boldsymbol{\mu},\mathbf{\Sigma}\otimes\mathbf{\Lambda}).$$

Following this definition, since $\mathbf{y}$ has its associated multivariate Gaussian probability density function (pdf), it is possible to define a pdf on matrix-variate Gaussian random variable $\mathbf{Y}$.

Proposition 2.1. *Let* $\mathbf{Y} \sim \mathcal{MN}_{p,q}(\mathbf{M},\mathbf{\Sigma},\mathbf{\Lambda})$*, the probability density function of* $\mathbf{Y}$ *is:*

$$p(\mathbf{Y}) = (2\pi)^{-pq/2}|\mathbf{\Sigma}|^{-p/2}|\mathbf{\Lambda}|^{-q/2}\text{etr}\left(-1/2(\mathbf{Y}-\mathbf{M})\mathbf{\Sigma}^{-1}(\mathbf{Y}-\mathbf{M})^\top\mathbf{\Lambda}^{-1}\right),$$

where etr(.) *denotes the exponential of the trace.*

Proof. By Definition 2.3, $\mathbf{y} = \text{vec}(\mathbf{Y}) \sim \mathcal{N}_{pq}(\boldsymbol{\mu} = \text{vec}(\mathbf{M}), \mathbf{\Sigma}\otimes\mathbf{\Lambda})$, then the pdf of $\mathbf{y}$ is:

$$p(\mathbf{y}) = (2\pi)^{-pq/2}|\mathbf{\Sigma}\otimes\mathbf{\Lambda}|^{-pq/2}\exp\left(-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^\top\{\mathbf{\Sigma}\otimes\mathbf{\Lambda}\}^{-1}(\mathbf{y}-\boldsymbol{\mu})\right).$$

On one hand: $|\mathbf{\Sigma}\otimes\mathbf{\Lambda}|^{-pq/2} = |\mathbf{\Sigma}|^{-p/2}|\mathbf{\Lambda}|^{-q/2}$ thanks to the Kronecker product property with determinant [155, Chapter 8]. On the other hand, thanks to the properties between *vec* operator, the *Kronecker product* and the *Trace* (Tr) in [155, Theorem 8.12]:

$$\begin{aligned}-\frac{1}{2}(\mathbf{y}-\boldsymbol{\mu})^\top\{\mathbf{\Sigma}\otimes\mathbf{\Lambda}\}^{-1}(\mathbf{y}-\boldsymbol{\mu}) &= -\frac{1}{2}\text{vec}(\mathbf{Y}-\mathbf{M})^\top\{\mathbf{\Sigma}\otimes\mathbf{\Lambda}\}^{-1}\text{vec}(\mathbf{Y}-\mathbf{M}),\\ &= \text{Tr}\left(-\frac{1}{2}(\mathbf{Y}-\mathbf{M})\mathbf{\Sigma}^{-1}(\mathbf{Y}-\mathbf{M})^\top\mathbf{\Lambda}^{-1}\right).\end{aligned}$$

□

Let us add that the separability is not restricted to the Gaussian distribution. [176] proposed a classification method with matrices following a t-distribution.

Maximum likelihood estimation

Let $\mathbf{Y} \in E$ a random matrix and $z \in \{1, \dots, C\}$ a discrete random variable. The generative model is defined by:

$$\mathbf{Y}|z = c \sim \mathcal{MN}_{p,q}(\mathbf{M}_c, \mathbf{\Sigma}_c, \mathbf{\Lambda}_c), \tag{2.8}$$

with parameters $\boldsymbol{\theta}_c = \{\mathbf{M}_c, \mathbf{\Sigma}_c, \mathbf{\Lambda}_c\}$ and $c \in \{1, \dots, C\}$.

For class c we have the set $\mathcal{S}_c = \{\mathbf{Y}_i, z_i = c\}_{i=1}^{n_c}$ of n_c *i.i.d.* samples, distributed according to a matrix-variate Gaussian distribution with mean $\mathbf{M}_c$ and covariance matrices $\mathbf{\Sigma}_c$ and $\mathbf{\Lambda}_c$ (2.8). From Proposition 2.1 and the linearity of the trace, the likelihood is:

$$(2\pi)^{-\frac{n_c pq}{2}} |\mathbf{\Sigma}_c|^{-n_c p/2} |\mathbf{\Lambda}_c|^{-n_c q/2} \text{etr}\left(-\frac{1}{2} \sum_{i|z_i=c} (\mathbf{Y}_i - \mathbf{M}_c)\mathbf{\Sigma}_c^{-1}(\mathbf{Y}_i - \mathbf{M})^\top \mathbf{\Lambda}_c^{-1}\right). \tag{2.9}$$

The optimal parameters $\hat{\boldsymbol{\theta}}_{c,MLE} = \{\hat{\mathbf{M}}_c, \hat{\mathbf{\Sigma}}_c, \hat{\mathbf{\Lambda}}_c\}$ of (2.9) are (see [164]):

$$\hat{\mathbf{M}}_c = \frac{1}{n_c} \sum_{i|z_i=c} \mathbf{Y}_i,$$

and each covariance matrix is estimated given the other covariance matrix as:

$$\hat{\mathbf{\Sigma}}_c = \frac{1}{n_c p} \sum_{i|z_i=c} (\mathbf{Y}_i - \hat{\mathbf{M}})^\top \hat{\mathbf{\Lambda}}_c^{-1} (\mathbf{Y}_i - \hat{\mathbf{M}}),$$

$$\hat{\mathbf{\Lambda}}_c = \frac{1}{n_c q} \sum_{i|z_i=c} (\mathbf{Y}_i - \hat{\mathbf{M}})\hat{\mathbf{\Sigma}}_c^{-1} (\mathbf{Y}_i - \hat{\mathbf{M}})^\top.$$

However, this solution is, firstly, not unique because of identifiability issues in the Kronecker product. Secondly, convergence is obtained by iteratively estimating both matrices.

Flip-flop estimation

The two covariance matrices are estimated iteratively using the so-called *flip-flop* algorithm (see Algorithm 1, steps 3) [53, 123, 164]. In the following we consider a normalizing constant on $\mathbf{\Sigma}_c$ written $r(\mathbf{\Sigma}_c)$ for class $c \in \{1, \dots, C\}$, *i.e.* $\mathbf{\Sigma}$ has a norm equal to 1. One may also consider $\mathbf{\Lambda}_c$ which is normalized. The normalizing constant is usually a norm of a matrix, it appears in steps 3.b and 3.c of Algorithm 1.

Algorithm 1: Flip-flop algorithm

Input : Sample $\{(\mathbf{Y}_i, z_i) \in \mathbb{R}^{p \times q} \times \{1, \dots, C\}, i = 1, \dots, n\}$ and initialization $(\mathbf{\Lambda}_0^{(0)}, \dots, \mathbf{\Lambda}_C^{(0)})$.

Output: $\left(\hat{\theta}_{c,MLE}\right)$, $c = 1, \dots, C$.

1 **for** $c = 1$ **to** C **do**
2 (1) Initialize $k \leftarrow 1$;
3 (2) Compute $\hat{\mathbf{M}}_c \leftarrow \frac{1}{n_c} \sum_{i|z_i=c} \mathbf{Y}_i$;
4 **repeat**
5 (3.a) Update $\hat{\mathbf{\Sigma}}_c^{(k)} \leftarrow \frac{1}{n_c p} \sum_{i|z_i=c} (\mathbf{Y}_i - \hat{\mathbf{M}})^\top \{\mathbf{\Lambda}_c^{(k-1)}\}^{-1} (\mathbf{Y}_i - \hat{\mathbf{M}})$;
6 (3.b) Compute $\eta \leftarrow r\left(\hat{\mathbf{\Sigma}}_c^{(k)}\right)$;
7 (3.c) Update $\hat{\mathbf{\Sigma}}_c^{(k)} \leftarrow \hat{\mathbf{\Sigma}}_c^{(k)} / \eta$;
8 (3.d) Update $\hat{\mathbf{\Lambda}}_c^{(k)} \leftarrow \frac{1}{n_c q} \sum_{i|z_i=c} (\mathbf{Y}_i - \hat{\mathbf{M}}) \{\hat{\mathbf{\Sigma}}_c^{(k)}\}^{-1} (\mathbf{Y}_i - \hat{\mathbf{M}})^\top$;
9 **until** *until (2.9) as converged;*;

The normalizing constant $r(.)$ may be, for example, the Frobenius norm ($\eta = \|\mathbf{\Sigma}_c\|_F = \sqrt{\text{Tr}(\mathbf{\Sigma}_c \mathbf{\Sigma}_c^\top)}$). An other solution is to remove the normalization ($\eta = 1$) and add a special initialization of $\mathbf{\Sigma}$ to ensure that $(\mathbf{\Sigma})_{q,q} = 1$ as a constraint. The latter one has been presented and demonstrated in [166], they have also proved that the associated algorithm converges to a unique extrema.

In our context, the use of this distribution will be extended using Gaussian processes in Chapter 5 with normalizing constant as in flip-flop.

2.2 Gaussian processes

This section focuses on Gaussian processes for both regression and classification tasks. Gaussian process extends the multivariate Gaussian distribution to a continuous set of inputs (also known as *function-space view* [190]). Recall that $\mathcal{S} = \{(\mathbf{y}_i, z_i)|(\mathbf{y}_i, z_i) \in E \times \{1, \dots, C\}\}_{i=1}^n$ denotes a set of n *i.i.d.* samples.

In Section 2.2.1, kernel functions are introduced as they can be used within Gaussian processes, then the Gaussian processes are defined and presented from the *function-space view* in Section 2.2.2, Gaussian processes for classification are described in Section 2.2.3. Finally Section 2.2.4 introduces multi-outputs Gaussian processes.

2.2.1 Kernel functions

Before introducing Gaussian processes, this Section presents a key element which measures the similarity between two samples in space E, denoted by $\mathbf{y}_i$ and $\mathbf{y}_j$, $(i, j) \in \{1, \dots, n\}^2$. Kernels functions, or operators, are symmetric positive-definite functions.

Definition 2.4 (positive semi-definite functions)**.** Let E be an arbitrary finite or infinite space. Let $K : E \times E \to \mathbb{R}$ be a binilear function. K is said to be *positive semi-definite* (psd) if and only if, for all $\mathbf{y} \in E$:

$$K(\mathbf{y}, \mathbf{y}) \geq 0.$$

Positive-definiteness holds if and only if $K(\mathbf{y}, \mathbf{y}) = 0 \Leftrightarrow \mathbf{y} = 0$.

In a so-called "*feature space*" ($\mathbf{y}$ finite or infinite) denoted by $\mathcal{H}$, kernel functions are equivalent to an inner product.

Definition 2.5 (Kernel)**.** Let $\mathcal{H}$ be a Hilbert space with the associated inner product $\langle ., . \rangle_{\mathcal{H}}$, let $\phi : E \to \mathcal{H}$ be a linear or non-linear *mapping function*. Let $\mathbf{y}_i$ and $\mathbf{y}_j$ two samples in E, the kernel function $K : E \times E \to \mathbb{R}$ is defined by:

$$K(\mathbf{y}_i, \mathbf{y}_j) = \langle \phi(\mathbf{y}_i), \phi(\mathbf{y}_j) \rangle_{\mathcal{H}}.$$

Let $\mathbf{K}$ the matrix of size $n \times n$ defined by $(\mathbf{K})_{i,j} = K(\mathbf{y}_i, \mathbf{y}_j), (i, j) \in \{1, \dots, n\}^2$, the matrix $\mathbf{K}$ is called the *Gram matrix*.

Theorem 2.1. *K is a positive (semi-)definite function if and only if the Gram matrix is symmetric positive (semi-)definite.*

Theorem 2.1 is a weaker result from the Mercer's theorem [126].

The family of positive (semi-)definite kernels is wide. Firstly, kernels can be separated into two categories, **stationary** and **non-stationary** kernels. In both cases, we define $\langle ., . \rangle_E$ an inner product on space E and $\|.\|_E$ the associated norm.

Definition 2.6 (Stationary kernels)**.** Let $\mathbf{y}_i$ and $\mathbf{y}_j$ two samples from E. A *stationary kernel* is a kernel function K which only depends on a distance d_E in space E between the two samples:

$$K(\mathbf{y}_i, \mathbf{y}_j) = \tilde{K}\left(d_E(\mathbf{y}_i - \mathbf{y}_j)\right).$$

Among the family of stationary kernels, some examples are presented in the following. Firstly the *Matérn* kernel is, with a positive parameter ν, defined by:

$$K(\mathbf{y}_i, \mathbf{y}_j) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu}\|\mathbf{y}_i - \mathbf{y}_j\|_E}{\ell} \right)^{\nu} K_{\nu} \left(\frac{\sqrt{2\nu}\|\mathbf{y}_i - \mathbf{y}_j\|_E}{\ell} \right), \quad \mathbf{y}_i, \mathbf{y}_j \in E, \tag{2.10}$$

where Γ is the Gamma function and K_ν a modified Bessel function. ℓ is a length scale which measures, for a covariance operator, how strong two samples are correlated.

Taking $\nu \to \infty$ in (2.10) gives us the well-known Squared-Exponential kernel, or *Radial Basis Function* (RBF):

$$K(\mathbf{y}_i, \mathbf{y}_j) = \exp\left(-\frac{\|\mathbf{y}_i - \mathbf{y}_j\|_E^2}{2\ell^2} \right), \quad \mathbf{y}_i, \mathbf{y}_j \in E. \tag{2.11}$$

Many other examples of stationary kernels can be found in [190, Chapter 4] or [118, Section 5.2] such as *periodic* kernels, *rational quadratic* kernels, etc.

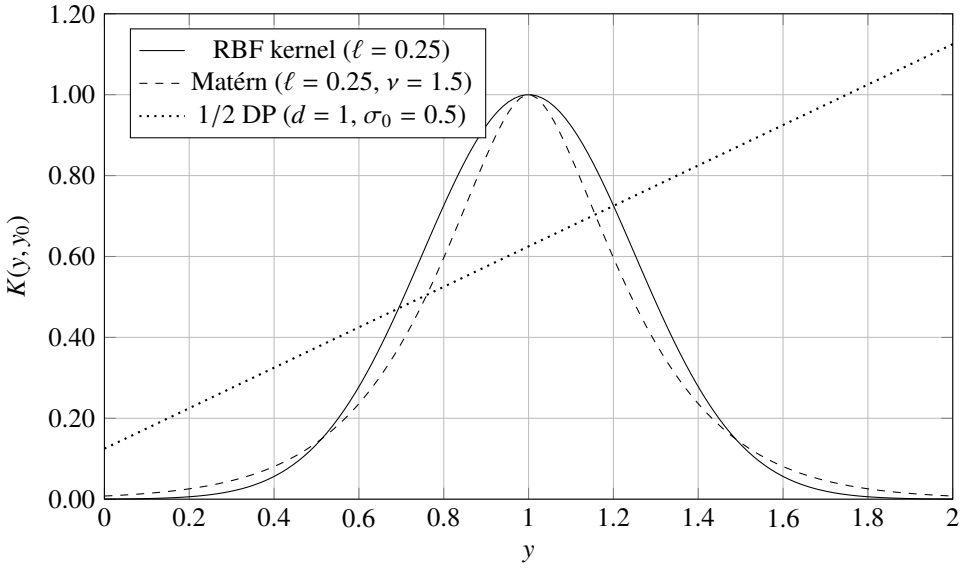


Figure 2.4: Different kernels $K(\mathbf{y}, y_0)$, evaluated for $\mathbf{y} \in [0, 2]$ and $y_0 = 1$: (full line) the RBF kernel with a length-scale $\ell = 0.25$, (dashed line) the *Matérn* kernel with $\nu = 1.5$ and same length-scale as RBF, and (dotted line) the *dot product* (DP) kernel at order 1 and a constant $\sigma_0 = 0.5$, this kernel's outputs are divided by 2 to fit in window.

The non-stationary family is wider, we just mention here the *dot product* kernel defined at order d and up to an additive constant σ_0^2 by:

$$K(\mathbf{y}_i, \mathbf{y}_j) = \left(\langle \mathbf{y}_i, \mathbf{y}_j \rangle_E + \sigma_0^2\right)^d, \quad \mathbf{y}_i, \mathbf{y}_j \in E.$$

If $d > 1$, we also described them as polynomial kernels are defined [190, Chapter 4]. Other non-stationary kernels can be found in the literature such as *neural network* kernels [189] or *spectral* kernels [147]. We also mention here that colored noise (opposed to white noise) can be represented by a non-stationary kernel. Figure 2.4 illustrates some kernels for a one-dimensional space ($E = \mathbb{R}$).

Kernels can also be used in non-parametric techniques for classification [84, Chapter 6]. If the data are linearly not separable (for example using LDA or QDA classifiers), the *mapping function*, introduced in Definition 2.5, can be used to transform the data into a new space, called *feature* space, where the data are separable. The mapping function can be unknown, this technique is also known as *kernel trick* and based on the Mercer's theorem [126].. Support Vector Machines [190, Section 6.4] from the preview in Chapter 1 are using this principle.

Kernels methods are popular among non-parametric techniques in machine learning [88] and can be used on vectors ($E = \mathbb{R}^d$) but also with more complex data (for example when $d \to \infty$). Thus, kernels can be defined on many different type of data such as strings [113], graphs [101] or vector-valued functions [3, 61]. Different types of benchmarks where kernels are applied are presented in [23].

2.2.2 Gaussian processes for regression

Gaussian processes are part of stochastic processes. In a *function-space view* [190], its main advantage is to generalize the distribution on vectors to a continuous process.

Definition 2.7 (Gaussian process)**.** A (uni-dimensional) *Gaussian process* (GP) is a stochastic process $\mathbf{y}$ such that any finite-dimensional marginal follows a multivariate Gaussian distribution.

We write:

$$Y \sim \mathcal{GP}(m, K),$$

where $m(t) = \mathbb{E}(Y_t)$ is the mean function and $K(t, t') = \mathbb{E}\left[(Y_t - m(t))(X_{t'} - m(t'))\right]$ is the covariance operator. Y is real-valued and $t, t' \in \mathcal{T}$ are indexes (times for time-series). $\mathcal{T} = [t_{\min}, t_{\max}]$ denotes the compact set of $\mathbb{R}$ where the samples are observed. Let us highlight that the mean function m and the covariance operator K fully describe the process on $\mathcal{T}$. By Definition 2.7, if $Y \sim \mathcal{GP}(m, K)$ and $\{t_1, \ldots, t_q\} \in \mathcal{T}^q$, then the marginal $\mathbf{y} = [Y_{t_1}, Y_{t_2}, \ldots, Y_{t_q}]^\top$

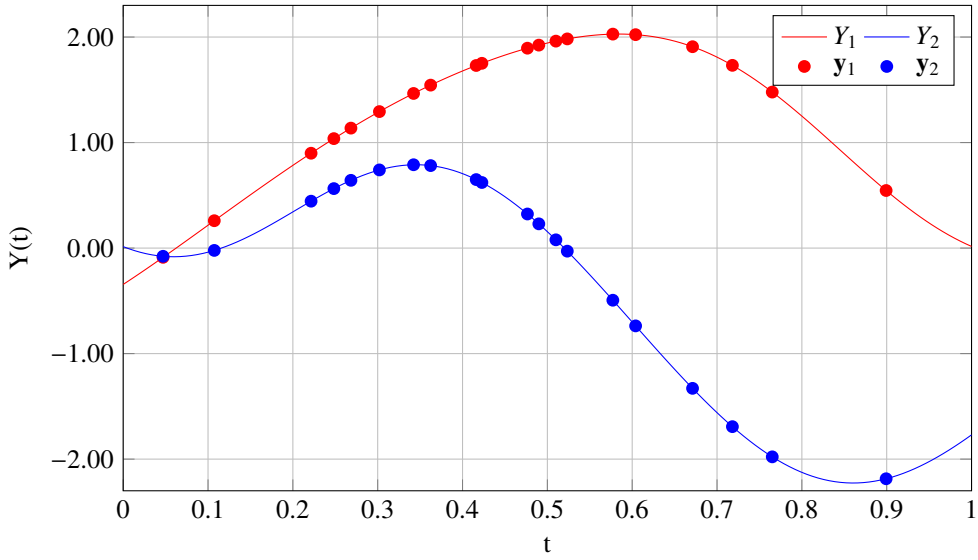


Figure 2.5: Two samples from the same centered GP ($\forall t \in \mathcal{T} = [0,1], m(t) = 0$) with a RBF kernel, (2.11) with length-scale $\ell = 0.25$. The dots represent the marginals from each sample.

of size q is distributed according to a Gaussian distribution as:

$$\mathbf{y} = \begin{bmatrix} Y_{t_1} \\ Y_{t_2} \\ \vdots \\ Y_{t_q} \end{bmatrix} \sim \mathcal{N}_q \left(\begin{bmatrix} m(t_1) \\ m(t_2) \\ \vdots \\ m(t_q) \end{bmatrix}, \begin{bmatrix} K(t_1,t_1) & K(t_1,t_2) & \dots & K(t_1,t_q) \\ K(t_2,t_1) & K(t_2,t_2) & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ K(t_q,t_1) & \dots & \dots & K(t_q,t_q) \end{bmatrix} \right).$$

In the following we write the mean $\boldsymbol{\mu} = [m(t_1), m(t_2), \dots, m(t_q)]^\top$ and the covariance matrix $\boldsymbol{\Sigma}$ such that $(\boldsymbol{\Sigma})_{i,j} = K(t_i, t_j)$, $i, j \in \{1, \dots, q\}$.

Figure 2.5 illustrates two samples Y_1 and Y_2 from a GP with zero mean and radial basis function (RBF) covariance operator. $\mathbf{y}_1$ is a marginal of Y_1 and $\mathbf{y}_2$ a marginal of Y_2 on the same set of indices. The covariance function can be defined by any kernel functions as the symmetric positive semi-definite properties ensure the definition of covariance, see [190].

Bayesian regression problem

Let us now consider a vector $\mathbf{y} \in \mathbb{R}^q$ that is a marginal of size q from a Gaussian process Y:

$$Y \sim \mathcal{GP}(m, K).$$

The vector of inputs is denoted by $\mathbf{t} \in \mathcal{T}^q$. The regression problem is to find a mean function m and a covariance operator K, from which the input-output data $\{\mathbf{t}, \mathbf{y}\}$ are sampled, to predict new outputs from an unobserved input $t_\star \in \mathcal{T}$.

The Bayesian framework considers the GP distribution on the function Y as a *prior* and a new sample $y_\star$ is an output from the input $t_\star$ by the process Y [190, Chapter 2]. As a consequence, as $\mathbf{y}$ and $y_\star$ are marginals from the same Gaussian process, we have:

$$\begin{pmatrix} \mathbf{y} \\ y_\star \end{pmatrix} \sim \mathcal{N}_{q+1} \left[\begin{pmatrix} \boldsymbol{\mu} \\ m(t_\star) \end{pmatrix}, \begin{pmatrix} \boldsymbol{\Sigma} & \mathbf{k}^\top \\ \mathbf{k} & K(y_\star, y_\star) \end{pmatrix} \right], \tag{2.12}$$

with $\boldsymbol{\mu} = [m(t_1), \dots, m(t_q)]^\top$, $(\boldsymbol{\Sigma})_{i,j} = K(t_i, t_j)$ and $\mathbf{k} = [K(t_\star, t_1), \dots, K(t_\star, t_q)]$. If we consider a vector of $q_\star$ outputs $\mathbf{y}_\star$ from inputs $\mathbf{t}_\star$, from (2.12) it is straightforward that:

$$\begin{pmatrix} \mathbf{y} \\ \mathbf{y}_\star \end{pmatrix} \sim \mathcal{N}_{q+q_\star} \left[\begin{pmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu}_\star \end{pmatrix}, \begin{pmatrix} \boldsymbol{\Sigma} & \mathbf{K}^\top \\ \mathbf{K} & \boldsymbol{\Sigma}_\star \end{pmatrix} \right],$$

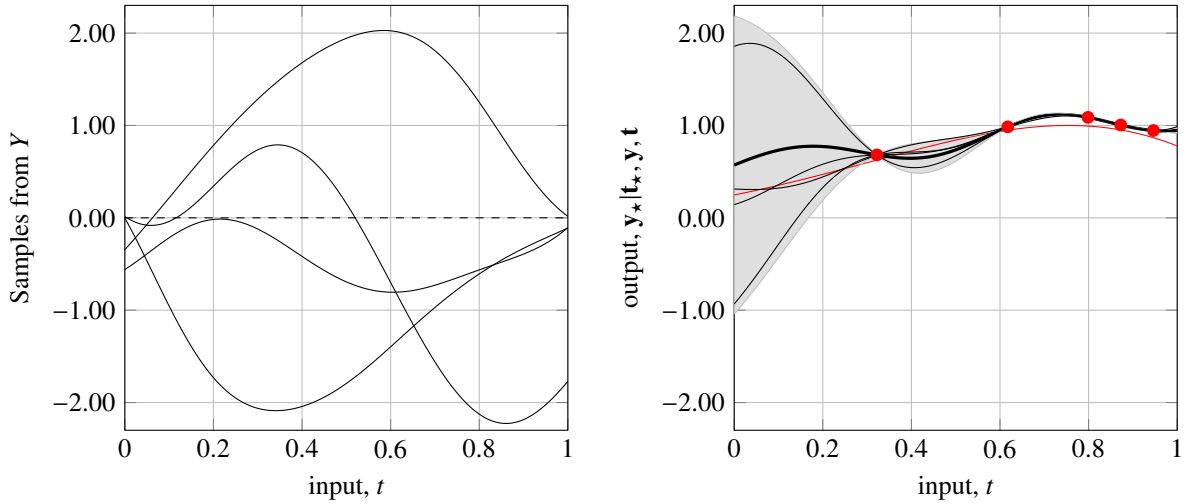


Figure 2.6: On the left: samples from a centered GP (black lines, as Figure 2.5), zero-mean is highlighted by a dashed line. On the right: mean $\mathbb{E}(\mathbf{y}_\star|\mathbf{t}_\star, \mathbf{y}, \mathbf{t})$ (thick black line) with 95% confidence interval conditionally (grey zone) to observed noisy samples (red dots) from a toy function (red line). Black lines are sampled from the conditional distribution (2.13).

$\boldsymbol{\mu}_\star = [m(t_{1,\star}), \ldots, m(t_{q_\star,\star})]$, $(\mathbf{K})_{k,j} = K(t_{k,\star}, t_j)$ and $(\boldsymbol{\Sigma}_\star)_{k,l} = K(t_{k,\star}, t_{l,\star})$ where $(k,l) \in \{1, \ldots, q_\star\}^2$ and $j \in \{1, \ldots, q\}$.

By the properties of Gaussian distributions, $\mathbf{y}_\star|\mathbf{t}_\star, \mathbf{y}, \mathbf{t}$ is a $q_\star$-variate Gaussian vector with mean and covariance matrix:

$$\begin{aligned} \mathbb{E}(\mathbf{y}_\star|\mathbf{t}_\star, \mathbf{y}, \mathbf{t}) &= \boldsymbol{\mu}_\star + \mathbf{K}\boldsymbol{\Sigma}^{-1}(\mathbf{y} - \boldsymbol{\mu}) \\ \operatorname{cov}(\mathbf{y}_\star|\mathbf{t}_\star, \mathbf{y}, \mathbf{t}) &= \boldsymbol{\Sigma}_\star - \mathbf{K}\boldsymbol{\Sigma}^{-1}\mathbf{K}^\top. \end{aligned} \tag{2.13}$$

Figure 2.6 illustrates the mean ($\mathbb{E}(\mathbf{y}_\star|\mathbf{t}_\star, \mathbf{y}, \mathbf{t})$) and 95% confidence interval ($\pm 1.96\sqrt{\mathbb{V}(\mathbf{y}_\star|\mathbf{t}_\star, \mathbf{y}, \mathbf{t})}$) on $\mathcal{T} = [0, 1]$, computed thanks to (2.13) with $m(t) = 0$ and K a RBF kernel function. We refer to [190, Chapter 2, Algorithm 2.1] for a practical implementation.

Gaussian processes are well-known in regression studies [3, 134, 149]. They are used for example to smooth observations [177] or in the context of earth-observation to analyse data [31]. Extensions in the regression context are proposed as Manifold GP [30], or GP with constraints [100, 139].

Finally, non-Gaussian processes are also studied in the literature. One of the most promising distribution is to use *Student-t processes* [156]. A student-t distribution allows heavier tails as presented in Figure 2.1, they are robust w.r.t. outliers in the *latent function*. In [37] the authors presented multivariate Student-t processes and compared them to Gaussian distributions (see Section 2.2.4).

This manuscript does not go any further in the field of regression as the context of our work is oriented towards classification. However, it is important to introduce GPs in a function-space view and their conditional distribution on a regression framework as they are used in Chapter 4 for temporal reconstruction of satellite images with Gaussian processes.

2.2.3 Gaussian processes for classification

Gaussian processes are flexible and can be adapted to different situations. In this section, our aim is to transform real-valued inputs into C discrete outputs (or classes). To this end, Gaussian processes are used in a discriminative approach, *i.e.*, given $c \in \{1, \ldots, C\}$, one GP models the posterior probability $\mathbb{P}(z = c|\mathbf{y})$.

Gaussian processes as latent functions

As introduced, the use of Gaussian processes for classification is based on a set of centered GP priors on a functional f, called the *latent function*, from an input space to the space of real numbers ($f : E \to \mathbb{R}$).

The main idea of GP for classification is to define a conditional Bernoulli distribution thanks to a transformation of the latent function f through a *logit* function $\sigma : \mathbb{R} \to [0, 1]$. Let c be a given class, the conditional Bernoulli

distribution for class c is denoted by $\mathbb{P}(Z = c|\mathbf{y})$ (with the previous notations) and is defined by:

$$\mathbb{P}(Z = c|\mathbf{y}) = \sigma(f^c(\mathbf{y})), \tag{2.14}$$

with $f^c \sim \mathcal{GP}(0, K_c)$, *i.e.* f^c is completely described by the covariance operator K_c.

Let us now consider C classes. Recall that $\mathcal{S}$ is the set defined by $\mathcal{S} = \{\mathbf{Y}, \mathbf{z}\} \in \mathbb{R}^{n\times q} \times \{1, \dots, C\}^n$. We define $\mathbf{f}$ as the vector of latent functions of size nC defined by:

$$\mathbf{f} = [f_1^1, \dots, f_n^1, f_1^2, \dots, \dots, f_1^C, \dots, f_n^C]^\top,$$

where f_i^c is the latent function for sample $(\mathbf{Y})_i$ of size q for class c.
Then $\mathbf{f} \sim \mathcal{GP}(0, K)$ where $K = \text{diag}(K_1, \dots, K_C)$ as the C processes are assumed to be independent and K_c is the covariance operator in class c.

Inference and approximations for the *a-posteriori* probabilities

Inference is complex, it aims to return the *a-posteriori* probabilities for each class $c \in \{1, \dots, C\}$. It is done in two steps [190, Chapter 3]:

1. Compute the likelihood density distribution over the latent function $\mathbf{f}_\star = [f_\star^1, \dots, f_\star^C]^\top$ for a test case $\mathbf{y}_\star$:

$$p(\mathbf{f}_\star|\mathbf{Y}, \mathbf{z}, \mathbf{y}_\star) = \int p(\mathbf{f}_\star|\mathbf{Y}, \mathbf{y}_\star, \mathbf{f}) p(\mathbf{f}|\mathbf{Y}, \mathbf{z}) d\mathbf{f}, \tag{2.15}$$

   where $p(\mathbf{f}|\mathbf{Y}, \mathbf{z}) = p(\mathbf{z}|\mathbf{f}, \mathbf{Y}) p(\mathbf{f}|\mathbf{Y}) / p(\mathbf{z}|\mathbf{Y})$ is the posterior over the latent variables for known data.

2. Produce the conditional Bernoulli distribution on $\mathbf{y}_\star$ for all classes $c \in \{1, \dots, C\}$:

$$\hat{\mathbb{P}}(Z = c|\mathbf{y}_\star) \propto \int \sigma(f_\star^c) p(f_\star^c|\mathbf{Y}, \mathbf{z}, \mathbf{y}_\star) df_\star, \tag{2.16}$$

   with the likelihood defined in (2.15). The notation $\hat{\mathbb{P}}$ refers to the use of $\mathbf{f} = \hat{\mathbf{f}}$, *i.e.* the latent function which maximizes (2.15).

In this situation, the integral (2.16) is intractable for some choice of sigmoid functions. Additionally the likelihood (2.16) is non-Gaussian [190, Chapter 3]. To overcome this problem, a solution is to approximate the posterior with a Gaussian distribution, two solutions are found in the literature: the *Laplace approximation* [189], we refer to [190, Algorithms 3.3 and 3.4] for practical implementation, and the *expectation propagation* (EP) method [127, 133], see [190, Algorithms 3.5 and 3.6]. The following describes the two approximations.

On one hand, the *Laplace approximation* approaches the posterior density $p(\mathbf{f}|\mathbf{Y}, \mathbf{z})$ over $\mathbf{f}$ by a Gaussian approximation, denoted by $q(\mathbf{f}|\mathbf{Y}, \mathbf{z}) = \mathcal{N}(\mathbf{f}|\hat{\mathbf{f}}, A^{-1})$ [190, Section 3.4] where $\hat{\mathbf{f}}$ is the latent function $\mathbf{f}$ which maximizes the likelihood (2.15) and $A = \nabla\nabla_{\mathbf{f}=\hat{\mathbf{f}}} \log p(\mathbf{f}|\mathbf{Y}, \mathbf{z})$ is the Hessian of the negative log posterior evaluated at $\hat{\mathbf{f}}$. The new likelihood is Gaussian and denoted by $q(\mathbf{f}_\star|\mathbf{Y}, \mathbf{z}, \mathbf{y}_\star)$. In [190, Section 3.5], they showed that the predictive Gaussian likelihood mean and covariance are:

$$\begin{aligned} \mathbb{E}(\mathbf{f}_\star|\mathbf{Y}, \mathbf{z}, \mathbf{y}_\star) &= \mathbf{Q}_\star^\top (\mathbf{z}^c - \left[\hat{\mathbb{P}}(Z = c|\mathbf{y}_1), \dots, \hat{\mathbb{P}}(Z = c|\mathbf{y}_n)\right]^\top), \\ \text{cov}(\mathbf{f}_\star|\mathbf{Y}, \mathbf{z}, \mathbf{y}_\star) &= \mathbf{\Sigma} + \mathbf{Q}_\star^\top \mathbf{K}^{-1} (\mathbf{K}^{-1} + \mathbf{W})^{-1} \mathbf{K}^{-1} \mathbf{Q}_\star, \end{aligned} \tag{2.17}$$

where $\mathbf{Q}_\star$ is a block diagonal matrix defined by $\mathbf{Q}_\star = \text{diag}(\mathbf{k}_1(\mathbf{y}_\star), \dots, \mathbf{k}_C(\mathbf{y}_\star))$ of size $Cn \times C$ with $\mathbf{k}_c(\mathbf{y}_\star) = [K_c(\mathbf{y}_1, \mathbf{y}_\star), \dots, K_c(\mathbf{y}_n, \mathbf{y}_\star)]^\top$, $\mathbf{K} = \text{diag}(\mathbf{K}_1, \dots, \mathbf{K}_C)$ is the previous diagonal matrix where each $\mathbf{K}_c$ is the Gram matrix in class c and $\mathbf{W}$ is a diagonal matrix [190, Equation (3.38)]. Finally $\mathbf{\Sigma}$ is a diagonal matrix of size $C \times C$ where $(\mathbf{\Sigma})_{c,c} = K_c(\mathbf{y}_\star, \mathbf{y}_\star) - \mathbf{k}_c^\top(\mathbf{y}_\star) \mathbf{K}_c^{-1} \mathbf{k}_c(\mathbf{y}_\star)$.

On the other hand, the *Expectation propagation* algorithm (EP) has been presented by [127]. A main difference with the Laplace approximation is that the EP algorithm is used in the binary classification case ($C = 2$), corresponding to (2.14) where σ can be changed easily. It means that only one posterior probability is necessary, for example we consider the first class $\mathbb{P}(Z = 1|\mathbf{y}_\star)$ (as $\mathbb{P}(Z = 2|\mathbf{y}_\star) = 1 - \mathbb{P}(Z = 1|\mathbf{y}_\star)$), *i.e.* one latent function is necessary and $\mathbf{f} = [f_1, \dots, f_n]^\top$. The non-Gaussian posterior is now given by the Bayes' rule as:

$$p(\mathbf{f}|\mathbf{Y}, \mathbf{z}) = \frac{p(\mathbf{f}|\mathbf{Y})}{p(\mathbf{z}|\mathbf{Y})} \prod_{i=1}^{n} p(\mathbf{y}_i|f_i),$$

where the likelihood $p(\mathbf{y}_i|f_i)$ remains non-Gaussian. However, with the use of the probit likelihood (cumulative density function of a standard Gaussian distribution), the integral of the posterior can be computed. EP framework approximates the likelihood thanks to a *local likelihood* [190, Section 3.6]. We also refer to [190, Section 3.6, Algorithms 3.5 and 3.6] for a complete and practical implementation.

To understand the limitation of Gaussian processes for classification when dealing with large data-sets (as land use of land cover applications), it is important to highlight that these algorithms scale in $O((C+1)n^3)$, *i.e.* cubic w.r.t. the number of samples. Additionally, within the Laplace approximation algorithm, a Monte-Carlo loop has to be done to obtain the posterior probabilities (2.14).

Large data-sets approximations and scaling

The following presents how, in the literature, the problem of large data-sets with GP for classification has been tackled.

Recall that, for regression and classification, the computation complexity is $O(n^3)$ as the Gram matrix $\mathbf{K}$ has to be inverted and depends on the number of samples n in the set of data $\mathcal{S}$. In order to reduce complexity, a first case to consider is to work on a well chosen subset of $\mathcal{S}$. Also called sub-sampling, it takes a smaller number m of data where $m \ll n$. The algorithm scales in $O(m^3)$ and [85] presented some theoretical results on the error made by this strategy. A second case is to sparsify the matrix $\mathbf{K}$ to make the inversion easier. One example is the *Nyström technique*, it replaces the matrix $\mathbf{K}$ by $\tilde{\mathbf{K}}$ defined in [191] by:

$$\tilde{\mathbf{K}} = \mathbf{K}_{n,m}\mathbf{K}_{m,m}^{-1}\mathbf{K}_{m,n}, \tag{2.18}$$

where $\mathbf{K}_{p,q}$ are blocks of size $p \times q$ from $\mathbf{K}$ constructed by the evaluation of the covariance operator K between the first p samples and the first q samples. In practice, $m \ll n$ and the inversion cost of $\tilde{\mathbf{K}}$ in (2.18) is $O(m^2 n)$. We refer to [70, 191] and [190, Section 8.3] for discussions on the limits of this method and a criteria to select m. An other approach is to approximate the marginal likelihood. A typical one is the use of the variational inducing point framework [86] where the likelihood is lower bounded by a quantity with lower complexity. It results in a complexity in $O(m^3)$.

A complete review, considering these approximations and many more can be found in [111]. They also reviewed extensions as deep GP [47] or more complex structures as multi-task GP [3] which is reviewed in the next Section. Finally, let us highlight that similar problems are found in the use of GP for regression as in Section 2.2.2. A review of scalable approximations can be found in [145].

The GP classifiers will not be used as a latent function to model the *a-posteriori* probability. The use of GPs in our work will be based on a generative approach. We will show that the inference algorithm scales linearly w.r.t. n.

2.2.4 Multi-output Gaussian processes

This section presents a brief review of GP techniques to deal with multi-output processes. Let $\mathbf{Y}(t) = [Y_1(t), \dots, Y_p(t)]^\top, t \in \mathcal{T}$ a process with p outputs. This section aims to present GP models that take into account correlation between outputs $Y_b, b \in \{1, \dots, p\}$.

The models of multi-output GP were originally introduced in the geostatistics literature [45, 75]. For any given $P \in \mathbb{N}^\star$ (let us highlight that the number of components (P) and the number of outputs (p) are not necessarily equals), the p models are composed by P latent functions, or processes, such that the b-th function, $b \in \{1, \dots, p\}$, is a linear combination of the latent functions. $\forall (t, t') \in \mathcal{T}^2$:

$$Y_b(t) = \sum_{\ell=1}^{P} a_{b,\ell} u_\ell(t), \tag{2.19}$$

where the latent functions $\{u_\ell(t)\}_{\ell=1}^P$ are centered GP with covariance operator given by $\operatorname{cov}(u_\ell(t), u_{\ell'}(t')) = k_\ell(t, t')$ if $\ell = \ell'$, zero otherwise. This model is known as the *Linear Model of Coregionalization* [75] (or LMC). The LMC covariance operators are known to have a sum of separated kernels [3], *i.e.* the covariance operator in time does not model the covariance between (outputs) and the linear combination of outputs does not depend on the time.

Let $P_\ell \leq P$ be the total number of functions $\{u_k(t)\}_{k=1}^{P_\ell}$ sharing the same covariance [3, Equation (18)], the general covariance operator matrix, sized $p \times p$, associated with (2.19) is:

$$K(t, t') = \sum_{\ell=1}^{P} \mathbf{C}_\ell k_\ell(t, t'), \tag{2.20}$$

where $\mathbf{C}_\ell$, a $p \times p$ kernel matrix, is known as the *coregionalization matrix*, $\mathbf{C}_\ell = \mathbf{A}_\ell \mathbf{A}_\ell^\top$ where $(\mathbf{A}_\ell)_{b,k} = a_{b,\ell}^k$ is a matrix of size $p \times P_\ell$. This model is used mostly in regression context with data of fixed length, [38] uses LMC on time-series to monitor patient state using clinical covariates as features. [104] uses spatial information on one hand and soil properties on the other hand to model soil's variograms.

For a fiven $\ell \in \{1, \dots, P\}$, let $\mathbf{B}$ be a $p \times p$ matrix which does not depend on ℓ. A particular case of LMC is to consider $\mathbf{C}_\ell$ equal to $\mathbf{B}$ up to a multiplicative constant c_ℓ: $\mathbf{C}_\ell = \mathbf{B}c_\ell$. Then

$$K(t, t') = \mathbf{B} \sum_{\ell=1}^{P} c_\ell k_\ell(t, t') = \mathbf{B}\tilde{k}(t, t'). \tag{2.21}$$

This model is known as *Intrinsic Coregionalization Model* [3] (ICM) where the covariance operator matrix (2.20) is given in (2.21). ICM has been derived under multiple names in the machine learning literature where, most of the time, the introduction of the model is different.

An example, from [18], is the *multi-task* Gaussian processes model. It assumes that the family of latent functions $\{Y_b\}_{b=1}^p$ in one direction has a GP prior to induce correlation between "tasks". For $(b, b') \in \{1, \dots, p\}^2$, $(t, t') \in \mathcal{T}^2$, the model is defined by the correlation:

$$\langle Y_b(t), Y_{b'}(t') \rangle = (\mathbf{B}^Y)_{b,b'} k(t, t'), \qquad Y_b(t) \sim \mathcal{N}(f_b(t), \sigma_b^2),$$

where $Y_b(t)$ is the b-th output at input $t \in \mathcal{T}$, $\mathbf{B}^Y$ is the covariance between outputs and k is the covariance operator on inputs. Other works derived from ICM are presented in [16, 131].

These models are mostly applied in the context of regression with inputs of same length, we refer to [37, 110] for recent reviews with various applications as species distribution models [94] or remote sensing [141]. Some strategies with large data-sets are also proposed as in [108] but most of them remain quadratic or cubic w.r.t. n the number of samples. An other example is the *Semi-parametric Latent Factor Model* [172] for multiple response variables using GP as a prior, it assumes the same structure as (2.21) but the complexity of the conditional dependency between responses imposes to use sparse approximations of the posterior. Nowadays, the multi-outputs GP are used also for classification [50] where the principle is similar to GP for classification (Section 2.2.3) or [162] with a variational approach.

Recently, extensions of LMC are proposed as the *convolutional Gaussian processes*. The convolutional Gaussian processes have been defined in [2]. Starting from (2.19) and adding a convolution between covariance kernels on outputs $\{a_b(t)\}_{b=1}^p$ (smoothing kernels in [2]) and P latent functions $\{u_\ell\}_{\ell=1}^P$. Let $(t, s) \in \mathcal{T}^2$, one of the most general form of the model is:

$$f_b(t) = \sum_{\ell=1}^{P} \int_{\mathcal{T}} a_{b,\ell}(t - s) u_\ell(s) ds, \tag{2.22}$$

with $b \in \{1, \dots, p\}$ and $\ell \in \{1, \dots, P\}$ and $\{u_\ell(t)\}_{\ell=1}^P$ are centered GP. It results in a dependency between outputs which is varying with inputs.

Other extensions from convolutional Gaussian processes are presented in [25] where outputs share similar latent functions. In [33] they added a graph regularization and proposed an Expectation Maximization (EM) algorithm for inference. These extensions are often applied in the context of regression.

2.3 Statistical modelling for Sentinel-2 satellite image time-series

This chapter has presented a state-of-the-art in supervised model-based classification with an in-depth study of Gaussian distributions. These techniques are presented to understand both discriminative analysis and matrix-variate Gaussian distributions as they are keys to understand the contributions of this thesis. To the best of our knowledge, no technique is well suited for application to SITS with missing time stamps at large scale with the challenges arise from Sentinel-2 mission.

This chapter has also presented Gaussian processes for regression and classification. Learning Gaussian processes for classification has been shown to be cubic with respect to the number of samples in the provided training set. The multi-output extensions of Gaussian processes have been reviewed, this thesis will contribute specifically to this field by formerly defining a mixture of multivariate GP which scales to large datasets with missing values. Recent extensions with convolutional GP are presented for a discussion on these models.

The following introduces the contributions. We propose two generative models based on Gaussian processes with application to classification of irregularly and unevenly sampled time-series.

Chapter 3

Model-based classification for irregularly sampled time-series

Outline

French introduction

Ce chapitre a pour objectif d'introduire les deux contributions de cette thèse. Ces deux contributions peuvent être spécifiées par un modèle de processus gaussiens. Ces deux modèles, nommés MIMGP et M2GP, font l'hypothèse, pour le premier, d'indépendance entre les bandes spectrales des SITS de Sentinel-2, et pour le second, d'une combinaison linéaire de processus gaussiens latents ayant le même opérateur de covariance temporel.

Ces deux modèles de classification sont introduits pour modéliser d'un point de vue fonctionnel les séries temporelles. Considérer des séries temporelles observées à des instants différents revient alors à considérer une discrétisation temporelle de la fonction modélisée. Ensuite la reconstruction de données manquantes (nuages ou ombre) est détaillée.

Ce chapitre se conclut sur la présentation des annexes consacrées à la description du code, pour lequel le choix du langage python a été fait et optimisé avec le langage cython (langage compilé).

Let $\mathbf{Y}^\star$ (the notation will be argued later) be a random matrix of size $p \times q$, it contains the SITS reflectance at p spectral bands and q temporal acquisitions after noise removal. Let Z be a discrete random variable taking its values in $\{1, \dots, C\}$. The decision rule, from (2.1), is given by:

$$\begin{aligned} \hat{c} &= \arg\max_c \mathbb{P}(Z = c|\mathbf{Y}^\star) \\ &= \arg\max_c \pi_c p(\mathbf{Y}^\star|Z = c), \qquad c \in \{1, \dots, C\}. \end{aligned}$$

Our aim is to build a generative model, *i.e.* to assume a distribution on the conditional random variable $\mathbf{Y}^\star|Z = c$ where $\mathbf{Y}^\star \in E$ is the reflectance, and is distributed conditionally to its class membership $Z = c$. This chapter describes how Gaussian processes can model irregularly and unevenly sampled time-series and links our contributions to state of the art.

Section 3.1 explains how Gaussian processes overcome the issue of irregular samples and presents their use in the classification step. Section 3.2 shows how the models can be used to perform the continuous reconstruction of missing time-stamps using the complete mixture. Finally Section 3.3 introduces the implementation of two models.

3.1 Classification of irregularly sampled time-series

Section 3.1.1 presents the general model and explains why such model is able to handle irregularly and unevenly sampled time-series. Section 3.1.2 presents a one-dimensional irregularly sampled time-series classifier and presents the classifier for multiple spectral bands with independence assumption. Finally Section 3.1.3 presents the model with linear dependency between wavelengths.

3.1.1 Continuous representation of SITS

The following defines the general model behind the two contributions.

Preliminary definitions

In the following, E denotes the data space for re-sampled time-series data ($E = \mathbb{R}^q$). In the following we assume to observe the time-series continuously on a given time window $\mathcal{T} = [t_{\min}, t_{\max}]$. The multi-dimensional functions with p square integrable processes is denoted by $\mathbf{Y}(t)$ and defined as:

$$\begin{aligned} \mathbf{Y} : \mathcal{T} &\to \mathbb{R}^p, \\ t &\mapsto [Y_1(t), \ldots, Y_p(t)]^\top. \end{aligned} \tag{3.1}$$

A discretized observation from a continuous process $\mathbf{Y}$ is called *marginal* and is denoted by $\mathbf{Y}^\star$. As presented previously, it is a random matrix of size $p \times q$.

In practice, the observations are Sentinel-2 SITS. The reflectance are included in the set $\mathcal{S}$ with the associated class membership. Considering the i-th sample yields to the marginal matrix $\mathbf{Y}^{i,\star}$ of size $p \times q_i$. If, $\forall i \in \{1, \ldots, n\}$, $q_i = q$ and, moreover, the time stamps are equal, then the data space $E = \mathcal{M}_{p,q}(\mathbb{R})$ is the space of matrices with same size. The classifiers are explored in Section 2.1.

In our context, clouds and shadows noises occur at random spatial locations and time stamps. The resulting effect is different sample sizes, *i.e.* $q_i \neq q_j$, and different time stamps. An example of two simulated time-series is given in Figure 3.1: in this case $q_1 = 18$ and $q_2 = 14$. By taking into account level 2A products (surface reflectance $\tilde{\mathbf{Y}}^{i,\star}$ on p spectral bands, the associated mask $\mathbf{M}^i$) and removing noisy data (mask values above '0') yields the set $\mathcal{S} = \{\mathbf{Y}^{i,\star}, z_i\}_{i=1}^n$ of n irregularly sampled SITS where each i-th sample has its own size: $p \times q_i$ as the number of clear dates may change; z_i is the associated class membership to sample i, $i \in \{1, \ldots, n\}$.

General model

The general model assumes that, conditionally to $z = c$:

$$\mathbf{Y}(t) = \mathbf{A}_c \mathbf{W}(t) + \mathbf{m}_c(t), \forall t \in \mathcal{T}, \tag{3.2}$$

where $(b, b') = \{1, \ldots, p\}^2$, $W_b \sim \mathcal{GP}(0, K_{b,c})$ are independent latent processes. For $b \neq b'$, the independence property is denoted by $W_b \perp\!\!\!\perp W_{b'}$. $\mathbf{m}_c : \mathcal{T} \to \mathbb{R}^p$ is a vector mean function and is presented in Section 5.3 and $K_{b,c}$ is the b-th covariance operator for class c. Model (3.2) results in the *function-space view* [190, Section 2.2] framework as it defines a distribution over functions.

Equivalently, conditionally to $Z = c$, the b-th spectral band is a linear combination of the p latent processes:

$$Y_b(t) = m_{b,c}(t) + \sum_{\ell=1}^{p} (\mathbf{A})_{b,\ell} W_k(t). \tag{3.3}$$

where $W_k \sim \mathcal{GP}(0, K_{k,c})$.

3.1.2 One-dimensional process and the Mixture of Independent multivariate Gaussian processes

The first contribution assumes that the p spectral bands are idependent, *i.e.* we consider p one-dimensional processes. For one spectral bands, the model (3.2) becomes:

$$Y(t) \triangleq Y_1(t) = a_c W_1(t) + m_c(t), \forall t \in \mathcal{T},$$

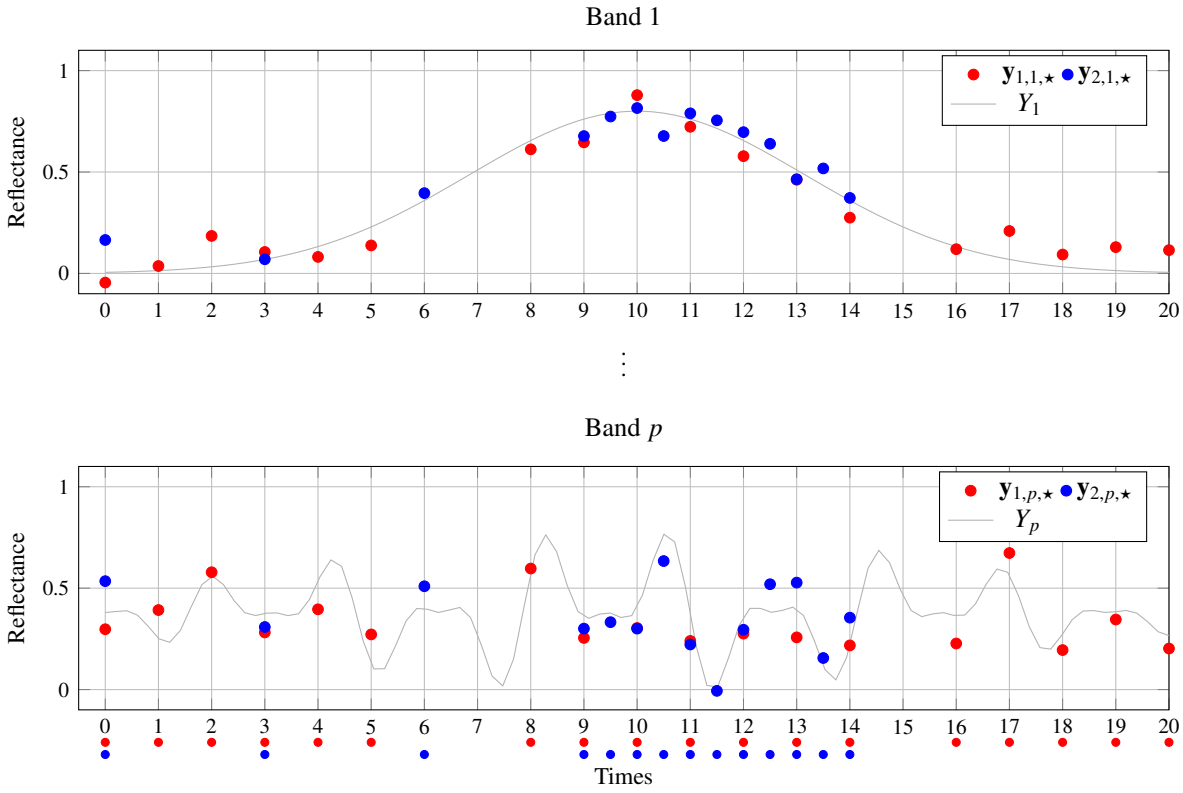


Figure 3.1: Simulated satellite image time series at bands 1 and p. The dots below the temporal axis correspond to the temporal sampling.

where $W \triangleq W_1 \sim \mathcal{GP}(0, \tilde{K}_c)$. Then it is straightforward that:

$$Y \sim \mathcal{GP}(m_c, a_c^2 \tilde{K}_c).$$

Then the problem is to consider, for a fixed spectral band, the i-th sample which is assumed to follow a multivariate normal distribution (see Definition 2.7).

Mixture of Independent multivariate Gaussian processes

This section introduces our first contribution in Chapter 4, with associated supplementary materials in Appendix B. From (3.2), the model assumes that, conditionally to $Z = c$, the p processes $(Y_1, \ldots, Y_p)$ are independent, *i.e.* $\mathbf{A}_c = \mathbf{I}$, it comes the following decision rule:

$$\begin{aligned}\hat{c} &= \arg\max_c \pi_c p(\mathbf{Y}^\star | Z = c) \\ &= \arg\max_c \pi_c \prod_{b=1}^{p} p(Y_b^\star | Z = c),\end{aligned}$$

where $Y_b \sim \mathcal{GP}(m_{b,c}, K_{b,c})$. It consists of p independent processes, models inference is detailed in Section 4.2.3. The classification accuracy results, computed on a separate validation set, are presented in Section 4.6 and are compared with the state of the art methods introduced in Chapter 1. State-of-the-art methods take as input linearly re-sampled time-series as presented in Figure 1.12.

Inference of model parameters (mean coefficients and kernel parameters) on the set $\mathcal{S}$ for is provided in Section 4.2.3. It is inferred using a *maximum likelihood estimator* (MLE). This estimation is possible as every sample is a marginal from a continuous process and the complete process is inferred from MLE, despite irregular time stamps for each sample i.

To the best of our knowledge, this contribution introduces the first model, namely Mixture of Independent Multivariate Gaussian Processes (MIMGP), which is able to classify irregularly sampled data. We complete the presentation of MIMGP with an application to Sentinel-2 SITS using non-noisy data, $p = 10$ spectral bands and

time stamps distributed as in Figure 4.6. Additionally, this model is able to reconstruct time-series at any time $t^\dagger \in \mathcal{T}$ and is described in Section 3.2.

When the class memberships are unknown (*unsupervised* framework), the definition of an unsupervised classifier is straightforward and inference with EM (Expectation-Maximization) algorithm is simple thanks to the Gaussian assumption.

3.1.3 Mixture of multivariate Gaussian processes

Highlighted in the first contribution, one of the weaknesses of the MIMGP model is the independence assumption. To this end we propose a new model which deals linearly with the p outputs of the vector function $\mathbf{Y}$ (see (3.1)): the mixture of multivariate Gaussian processes (M2GP, Chapter 5). Let us describe the construction of M2GP.

Our aim in Chapter 5 is to obtain the mean and the variance of the Gaussian distribution $\mathbf{Y}^{i,\star}|z_i = c$ (see Section 5.8) for the i-th matrix sample of size $p \times q_i$ or, similarly, $\mathbf{y}_{i,\star} = \text{vec}(\mathbf{Y}^{i,\star})$. The mean is straightforward to compute, and the covariance is given by:

$$\text{cov}\left(Y_b(t_j^i), Y_{b'}(t_{j'}^i)\right) = \text{cov}\left(\mathbf{a}_{c,b} W_b(t_j^i), \mathbf{a}_{b',c} W_{b'}(t_{j'}^i)\right)$$

where $\mathbf{a}_{c,b}$ is the b-th line of $\mathbf{A}_c$, then, as $W_b \perp\!\!\!\perp W_{b'}$ for $b \neq b'$, then:

$$\begin{aligned}\text{cov}\left(Y_b(t_j^i), Y_{b'}(t_{j'}^i)\right) &= \mathbf{a}_{c,b} \Sigma_{j,j'}^{b,c,i} \mathbf{a}_{c,b'}^\top, \qquad \text{if } b = b', \\ &= 0, \qquad\qquad \text{otherwise.}\end{aligned}$$

Applied to all time stamps $\mathbf{t} = [t_1^i, \ldots, t_{q_i}^i]$ and all p spectral wavelengths, it comes:

$$\text{cov}(\mathbf{y}_{i,\star}|z_i = c) = \begin{pmatrix} \mathbf{A}_c \text{diag}\left([\Sigma_{1,1}^{1,c,i}, \ldots, \Sigma_{1,1}^{p,c,i}]\right) \mathbf{A}_c^\top & \mathbf{A}_c \text{diag}\left([\Sigma_{1,2}^{1,c,i}, \ldots, \Sigma_{1,2}^{p,c,i}]\right) \mathbf{A}_c^\top & \ldots \\ \mathbf{A}_c \text{diag}\left([\Sigma_{2,1}^{1,c,i}, \ldots, \Sigma_{2,1}^{p,c,i}]\right) \mathbf{A}_c^\top & \mathbf{A}_c \text{diag}\left([\Sigma_{2,2}^{1,c,i}, \ldots, \Sigma_{2,2}^{p,c,i}]\right) \mathbf{A}_c^\top & \ldots \\ \vdots & \vdots & \ddots \end{pmatrix}.$$

The covariance matrix is of size $pq_i \times pq_i$ which makes the inference numerically difficult. One solution, from (3.2), is to assume that $K_{b,c} = K_c, \forall b \in \{1, \ldots, p\}$, *i.e.* $\forall b, W_b \sim \mathcal{GP}(0, K_c)$. This defines the M2GP model (5.1).

The hypothesis of the same time-covariance operator yields a Kronecker product structure which allows a significant decrease of numerical complexity. All marginals are now distributed according to a matrix-variate Gaussian distribution (see Definition 2.3).

Parameters estimation is presented in Section 5.6.2. Figure 5.6 shows significant performance improvements when adding some dependence compared with MIMGP. Finally comparisons on Sentinel-2 SITS are reported in Table 5.2 with LULC maps in Figure 5.12.

3.2 Reconstruction of noisy time stamps

An additional and promising application is the reconstruction of noisy reflectance values. An example is the use of Sentinel-2 SITS for phenological research, a review of applications can be found in [128]. As presented in Chapter 1, reconstruction of missing values is also an active area of study in remote sensing [158].

3.2.1 Imputation with independence assumption

The imputation of missing values is possible thanks to the Gaussian marginal at a new time input $t^\dagger \in \mathcal{T}$ from the continuous process modeled by a GP. The joint distribution (2.12) allows the imputation of missing values when the class membership is known using conditional expectation and conditional covariance [17, p.63] defined in (2.13). The reconstruction of the p spectral bands is done independently using the conditional expectation (4.11).

An advantage of this reconstruction is the confidence interval computed thanks to (4.12).

3.2.2 Imputation with M2GP

Imputation of missing values using the M2GP model differs. Indeed, Proposition 5.3 presents the expectation and variance of the conditional distribution. The expectation does not depend on the linear dependency between spectral bands but all p reflectances are imputed at once. The covariance retrieves the Kronecker structure induced by the model. These observations are directly linked to the uncorrelated lines and columns in the matrix-variate Gaussian distribution (Definition 2.3).

The reconstruction quality is the same using M2GP or MIMGP. Figure 5.6 compares the two models on toy data-sets by computing the normalized mean square error (5.21).

3.2.3 Imputation using the complete mixture

When the class membership is unknown, the two models allow to compute the reconstruction of missing values using the mixture of GPs. (4.13) (or (5.17) for M2GP) allows the reconstruction using the posterior probabilities to belong to all classes.

Section4.7 presents reconstructions and a comparison with state of the art non parametric smoothing techniques. As in [184], we compared the reconstruction using the mean of the conditional distribution (4.13) with the Whittaker smoother [54].

3.3 Implementation - python code

The code has been written using python and the optimization has been done using cython (see Appendix C). The choice of a second language which is compiled has been done to avoid heavy "*for*" loops. Indeed, these loops can not be rewritten on a vectorial form where python performs well (the processing of each sample within the training or validation set induces a heavy loop). This work has been done for MIMGP but it can be adapted similarly for M2GP. However, both models are using parallel implementations on the classes.

During the tests of the code, some numerical issues arose for the estimation of the mean coefficients (Section 4.2.2) and, consequently, have interfered with the convergence of the gradient descent algorithm. The latter one uses fortran "*f_min_l_bfgs_b*" function [197] on kernel's parameters. These numerical issues are discussed in Appendix A with illustrations on a toy data-set.

CHAPTER 4

JOINT SUPERVISED CLASSIFICATION AND RECONSTRUCTION OF IRREGULARLY SAMPLED SATELLITE IMAGE TIMES SERIES

The following content has been published:
A. Constantin, M. Fauvel, and S. Girard, "Joint supervised classification and reconstruction of irregularly sampled satellite image times series", *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–13, 2021, To appear. DOI: 10.1109/TGRS.2021.3076667
Appendices are reported in Section 4.9 and supplementary materials in Appendix B.

Outline

French abstract

Les récentes missions satellitaires ont donné lieu à une énorme quantité de données d'observation de la terre, dont la plupart sont disponibles gratuitement. Dans ce contexte, les séries temporelles d'images satellitaires ont été utilisées pour étudier l'utilisation et l'occupation des sols. Cependant, les séries temporelles optiques, comme celles de Sentinel-2 ou de Landsat, sont fournies avec un échantillonnage temporel irrégulier pour différents emplacements spatiaux, et les images peuvent contenir des nuages et des ombres. Ainsi, des techniques de prétraitement sont généralement nécessaires pour classer correctement ces données. L'approche proposée est capable de traiter l'échantillonnage temporel irrégulier et les données manquantes directement dans le processus de classification. Elle est basée sur les processus gaussiens et permet d'effectuer conjointement la classification de pixel ainsi que la reconstruction des séries temporelles du pixel. La complexité de la méthode est linéaire en fonction du nombre de pixels, ce qui la rend utilisable dans des scénarios à grande échelle. Les résultats expérimentaux de classification et de reconstruction montrent que la méthode ne rivalise pas encore avec les classifieurs de l'état de l'art mais produit des reconstructions qui sont robustes par rapport à la présence de nuages ou d'ombres non détectés et ne nécessite aucun prétraitement temporel.

Abstract

Recent satellite missions have led to a huge amount of earth observation data, most of them being freely available. In such a context, satellite image time series have been used to study land use and land cover information. However, optical time series, like Sentinel-2 or Landsat ones, are provided with an irregular time sampling for different spatial locations, and images may contain clouds and shadows. Thus, pre-processing techniques are usually required to properly classify such data. The proposed approach is able to deal with irregular temporal sampling and missing data directly in the classification process. It is based on Gaussian processes and allows to perform jointly the classification of the pixel labels as well as the reconstruction of the pixel time series. The method complexity scales linearly with the number of pixels, making it amenable in large scale scenarios. Experimental classification and reconstruction results show that the method does not compete yet with state of the art classifiers but yields reconstructions that are robust with respect to the presence of undetected clouds or shadows and does not require any temporal preprocessing.

Keywords: Satellite Image Time Series (SITS), Sentinel-2, classification, reconstruction, irregular sampling, Gaussian processes, Earth Observation (EO), remote sensing.

4.1 Introduction

In the last decade, the successful launching of two satellites Sentinel-2 A and B offers a unique opportunity to record, analyze and predict the evolution of the Earth's land surface. Sentinel-2 mission provides a high resolution multispectral (13 spectral bands at 10m, 20m or 60m per pixel) acquisition with a 5 day revisit cycle [52]. It is planed for long-term operational observations (more than 15 years) and optical Sentinel-2 satellite image time-series (SITS) are available to users under a free and open data policy. Such a mission provides several terabytes of worldwide data per day [163].

Thanks to their spectral content and frequent update, SITS have found applications in various scientific field: water management [10, 140], agricultural systems mapping [58, 129, 180], urban area analysis [78, 92] or ecological monitoring [57, 106]. However, such abundance of images raises new challenges in terms of large scale multi-resolution SITS processing [39]. Issues related to *Big Earth Data* were obviously explored in the remote sensing community in order to calibrate and distribute the images seamlessly [8, 97, 148, 161, 167].

Image and signal processing issues were also investigated specifically to SITS properties. Spatial differences in terms of resolution were tackled using super resolution strategies to resample all the spectral bands at the same finest spatial resolution (10m) [135, 179, 186]. Fusion with other sensors, such as Landsat [157] or Sentinel-1 [59], was also considered to complement Sentinel-2 satellite images.

Classification of land cover or land use using machine learning techniques has emerged as a major application of SITS. Deep learning was intensively studied either using only one or few temporal acquisitions with convolutional neural networks (CNN) [159, 168], or using the full temporal stack of acquisitions using recurrent neural networks (RNN) [144, 152]. Combination of RNN applied on SITS with CNN applied on SPOT-6 image was also investigated for land cover mapping over Reunion Island [12]. Temporal convolution was investigated in [138]. Yet, due to the huge amount of data to be processed when large geographical area or large temporal domain (e.g.

1 year of acquisition) are considered, lighter machine learning techniques, such as Random Forest (RF), were proposed as well, with very good results in terms of classification accuracy [66, 93]. Similarly, Dersken *et al.* have shown that, when the number of spectro-temporal measurements per pixel is high, conventional hand-crafted features and RF perform as well as CNN in terms of classification accuracy but with a drastically reduced processing time [49].

One major issue arising when dealing with large SITS, *i.e.* using several tracks or tiles[1], is related to the irregular temporal sampling. Indeed, each Sentinel-2 track is acquired on different dates, and using images from different tracks results in pixels with different acquisition times. For instance, Figure 4.1 shows the acquisition dates for three Sentinel-2 tiles located over three parts of France. Even though some dates are similar, the total number of acquisitions is different from one tile to another. Furthermore, clouds and shadows, occurring at random, are another cause of irregular temporal sampling: Dates with shadow/cloud are usually considered as missing values and are not taken into account to build the feature vector of the corresponding pixels. Hence, even pixels belonging to the same tile can be represented by feature vectors of different sizes [93].

Classical machine learning algorithms take as input a time-series of finite and constant size. Random Forest, Support Vector Machine or RNN use a vector representation of input features, *i.e.*, a time-series, or pixel, is a collection of features stacked into a p-dimensional vector and each pixel has the same number of features. Similarly, CNN use a constant size patch representation of the image. Therefore, a pre-processing step, sometimes called *missing information reconstruction*, is usually applied on the SITS to recover pixels of same size [158]. Among the various existing techniques, temporal filters, such as the Savitzky-Golay filter [117] or the best index slope extraction (BISE) [183], and parametric or non parametric curve fitting [11, 54, 95, 114] are popular due to their simplicity and efficiency. At large scale (entire metropolitan France territory), linear temporal interpolation has shown to perform very well compared to spline interpolation [93]. Linear interpolation may also be combined with non parametric smoothers [184].

Methods taking account both spatial and temporal information have provided interesting results, see for instance non-local filters [28] or deep learning based approaches [46, 194, 195]. However, due to their high computational cost, they are hardly applicable in national/continental scale settings [49]. Furthermore, they are also known as "black-box" and hardly interpretable.

Recently, Gaussian Processes (GP) have gained attention in the remote sensing community [31, 56]. GP mixed Bayesian and kernel methods to build statistical learning machines, for classification or regression problems. They have proved to be accurate and interpretable through their hyperparameters (mean and covariance functions) [68, 130, 141, 182]. In this work, a Gaussian Process model is introduced, relying on a linear projection of the mean function on a well-chosen basis, conditionally to the classes of interest. It is able to deal with irregularly sampled pixels in the learning and classification steps as well as reconstructing missing data.

The contribution of this work is three-fold:

1. The definition of a GP model that handles irregular temporal sampling (Section 4.2), without resampling all the data on a common temporal grid.
2. Jointly to the classification, the proposed model allows for the reconstruction of pixels on any temporal grid, together with confidence intervals (Section 4.3). Furthermore, the mean and covariance functions associated with each class are available for analysis.
3. The reconstruction used the GP model with optimal parameters (from the likelihood point of view) and takes into account the class membership. Rather than relying only on the reconstruction error, as with conventional missing data reconstruction techniques, the proposed method used the classification model likelihood to control the reconstruction and its subsequent smoothing level.

The remainder of this paper is organized as follows. The proposed GP model and its estimation are described in Section 4.2. The associated classification and reconstruction methods are derived in Section 4.3. Section 4.4 details the SITS datasets while the experimental set-up is presented in Section 4.5. Experimental classification and reconstruction results are provided in Sections 4.6 and 4.7 respectively. Finally, Section 4.8 concludes this paper and opens discussion on future work.

4.2 Irregularly sampled Gaussian processes model

In the following $\mathcal{S} = \{(\mathbf{y}_i, z_i)\}_{i=1}^n$ denotes a set of n independent and identically distributed (i.i.d.) random multivariate and irregularly sampled pixels from a SITS, with their associated class labels, see Table 4.1 for an overview

[1] Sentinel-2 products are available as a collection of elementary tiles of size 100×100 km, see `https://sentinel.esa.int/web/sentinel/missions/sentinel-2/data-products`.

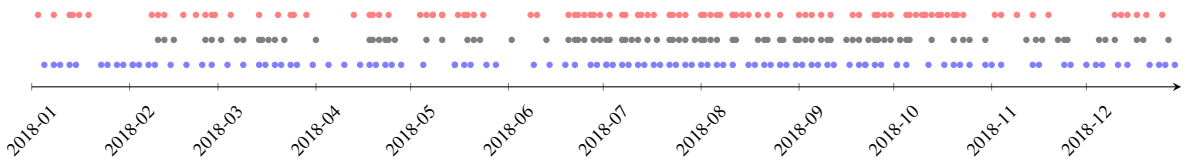


Figure 4.1: Acquisition dates for tile T31TDN (red), T31TCJ (black) and T31TGK (blue) in 2018. The location of the tiles is given in Fig. 4.5. Large temporal gaps between two dates correspond to very cloudy acquisitions: Images tagged as composed of more than 90% of cloudy pixels are not processed by the French data provider.

Table 4.1: Symbols and Notations

Symbol	Meaning
$\sim$	Distributed according to (a probability distribution).
$\mathcal{GP}$	Gaussian Process: $f \sim \mathcal{GP}(m, K)$, the function f is distributed as a GP with mean function m and covariance operator K. The function is indexed in time by t.
$\mathcal{N}_d(\boldsymbol{\mu}, \boldsymbol{\Sigma})$	d-variate Gaussian (Normal) distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.
$f^d_{\boldsymbol{\mu},\boldsymbol{\Sigma}}$	Density associated with $\mathcal{N}_d(\boldsymbol{\mu}, \boldsymbol{\Sigma})$.
$y\|x$	Random variable y conditionally on x,
$p(y\|x)$	Probability distribution of $y\|x$.
$\mathcal{S}$	A set of SITS or pixels.
n	The total number of SITS available in $\mathcal{S}$.
p	Number of wavelenghts of a SITS.
b	Band number, in $\{1, \ldots, p\}$.
$\mathcal{T}$	Temporal window where SITS are observed.
Y	$p-$ dimensional process: $\mathcal{T} \rightarrow \mathbb{R}^p$.
$\mathbf{y}_{i,b}$	ith pixel from wavelength b.
T_i	Number of temporal acquisitions associated with pixel i.
C	Number of classes.
c	Class value, in $\{1, \ldots, C\}$.
z_i	Discrete random variable associated with pixel i and representing its class membership.
π_c	Prior probability of class c.
$\boldsymbol{\alpha}, \boldsymbol{\theta}$	Model parameters.
J	Number of basis functions *i.e.* dimension of $\boldsymbol{\alpha}$.
$\|K\|$	Determinant of the matrix K.

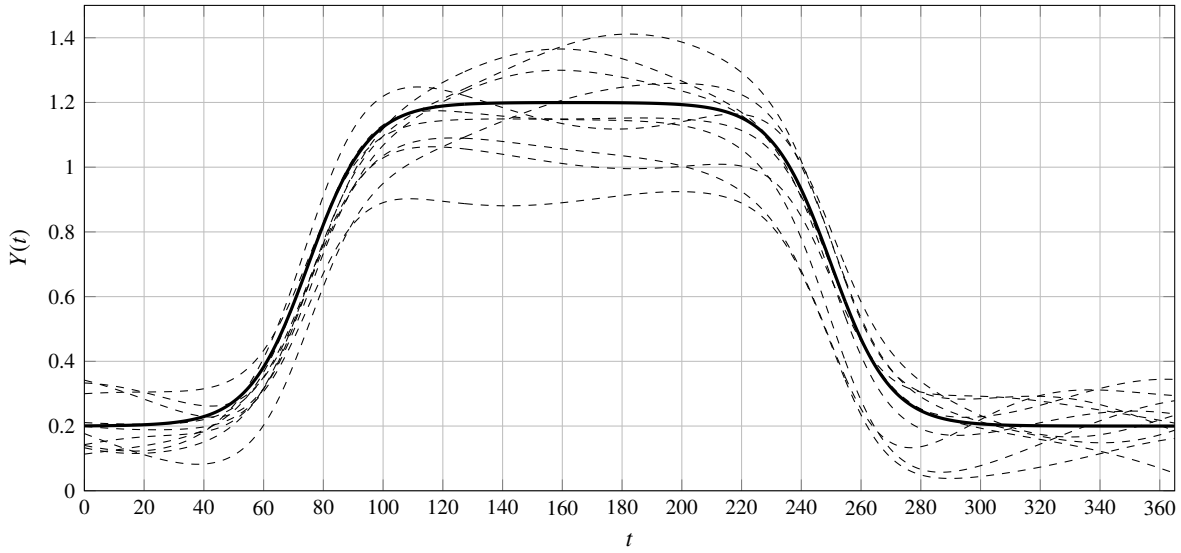


Figure 4.2: Simulated univariate Gaussian processes with squared exponential covariance function (length scale parameter set to 50). The continuous black line is the mean function and the dashed lines are the corresponding 10 realizations.

of the main notations. A pixel $\mathbf{y}$ is modeled by a random vector Y containing p random square integrable processes $\mathcal{T} \rightarrow \mathbb{R}^p$, where $\mathcal{T} = [t_{\min}, t_{\max}]$ is the time interval where the SITS are observed:

$$\begin{aligned} Y : \mathcal{T} &\rightarrow \mathbb{R}^p \\ t &\mapsto \left[Y_1(t), \ldots, Y_b(t), \ldots, Y_p(t)\right]^\top. \end{aligned}$$

This property is denoted by $Y \in L_2^p(\mathcal{T})$. The associated class z is modeled by a discrete random variable Z with possible values in $\{1, \ldots, C\}$. In the context of this work, p is the number of wavelengths and/or spectral indices, $t_{\min} = 0$ and $t_{\max} = 365$ (in day-of-year unit).

A (univariate) Gaussian process (GP) f is a stochastic process such that any finite-dimensional marginal follows a multivariate Gaussian distribution [190]. It is specified by its mean function m and covariance function K:

$$\begin{aligned} m(t_1) &= \mathbb{E}[f(t_1)], \\ K(t_1, t_2) &= \mathbb{E}[(f(t_1) - m(t_1))(f(t_2) - m(t_2))], \end{aligned}$$

with $t_1 \in \mathcal{T}$, $t_2 \in \mathcal{T}$ two given times where the SITS are observed, and we note $f \sim \mathcal{GP}(m, K)$. The covariance function K usually encodes *a priori* knowledge about the phenomena to be modeled. Figure 4.2 shows several realizations of a GP with a squared exponential covariance function (see Section 4.2.2).

In the following, a multivariate Gaussian process model is proposed to cope with the multivariate nature of SITS and their irregular temporal sampling.

4.2.1 Mixture of Independent Multivariate Gaussian Processes Model

Definition 4.1 (Mixture of Independent Multivariate Gaussian Processes)**.** The proposed model, namely Mixture of Independent Multivariate Gaussian Process (MIMGP), relies on two main assumptions:

A1 Each process Y_b, conditionally to $Z = c$, follows a Gaussian Process: $Y_b | Z = c \sim \mathcal{GP}(m_{b,c}, K_{b,c})$;

A2 Conditionally to $Z = c$, all components Y_b of Y are independent,

where $b \in \{1, \ldots, p\}$ is the spectral band and $c \in \{1, \ldots, C\}$ is the class. The first property states that each band of a pixel from a given class is a realization of a univariate GP with class and band specific mean and covariance functions. The second property is introduced for computational reasons, and is discussed in Section 4.6.2.

Let $\mathbf{y}_i$ be an observed pixel i at times $\{t_1^i, \ldots, t_{T_i}^i\}$. Its bth coordinate is represented by a vector of dimension T_i:

$$\mathbf{y}_{i,b} = [Y_b^i(t_1^i), \ldots, Y_b^i(t_{T_i}^i)]^\top.$$

By definition, conditionally to $Z = c$, this vector follows a T_i-variate Gaussian distribution

$$\mathbf{y}_{i,b}|Z_i = c \sim \mathcal{N}_{T_i}(\boldsymbol{\mu}_{i,b,c}, \boldsymbol{\Sigma}^i_{b,c}), \tag{4.1}$$

where $\mathcal{N}_{T_i}$ is the Gaussian distribution on $\mathbb{R}^{T_i}$ with vector mean $\boldsymbol{\mu}_{i,b,c} = [m_{b,c}(t^i_1), \ldots, m_{b,c}(t^i_\ell), \ldots, m_{b,c}(t^i_{T_i})]^\top$ and covariance matrix $(\boldsymbol{\Sigma}^i_{b,c})_{\ell,\ell'} = K_{b,c}(t^i_\ell, t^i_{\ell'})$.

Finally, in view of the independence assumption on the p wavelengths, $\mathbf{y}_i$ follows a product of p Gaussian densities

$$\mathbf{y}_i|Z_i = c \sim \prod_{b=1}^{p} \mathcal{N}_{T_i}(\boldsymbol{\mu}_{i,b,c}, \boldsymbol{\Sigma}^i_{b,c}). \tag{4.2}$$

4.2.2 Mean and covariance functions

The unknown parameters of the model are the mean functions $m_{b,c}$ and the covariance operators $K_{b,c}$, for each band $b \in \{1, \ldots, p\}$ and each class $c \in \{1, \ldots, C\}$. To deal with irregularly sampled observations, these parameters should be defined (and estimated) for any time t in the interval $\mathcal{T}$. To this end, projection techniques and parametric models are adopted respectively for the mean and the covariance functions. Parametric modeling of the latter ones are usual with GP, see for instance [190, Chapter 4]. However, in our specific problem, using a projection method for the mean function allows to cope nicely with the irregular temporal sampling.

Mean function

Let $\{\varphi_j\}_{j=1}^J$ be a subset of basis functions of $L_2(\mathcal{T})$. Then, $m_{b,c}$ can be written as

$$m_{b,c}(t) = \sum_{j=1}^{J} \alpha_{b,c,j}\varphi_j(t) + \varepsilon_{b,c}(t), \ t \in \mathcal{T},$$

with $\alpha_{b,c,j}$ the projection coefficient of $m_{b,c}$ on φ_j and where $\varepsilon_{b,c}$ is an approximation error term.

Any functional basis can be used, *e.g.*, Fourier, exponential, splines, ..., see for instance [84, Chapter 5]. In case of the Fourier basis, $\alpha_{b,c,j}$ represents the amplitude of the corresponding frequency in the Fourier expansion. In the following, the basis is assumed to be fixed while J, the number of basis functions, is an hyperparameter.

Introducing $\boldsymbol{\alpha}_{b,c} = [\alpha_{b,c,1}, \ldots, \alpha_{b,c,J}]^\top \in \mathbb{R}^J$ and $\mathbf{B}^i$ the $T_i \times J$ design matrix associated with pixel $\mathbf{y}_i$ defined by $(\mathbf{B}^i)_{\ell,j} = \varphi_j(t^i_\ell)$ with $\ell \in \{1, \ldots, T_i\}$, the mean vector in (4.1) can be written as

$$\boldsymbol{\mu}_{i,b,c} = \mathbf{B}^i\boldsymbol{\alpha}_{b,c}. \tag{4.3}$$

Covariance function

The covariance function is modeled using functions issued from the GP literature [190, Chapter 4]. A typical one would be a squared exponential covariance function with an additional colored noise [105] covariance function:

$$K_{b,c}(t, s) = \gamma^2_{b,c} \exp\left\{-\frac{(t-s)^2}{2h^2_{b,c}}\right\} + \sigma^2_{b,c}(t)\delta_{t,s}. \tag{4.4}$$

Introducing $\boldsymbol{\theta}_{b,c} = \{\gamma^2_{b,c}, h^2_{b,c}, \boldsymbol{\sigma}^2_{b,c}\}$, the covariance matrix in (4.1) is denoted in the following by

$$\boldsymbol{\Sigma}^i_{b,c} = \boldsymbol{\Sigma}^i(\boldsymbol{\theta}_{b,c}). \tag{4.5}$$

Parameters $\boldsymbol{\alpha}_{b,c}$ and $\boldsymbol{\theta}_{b,c}$ are estimated by maximizing the marginal log-likelihood, as explained in the next section.

4.2.3 Estimation

By plugging (4.3) and (4.5) in (4.2), it follows that

$$\mathbf{y}_i|Z_i = c \sim \prod_{b=1}^{p} \mathcal{N}_{T_i}(\mathbf{B}^i\boldsymbol{\alpha}_{b,c}, \boldsymbol{\Sigma}^i(\boldsymbol{\theta}_{b,c})). \tag{4.6}$$

Algorithm 2: Estimation of model parameters.

Input : $\mathcal{S}$,$\boldsymbol{\alpha}^0$,$\boldsymbol{\theta}^0$
Output: $\hat{\boldsymbol{\alpha}}$,$\hat{\boldsymbol{\theta}}$
for *c=1* **to** C **do**
  for *b=1* **to** p **do**
    repeat
      Update $\boldsymbol{\alpha}_{b,c}$ using (4.8);
      Do a gradient step w.r.t. $\boldsymbol{\theta}_{b,c}$ using (4.9);
    until $\ell_{b,c}(\boldsymbol{\alpha}_{b,c}, \boldsymbol{\theta}_{b,c})$ *has converged*;

Such expression is a consequence of the use of non-zero mean GPs [190, Section 2.7]. The associated negative marginal log-likelihood is given by

$$\ell(\boldsymbol{\alpha}, \boldsymbol{\theta}) = \sum_{b,c=1}^{p,C} \ell_{b,c}(\boldsymbol{\alpha}_{b,c}, \boldsymbol{\theta}_{b,c}),$$

with

$$\ell_{b,c}(\boldsymbol{\alpha}_{b,c}, \boldsymbol{\theta}_{b,c}) = \sum_{i|Z_i=c} \left[\log \left| \boldsymbol{\Sigma}^i(\boldsymbol{\theta}_{b,c}) \right| + (\mathbf{y}_{i,b} - \mathbf{B}^i \boldsymbol{\alpha}_{b,c})^\top \boldsymbol{\Sigma}^i(\boldsymbol{\theta}_{b,c})^{-1} (\mathbf{y}_{i,b} - \mathbf{B}^i \boldsymbol{\alpha}_{b,c}) \right] + \kappa,$$

where κ is a constant independent of the parameters. Since independence is assumed between each spectral component, the optimization ends up with $p \times C$ independent optimization problems:

$$(\hat{\boldsymbol{\alpha}}_{b,c}, \hat{\boldsymbol{\theta}}_{b,c}) = \arg \min_{\boldsymbol{\alpha}_{b,c}, \boldsymbol{\theta}_{b,c}} \ell_{b,c}(\boldsymbol{\alpha}_{b,c}, \boldsymbol{\theta}_{b,c}). \tag{4.7}$$

Marginal likelihood optimization is common practice in GP regression, while for binary classification, Laplace approximation is usually employed (and more computationally demanding) [190, Chapter 2 and 3]. Therefore, to the best of our knowledge, the use of marginal likelihood for multiclass classification is novel is this context.

Each sub-problem (4.7) is solved by gradient descent as usually done in likelihood optimization [190, Chapter 5]. The algorithm is based on an alternate optimization with respect to $\boldsymbol{\alpha}$ and $\boldsymbol{\theta}$ (see Algorithm 2). At iteration (k), the update rule for $\boldsymbol{\alpha}$ is given by

$$\boldsymbol{\alpha}_{b,c}^{(k+1)} = \left[\sum_{i|Z_i=c} \mathbf{B}^{i\top} \boldsymbol{\Sigma}^i(\boldsymbol{\theta}_{b,c}^{(k)})^{-1} \mathbf{B}^i \right]^{-1} \left[\sum_{i|Z_i=c} \mathbf{B}^{i\top} \boldsymbol{\Sigma}^i(\boldsymbol{\theta}_{b,c}^{(k)})^{-1} \mathbf{y}_{i,b} \right]. \tag{4.8}$$

If the design matrix has no redundancies, it can be shown that the matrix in the left hand side of (4.8) is indeed non singular provided that the number of basis functions J is smaller or equal to the total number of unique observations in the training set. A proof is given in Section I of the supplementary materials.

For $\boldsymbol{\theta}$, there is no close-form expression and a gradient step is required. The gradient is computed using the following partial derivative w.r.t. each coordinate θ_m of $\boldsymbol{\theta}$:

$$\frac{\partial}{\partial \theta_m} \ell_{b,c} = \sum_{i|Z_i=c} \operatorname{tr} \left(\left(\boldsymbol{\Sigma}^i(\boldsymbol{\theta}_{b,c}^{(k)})^{-1} - \boldsymbol{\beta}_i \boldsymbol{\beta}_i^\top \right) \frac{\partial \boldsymbol{\Sigma}^i(\boldsymbol{\theta}_{b,c}^{(k)})}{\partial \theta_m} \right) \tag{4.9}$$

with $\boldsymbol{\beta}_i = \boldsymbol{\Sigma}^i(\boldsymbol{\theta}_{b,c}^{(k)})^{-1} (\mathbf{y}_i - \mathbf{B}^i \boldsymbol{\alpha}_{b,c}^{(k+1)})$. Such an optimization procedure applied on simulated data from Figure 4.2 (*i.e.*, $C = 1$ and $p = 1$) leads to the estimation of the mean function displayed on Figure 4.3.

As a final remark, the $p \times C$ optimization problems can be solved in parallel since parameters are not shared between classes and spectral components.

4.2.4 Numerical Complexity

Usual GPs have complexity that scales in $O(n^3)$, making them unsuitable for very large scale problems. In contrast, the proposed method has a reduced complexity, that scales in $O(n(T_M^3 + J^3))$ where T_M is the maximal length of observed time-series ($T_M \geq T_i, \ \forall i \in 1, \ldots, n$). The first term comes from the inversion of $\boldsymbol{\Sigma}^i(\boldsymbol{\theta}_{b,c})$ and the second term comes from the update rule (4.8). In our application, T_M and J are much smaller than n, typically by several orders of magnitude. Furthermore, MIMGP allows for the classification of any irregularly new time-series as well as the reconstruction of the observed time-series using reconstruction techniques, as described in the following.

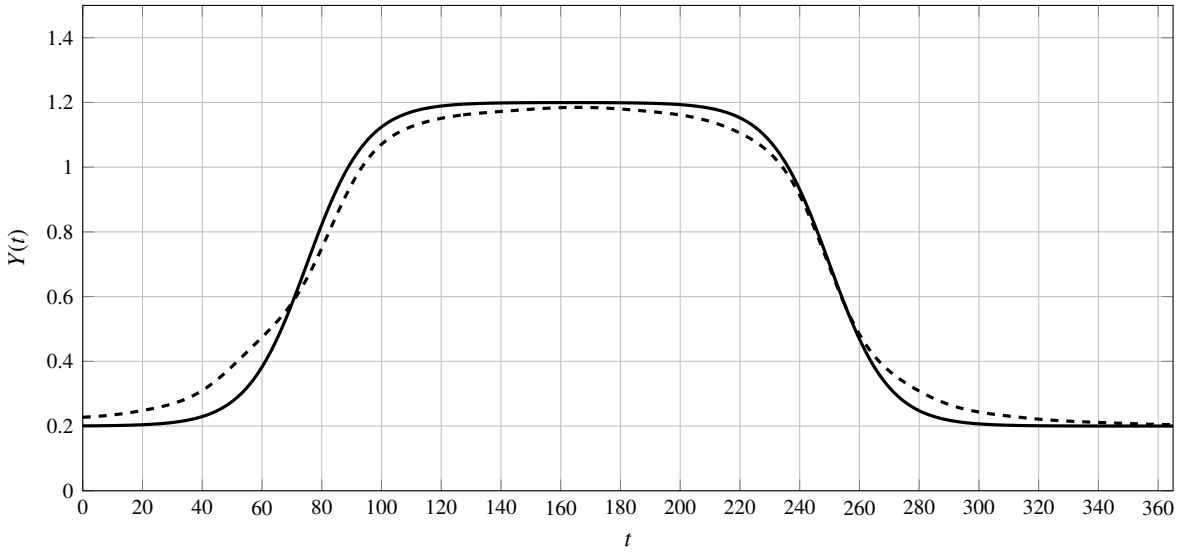


Figure 4.3: The continuous line is the mean function associated with the GP displayed in Figure 4.2 and the dashed line is the estimated one using Algorithm 2 and $J = 20$ Gaussian functions (see Section 4.5.1 for details on bases).

4.3 Classification and Reconstruction of Missing Values

Once the parameters are estimated, it is possible to classify a new pixel without any temporal resampling, as for training. In addition, MIMGP is also able to reconstruct missing values on any temporal scheme.

In the following, $\mathbf{y}_j$ denotes a new SITS observed at T_j times denoted by $\{t_1, \ldots, t_{T_j}\}$ which may not have been observed in the training set.

4.3.1 Classification of a new time-series

The *a posteriori* probability $\mathbb{P}(Z_j = c|\mathbf{y}_j)$ to belong to a class c given $\mathbf{y}_j$ is computed using the product of Gaussian densities given in (4.6) and Bayes' rule:

$$\begin{aligned}\mathbb{P}(Z_j = c|\mathbf{y}_j) &\propto \pi_c \mathbb{P}(\mathbf{y}_j|Z_j = c) \\ &\propto \pi_c \prod_{b=1}^{p} f_{T_j}(\mathbf{y}_{j,b}; \mathbf{B}^j \boldsymbol{\alpha}_{b,c}, \boldsymbol{\Sigma}(\boldsymbol{\theta}_{b,c})),\end{aligned} \tag{4.10}$$

where $f_d(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the d-variate Gaussian density with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ and π_c is the *prior* probability $\mathbb{P}(Z = c)$ of the class $c \in \{1, \ldots, C\}$.

In practice, π_c is estimated by its empirical counterpart $\hat{\pi}_c = n_c/n$ where n_c is the number of pixels assigned to class c in the set $\mathcal{S}$. Parameters $\hat{\boldsymbol{\alpha}}_{b,c}$ and $\hat{\boldsymbol{\theta}}_{b,c}$ are estimated thanks to Algorithm 2. The new time-series $\mathbf{y}_j$ is then assigned to the class of maximum posterior probability (MAP rule):

$$\hat{z}_j = \max_c \mathbb{P}(Z_j = c|\mathbf{y}_j).$$

4.3.2 Time-series reconstruction

Time-series reconstruction is achieved using conditional expectations of Gaussian distributions. Two cases are considered: Either the class membership of the considered pixel is known, or the class membership is estimated using the posterior probability.

Reconstruction when class membership is known

Let us write $Y_b^j(t^*)$ the unobserved value at wavelength b and time t^*. The following reconstruction rule, based on the conditional expectation, is considered:

$$\hat{Y}_{b,c}^j(t^*) := \mathbb{E}[Y_b^j(t^*)|\mathbf{y}_j, Z_j = c].$$

Using properties of Gaussian distributions [17, p.63] and replacing the unknown quantities by their estimated counterparts yield (see Appendix 4.9 for a proof):

$$\hat{Y}^j_{b,c}(t^*) = \mathbf{b}^*\hat{\boldsymbol{\alpha}}_{b,c} + \mathbf{k}(t^*, t^j_{1:T_j}|\hat{\boldsymbol{\theta}}_{b,c})^\top \boldsymbol{\Sigma}^j(\hat{\boldsymbol{\theta}}_{b,c})^{-1}(\mathbf{y}_{j,b} - \mathbf{B}^j\hat{\boldsymbol{\alpha}}_{b,c}) \tag{4.11}$$

and

$$\mathbb{V}[\hat{Y}^j_{b,c}(t^*)] = K(t^*, t^*|\hat{\boldsymbol{\theta}}_{b,c}) - \mathbf{k}(t^*, t^j_{1:T_j}|\hat{\boldsymbol{\theta}}_{b,c})^\top \boldsymbol{\Sigma}^j(\hat{\boldsymbol{\theta}}_{b,c})^{-1}\mathbf{k}(t^*, t^j_{1:T_j}|\hat{\boldsymbol{\theta}}_{b,c}), \tag{4.12}$$

where $\mathbf{k}(t^*, t_{1:T_j}|\hat{\boldsymbol{\theta}}_{b,c}) = [K(t^*, t_1|\hat{\boldsymbol{\theta}}_{b,c}), \ldots, K(t^*, t_{T_j}|\hat{\boldsymbol{\theta}}_{b,c})]^\top$, $\mathbf{b}^* = [\varphi_1(t^*), \ldots, \varphi_J(t^*)]$ contains the evaluation of the basis functions at time t^* and $\mathbf{B}^j$ is the $T_j \times J$ design matrix associated with $\mathbf{y}_j$.

Equations (4.11) and (4.12) provide respectively the reconstructed value and the variance of the reconstruction. They are given here for a single time t^* but similar formulas can be derived for multiple times $t^*_1, \ldots, t^*_{T^*}$. Interestingly, Equation (4.11) shows that the reconstruction is given by $\mathbf{b}^*\hat{\boldsymbol{\alpha}}_{b,c}$, the estimated mean of the GP, corrected by a value proportional to the error made at the acquisition time of pixel j. A similar remark holds for the variance: It can be interpreted as the estimated variance of the GP corrected by the variance of the process observed at the acquisition times of pixel j.

Reconstruction when class membership is unknown

The reconstruction of $\hat{Y}^j_b(t^*)$ is done as the average of the previous reconstructions $\hat{Y}^j_{b,c}(t^*)$ in each class (see (4.11)) weighted by the posterior probabilities estimated with (4.10):

$$\hat{Y}^j_b(t^*) = \sum_{c=1}^{C} \mathbb{P}(Z_j = c|\mathbf{y}_j)\hat{Y}^j_{b,c}(t^*). \tag{4.13}$$

A similar formula holds for the variance with an additional between-classes variance term:

$$\begin{aligned}\mathbb{V}[\hat{Y}^j_b(t^*)] = &\sum_{c=1}^{C} \mathbb{P}(Z_j = c|y_j)\mathbb{V}[\hat{Y}^j_{b,c}(t^*)] \\ &+ \sum_{c=1}^{C} \mathbb{P}(Z_j = c|y_j)\Big[\hat{Y}^j_{b,c}(t^*)^2 - \hat{Y}^j_b(t^*)^2\Big],\end{aligned}$$

where $\hat{Y}^j_{b,c}(t^*)$ and $\hat{Y}^j_b(t^*)$ are the reconstructions at t^* respectively when the class is known (4.11) and when the class is unknown (4.13). See Appendix 4.9 for a proof.

4.4 Sentinel-2 Satellite Image Time-Series Datasets

Three Sentinel-2 tiles of level 2A over the French metropolitan territory were downloaded from the Theia Land Data Center[2]. All available acquisitions between January 2018 and December 2018 for the two orbits of satellites Sentinel-2A and 2B were used. Figure 4.5 shows the location of the three tiles. They correspond to different climatic regions [93], with varying meteorological and topographical conditions.

Surface reflectance time-series were produced using the MAJA (Multi-sensor atmospheric correction and cloud screening-ATCOR Joint Algorithm) processing chain developed by the CNES-CESBIO and DLR [8]. It involves orthorectification, atmospheric correction, clouds and shadows detection. Spectral bands at 10m/pixel and 20m/pixel were used, for a total of 10 spectral bands. Bands at 20m/pixel were spatially up-sampled to 10m/pixel using the Orfeo ToolBox[3] [171]. Figure 4.6 displays the distribution of the number of clear dates (dates not tagged clouds, shadows or no-data in the raw time-series): It appears that the number of clear acquisitions per pixel significantly varies depending on the tile.

[2] `http://www.theia-land.fr/en/presentation/products`.

[3] Using "SuperImpose" application, see `https://www.orfeo-toolbox.org/CookBook/Applications/app_Superimpose.html`.

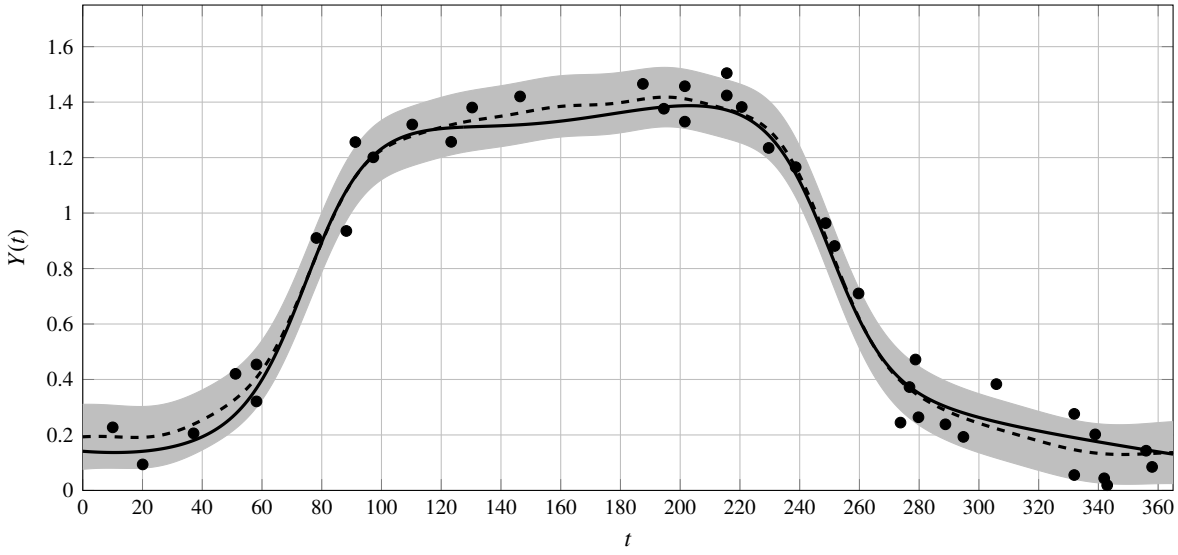


Figure 4.4: Time-series reconstruction. The black continuous line is one (continuous) realization of the GP from Figure 4.2. The black dots are the observed noisy acquisitions at several (discrete) times. The dashed line is the reconstructed time-series. The gray region displays a pointwise confidence region for the reconstruction: $\hat{Y}_b^j(t^*) \pm \sqrt{\mathbb{V}}[\hat{Y}_b^j(t^*)]$.

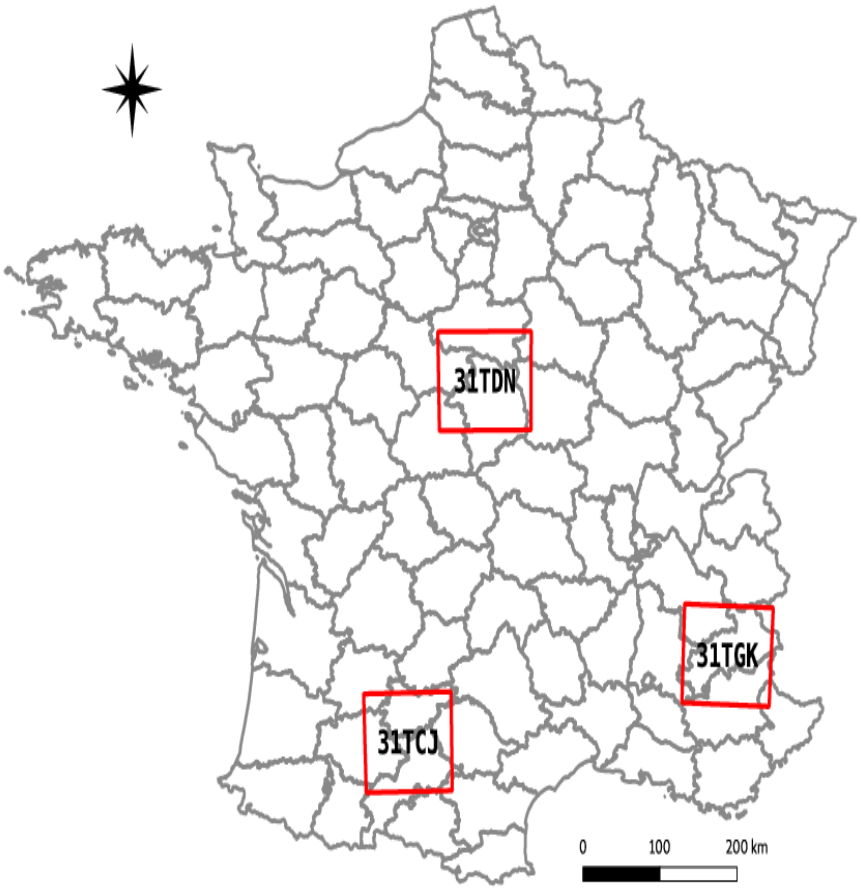


Figure 4.5: Location of the Sentinel-2 tiles. The label inside the square is the tile name.

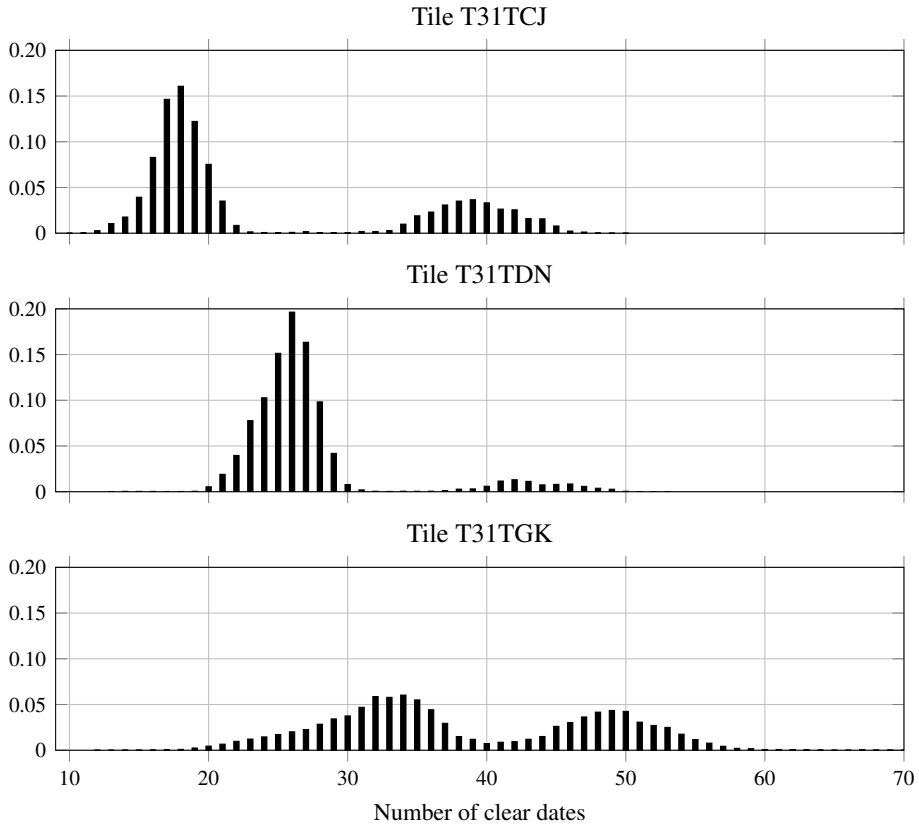


Figure 4.6: Proportion of pixels as functions of the number of clear dates.

Furthermore, the data from each tile were re-sampled and gap-filled (missing information due to clouds or shadows were reconstructed using linear interpolation) onto the same set of dates (every 5 days, starting from 2018-01-01 and ending 2018-12-27) as in [93]. Figure 4.11 shows the raw data and the re-sampled data for one random pixel from each tile in the same class. Hence, for each pixel location, three temporal informations are available after the pre-processing:

1. The raw multispectral time-series with irregular acquisition dates;

2. The mask time-series, indicating for each acquisition date the presence/absence of clouds/shadows. In our model, the presence of clouds/shadows implies that the corresponding raw spectral values are considered as missing values;

3. The re-sampled multispectral time-series with regular acquisition dates.

The reference data arise from the work of Inglada *et al.* [93]. They were extracted from freely available data source. Seventeen land cover classes were defined, ranging from artificial areas to vegetation and water bodies. Table 4.2 shows the exhaustive list of classes.

The reference data are provided as a set of spatial polygons overlapping the 3 tiles. Figure 4.7 shows an extract of these polygons. The training and validation set were constructed by stratifying pixels according to the polygons membership information: Pixels from one polygon fully belong to either the training or the validation set. Depending on the number of available referenced pixels per class, 10,000 (or less) pixels were extracted for the training and validation set, except for winter and summer crops, for which 30,000 and 40,000 pixels were extracted, respectively. Ten independent train/validation sets were generated for statistical validation. Table 4.2 shows the average number of pixels per class and per tile[4].

[4]Since a stratification w.r.t polygons is done, and polygons size can vary in terms of area, the number of pixels may vary slightly from one experiment to another. The model was trained on a total average of 178,000 pixels and was evaluated on 178,000 pixels.

Table 4.2: Land cover classes and number of extracted SITS in each tile

Class	T31TCJ	T31TDN	T31TGK
Artificial areas			
Continuous urban fabric	10,000	8,292	959
Discontinuous urban fabric	10,000	10,000	10,000
Industrial or commercial units	10,000	10,000	10,000
Road surfaces	10,000	9,906	3,664
Agricultural areas			
Winter crops	30,000	30,000	15,975
Summer crops	40,000	40,000	24,912
Forest and semi-natural areas			
Meadow	10,000	10,000	10,000
Orchards	10,000	2,775	10,000
Vines	10,000	8,719	153
Broad-leaved forest	10,000	10,000	10,000
Coniferous forest	9,957	10,000	10,000
Natural grasslands	9,939	3,022	10,000
Woody moorlands	9,972	10,000	10,000
Open spaces with little or no vegetation			
Bare rock	0	0	10,000
Beaches, dunes and sand plains	0	5,355	10,000
Glaciers and perpetual snow	0	0	10,000
Water bodies	10,000	10,000	10,000
Total	189,868	178,069	165,663

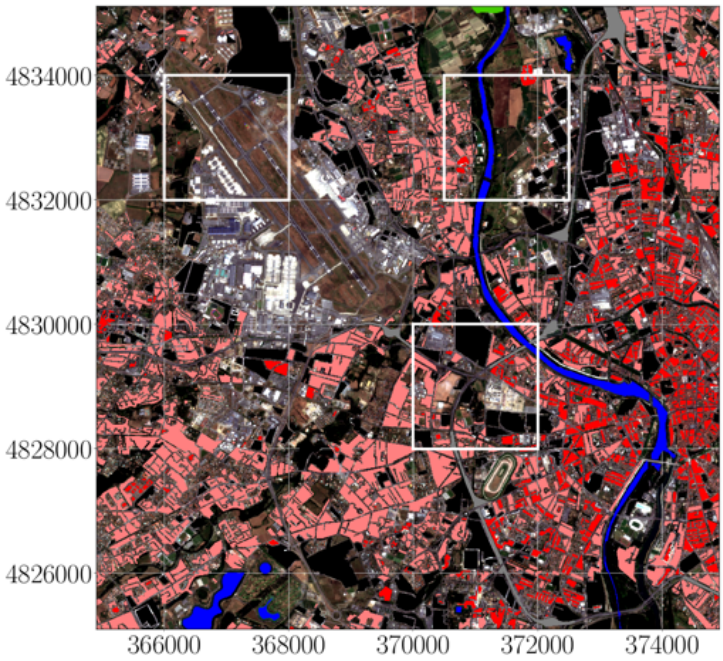


Figure 4.7: Reference data extraction where each color represents an extract of land cover classes from Table 4.2. The color code is described in Figure 4.9. The axes correspond to the geographical coordinates. The white continuous squares correspond to zoomed areas of the classification thematic maps provided in Section 4.6. The background image is an extract from Sentinel-2 optical images.

4.5 Experimental set-up

The model parameters are $\boldsymbol{\alpha}_{b,c}$ and $\boldsymbol{\theta}_{b,c}$ for each class c and each wavelength b. They are estimated using the training set of pixels. MIMGP has also some hyperparameters that are set before the learning step: The design matrix $\mathbf{B}$ with the associated number of basis functions J and the family of parametric kernels for the covariance operator. These settings are common to each class and each wavelength.

4.5.1 Functional bases

Four functional bases have been investigated: Two of them are local bases (non-zero only on subsets of $[0, \mathcal{T}]$), the *Gaussian* and *BSplines* bases, while the remaining ones are global bases, the well-known *Fourier* and *Polynomial* bases. An user-defined hyperparameter, denoted by J^*, is used to select the number of basis functions, as explained in the next paragraph. The choice of a basis setting corresponds to an assumption on the temporal behavior of the time-series: For instance, a *Fourier* basis can represent a periodic signal in the time domain.

- *Fourier*: The number of basis functions is $J = 2J^* + 1$ and
$$m(t) = \alpha_0 + \sum_{j=1}^{J^*} \Big[\alpha_j \cos(2\pi j \frac{t}{\mathcal{T}}) + \alpha_{j+J^*} \sin(2\pi(j+J)\frac{t}{\mathcal{T}})\Big].$$

- *Polynomial*: $m(t) = \alpha_0 + \sum_{j=1}^{J^*} \alpha_j t^j$, with $J = J^* + 1$.

- *Gaussian*: $m(t) = \sum_{j=1}^{J^*} \alpha_j \exp\left(-\frac{(t-t_j)^2}{d_j^2}\right)$, with $J = J^*$. The t_j, $j \in \{1, \ldots, J^*\}$ can be equidistant in $\mathcal{T}$ or chosen as quantiles of the distribution of the clean dates. The hyperparameter d_j is set such that $d_j^2 = 8|t_{j+1} - t_j|$ to ensure a sufficient overlap between two consecutive local exponential functions.

- *Cubic splines*: $m(t) = \sum_{j=0}^{J^*} \alpha_j S_j(t)$, with $J = J^* + 1$. S_j is the j^{th} bicubic spline on $[0, \mathcal{T}]$. The knots t_j are chosen as in the Gaussian case.

The invertibility condition associated with (4.8) implies that J should be smaller than 92, 100 and 103 for tiles T31TCJ, T31TDN and T31TGK, respectively.

4.5.2 Covariance function

Any convex combination of positive semi-definite kernels is a valid covariance function [190, Chapter 4]. In this work, the squared exponential covariance function added with a colored noise covariance function, as given in (4.4), is used.

The parameters $\boldsymbol{\theta}_{b,c} = \{\gamma_{b,c}^2,\ h_{b,c}^2,\ \boldsymbol{\sigma}_{b,c}^2\},\ \forall\{b,c\} \in [1,p] \times [1,C]$ are learned as described in Algorithm 2. The estimated covariance parameters provide some insights about the observed processes. In particular, the length-scale $h_{b,c}$ is related to the temporal behavior of the reflectance. For a given band b and conditionally to class c, the longer $h_{b,c}$ grows, the more correlated two distant dates are.

4.6 Supervised classification

The classification accuracy is assessed by the F_1 score, computed as the harmonic mean of the precision and recall for each class. We report the "mean F_1 score", which is the mean of F_1 scores computed on all the classes of the dataset.

First, the influence of the basis functions on the classification score is investigated. Then, comparison with other classifiers is reported and discussed. Convergence and model parameter analyses are provided in Section II of the supplementary materials.

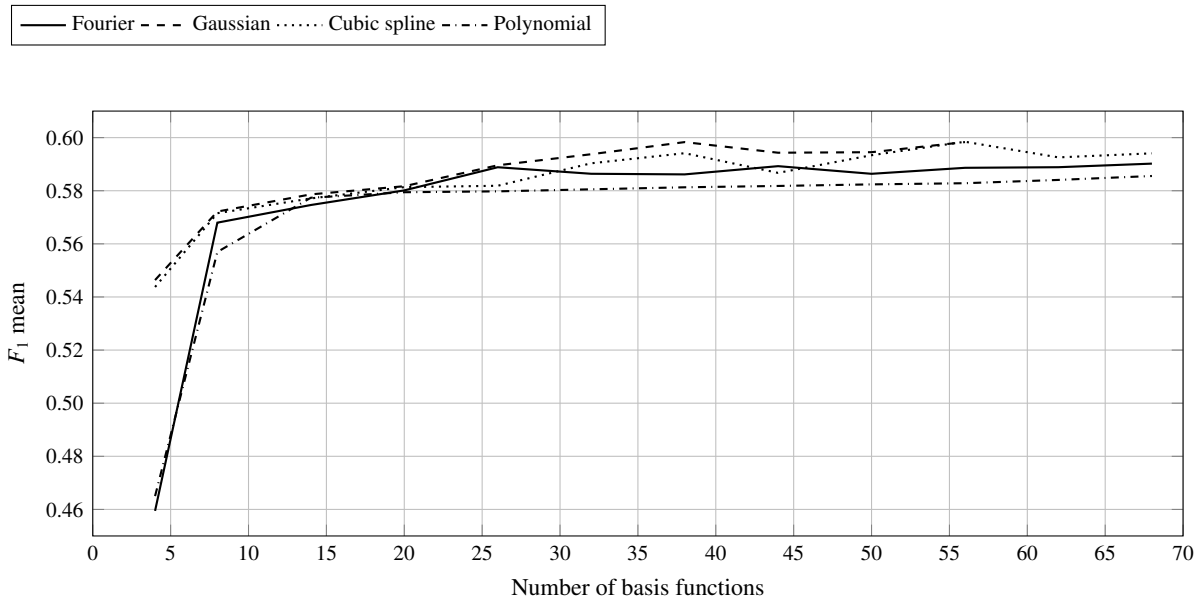


Figure 4.8: Averaged "mean F_1 scores" of the MIMGP model with the 4 bases as a function of the basis dimension. The results are reported for the tile T31TCJ.

4.6.1 Influence of the basis functions

The influence of the basis functions and its dimension are investigated. Figure 4.8 represents the averaged "mean F_1 scores" obtained on the 10 independent train/validation data sets for the tile T31TCJ. From the figure, it appears that the dimension of the basis should be larger than 9 for this tile to allow for a reasonable classification score, independently of the basis itself. Furthermore, MIMGP is robust w.r.t. basis selection since similar accuracy are obtained for all bases.

However, numerical instability has been observed for very large number of basis functions, in particular for the exponential basis. For instance, when the number of functions is greater than 50, consecutive local exponential bases overlap too much and the invertibility conditions are violated (see Section I of the supplementary materials). In practice, the Fourier basis is the most stable and was used to compare our method to other classifiers.

4.6.2 Comparison with other classifiers

The performances of MIMGP are compared to three other methods: Quadratic Discriminant Analysis (QDA) which involves a similar Gaussian assumption on regularly sampled data, a linear Support Vector Machine (SVM) fitted with a Stochastic Gradient Descent (SGD) [196], and Random Forests (RF) [26] which have shown state of the art results in large scale pixel-wise classification of SITS [49]. QDA, SVM and RF methods take the re-sampled multispectral time-series as input. The 10 spectral bands are stacked together to let these three methods account for dependence between spectral bands.

RF is used with 100 trees and a maximum depth of tree set to 25. MIMGP model is implemented with the Fourier basis and a RBF kernel with an additive colored noise which represents a total of 32 parameters for $\boldsymbol{\theta}$.

Table 4.3: Averaged "mean F_1 score" (mean(%) ± standard deviation) computed on 3 tiles. MIMGP model is parametrized by a Fourier basis with 19 parameters for T31TCJ, and 41 for T31TDN and T31TGK tiles.

	QDA	SVM	RF	MIMGP
T31TCJ	36.2 ± 2.5	70.1 ± 2.1	72.2 ± 1.6	57.6 ± 2.0
T31TDN	30.5 ± 2.8	74.8 ± 1.9	77.6 ± 1.5	65.1 ± 1.0
T31TGK	38.9 ± 2.1	62.1 ± 1.6	63.9 ± 1.6	45.5 ± 2.7

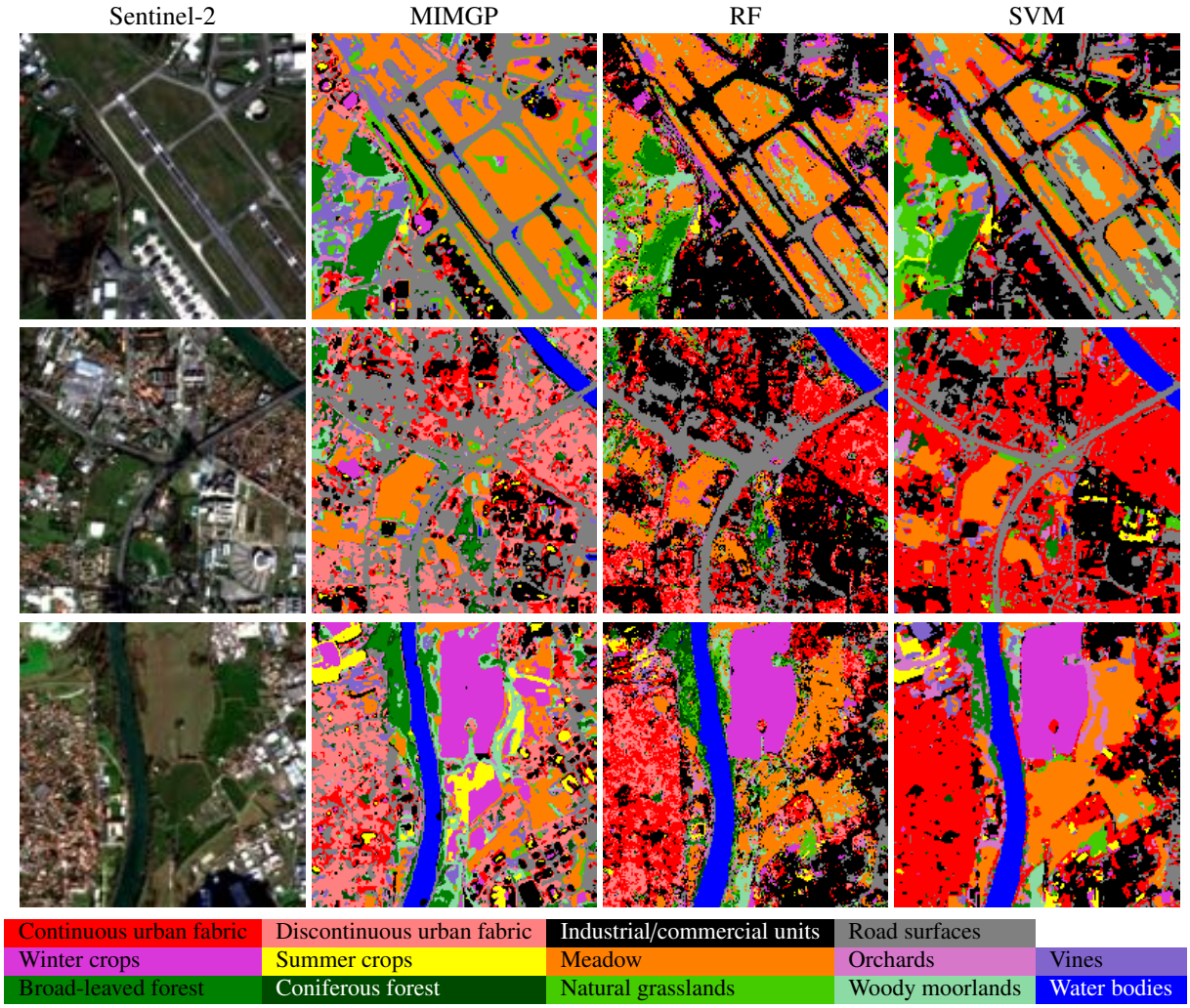


Figure 4.9: Thematic maps for 3 sites from tile T31TCJ. These sites are located as illustrated on Figure 4.7.

Table 4.3 summarizes the results for the four methods[5]. We can see that some significant improvements are achieved compared to the standard QDA classifier, even though MIMGP imposes independence between wavelengths. The QDA classifier suffers from numerical instability due to the high number of spectro-temporal features. However, MIMGP is not as accurate as the RF and the SVM classifiers applied on the reconstructed time-series.

Figure 4.9 shows three extracts of the classification map for tile T31TCJ[6]. Visually, significant differences can be observed both in terms of homogeneity and in terms of pixel-wise prediction. For instance, for the airport area, SVM predicts wrongly a lot of *natural grasslands* while MIMGP and RF predict correctly *meadow* for vegetation between runways. Overall, MIMGP and SVM seem to predict more homogeneous maps than RF. This can be clearly seen in the second row, where the city center classification maps obtained with RF contain salt and pepper noise.

Results provided in this section show that MIMGP is not at the level of state-of-the art methods such as RF or SVM. One critical point is the spectral independence assumption that is not true in practice and used here for computational purposes. Removing this constraint would help to improve the classification accuracy.

Yet, the computation of MIMGP does not require any temporal resampling processes and directly handles the raw time-series in the training and prediction steps. In comparison with standard GPs for classification, its complexity is linear w.r.t to the number of samples and cubic w.r.t to the number of temporal acquisitions, thus allowing large scale processing. Furthermore, it can also be used to reconstruct pixel time-series, as discussed in the next section.

[5] Per class F_1 scores are provided in the supplementary materials.
[6] Full classification map and extracts for the three tiles are provided in the supplementary materials.

4.7 Time-series reconstruction

Following Section 4.3.2, MIMGP is able to reconstruct missing values conditionally to a known, or predicted, class. Class mean functions and covariance operators are also obtained after the training step (for clarity, mean functions and covariance operators are provided in Section III of the supplementary materials). In the following, the quality of the reconstruction is compared with a state-of-the-art method: the Whittaker smoother [54, 184].

To assess the quality of the reconstruction, the validation set was used to ensure that reconstructed pixels were not seen during the training phase. We randomly remove one clear date from each pixel and reconstruct it with the trained MIMGP model and with the Whittaker smoother for which the regularization parameter is set using the V-curve technique for each individual pixel [62, 184]. The quality of the reconstruction is estimated using the normalized Mean Absolute Error (nMAE) computed as:

$$\text{nMAE} = \frac{\sum_{i=1}^{n_v} |Y_b^i(t^*) - \hat{Y}_b^i(t^*)|}{\sum_{i=1}^{n_v} |Y_b^i(t^*) - \overline{Y_b}|}$$

with $\overline{Y_b} = \frac{1}{n_v} \sum_{i=1}^{n_v} Y_b^i(t^*)$, n_v is the number of validation pixels. Table 4.4 summarizes the results obtained on the three tiles and for the 10 independent data sets. The scatter plots associated with nMAE results are provided in the supplementary materials.

From the table, MIGMP outperforms the Whittaker smoother in terms of nMAE for each band for tiles T31TDN and T31TGK. For tile T31TCJ, Whittaker is better for 5 bands. It is important to note that MIMGP provides better nMAE and does not need any additional fitting, apart the training step, in contrast to the Whittaker smoother that requires to be fitted for each pixel, as well its regularization parameter.

Figure 4.10 shows a part of the tile T31TGK reconstructed for the band 4 (red) on August 6, 2018. For this simulation, all the clear dates are used to reconstruct the time series for both methods. Interestingly, for the Whittaker smoother, the clouds mask is visible indicating a non continuous reconstruction of the dynamic. This artifact is not visible for the proposed method.

Table 4.4: Averaged normalized MAE (mean (%) ± Standard deviation) for each wavelength for the different tiles with the a Fourier basis of dimension $J = 19$.

S2 Band	Method	T31TCJ	T31TDN	T31TGK
Band 2	Whittaker	47.1 ± 0.5	123 ± 1.5	40.1 ± 1.2
	MIMGP	**46.9** ± 2.4	**46.5** ± 1.3	**36.3** ± 1.4
Band 3	Whittaker	45.2 ± 0.5	114 ± 1.2	42.3 ± 1.3
	MIMGP	**43.9** ± 3.7	**44.0** ± 1.6	**39.6** ± 1.4
Band 4	Whittaker	**37.7** ± 0.4	94.2 ± 1.0	43.5 ± 1.3
	MIMGP	42.7 ± 4.4	**45.2** ± 2.2	**41.3** ± 1.5
Band 5	Whittaker	**40.0** ± 0.5	99.9 ± 1.0	44.7 ± 1.3
	MIMGP	45.8 ± 0.4	**40.9** ± 2.0	**41.3** ± 2.2
Band 6	Whittaker	**32.6** ± 0.4	83.2 ± 1.0	51.1 ± 1.3
	MIMGP	38.6 ± 1.8	**46.4** ± 0.9	**46.9** ± 2.6
Band 7	Whittaker	30.0 ± 0.3	73.5 ± 0.9	51.1 ± 1.2
	MIMGP	**24.7** ± 2.7	**45.1** ± 0.9	**41.8** ± 1.1
Band 8	Whittaker	**32.6** ± 0.4	71.8 ± 0.9	52.9 ± 1.2
	MIMGP	33.1 ± 1.9	**44.6** ± 2.3	**42.9** ± 1.1
Band 8A	Whittaker	**29.4** ± 0.3	69.3 ± 0.9	51.9 ± 1.2
	MIMGP	33.2 ± 3.4	**43.8** ± 1.5	**42.6** ± 1.2
Band 11	Whittaker	**36.1** ± 0.5	53.1 ± 0.7	46.9 ± 0.6
	MIMGP	37.0 ± 2.0	**38.3** ± 1.7	**43.0** ± 1.3
Band 12	Whittaker	**34.1** ± 0.4	46.0 ± 0.6	51.3 ± 0.7
	MIMGP	39.5 ± 2.6	**38.7** ± 3.3	**46.6** ± 2.6

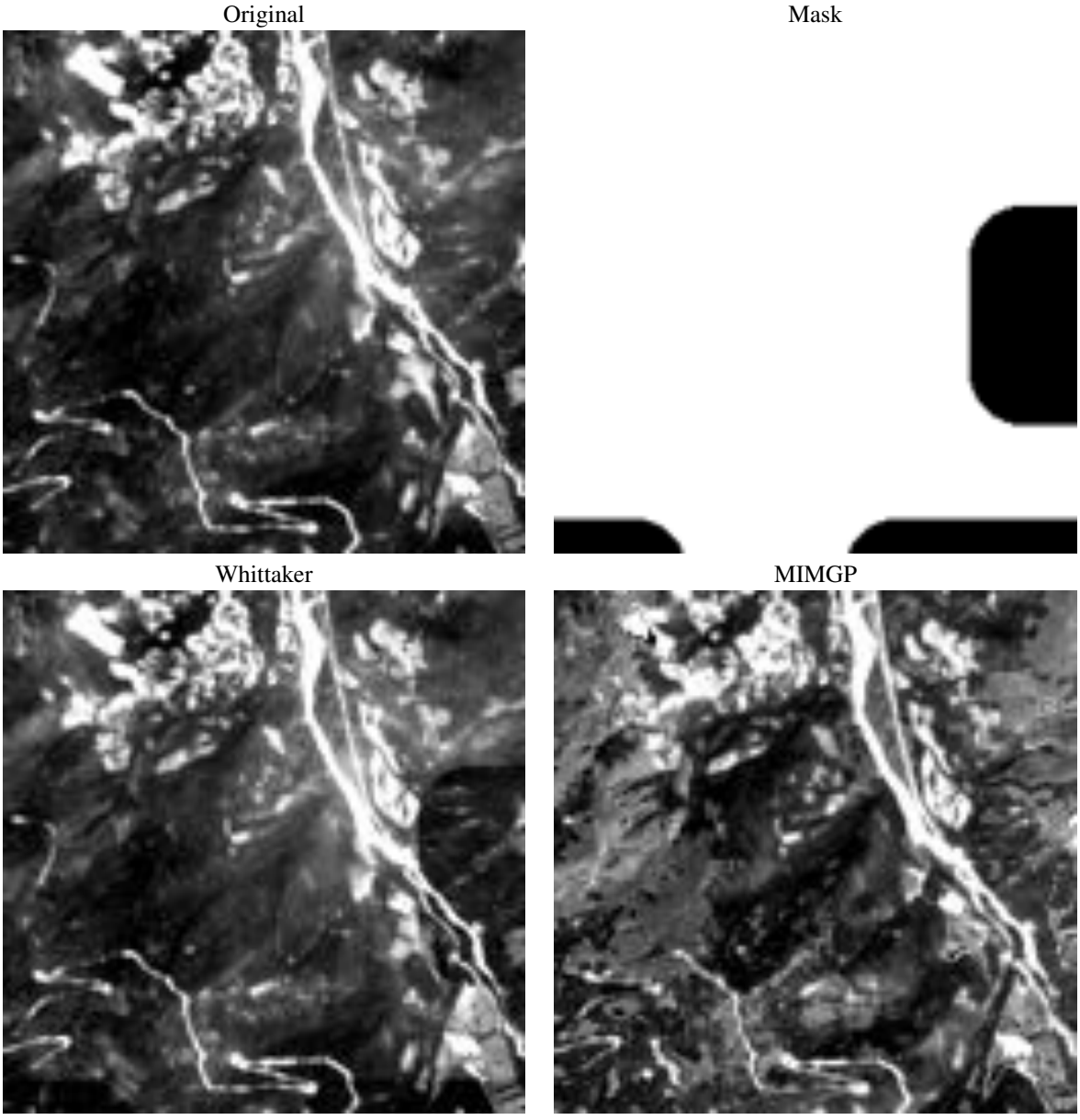


Figure 4.10: Image reconstruction from a site in tile T31TGK. This site is located as illustrated on Figure 5 of the supplementary materials. The top left panel is the original spectral band 4 (red) on August 6, 2018. The top right panel corresponds to the mask: white color for a clear pixel and black color for pixel with clouds/saturation as detected by the pre-processing chain. Bottom left and right are the reconstruction with the Whittaker smoother and the MIMGP, respectively.

Figure 4.11 illustrates the red wavelength (R) for two pixels of the same class (summer crops). They were selected to enlighten the robustness of the proposed method to an inaccurate cloud mask file. The first pixel has an undetected cloud for the third temporal acquisition (blue point in the upper Figure 4.11). A clear drop of the reflectance can be seen in the gap-filled infra-red (dashed red line). The reconstructed reflectance for MIMGP (black line) and for Whittaker smoother (black dotted line) do not exhibit such a drop in the reflectance. Similar comments can be done for the second pixel of the lower Figure 4.11 which has a undetected saturation (pixel value equal to zero). Other wavelengths are reported in the supplementary materials.

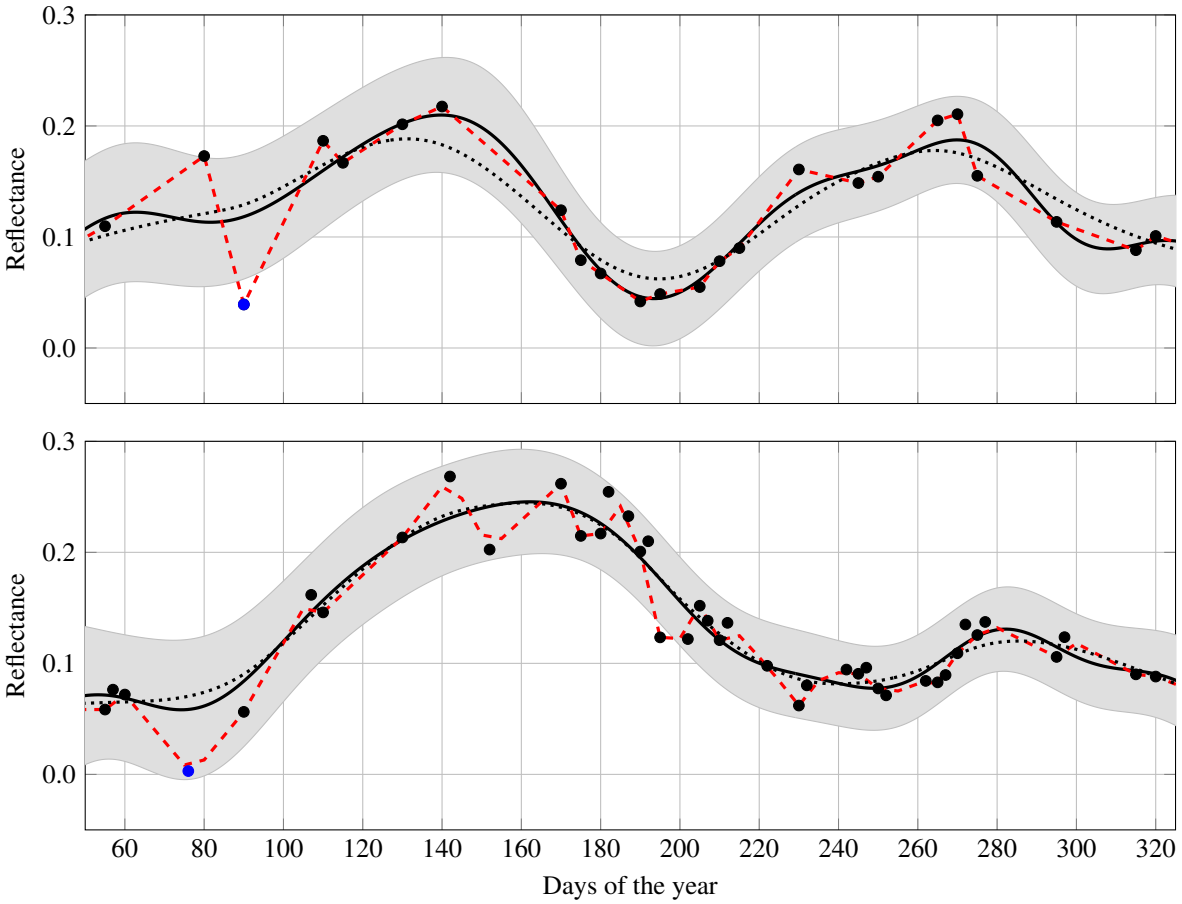


Figure 4.11: Time-series reconstruction where top and bottom figures represent the red reflectance for two different pixels. Dot points are the original values from the irregularly sampled time-series. Blue points are non-detected clouds and saturation by the satellite data preprocessing chain, for the top and bottom figure, respectively. The black continuous line is the conditional reconstruction of the signal and the gray region corresponds to the 95% confidence interval. Each day of the year was reconstructed with MIMGP. The red dashed line represents the linear interpolation taking into account all the dates tagged "clear", the black dotted line represents the smoothed signal using Whittaker smoother.

4.8 Conclusion

This work introduces a novel approach to jointly classify and reconstruct irregularly sampled multidimensional time-series. The proposed model, namely MIMGP, only involves a small number of parameters and is scalable to large datasets. The performances of the method were illustrated on a full-year of Sentinel-2 satellite image time-series dataset from 2018 involving a high number of temporal acquisitions. A by-product of MIMGP is also to infer a confidence interval on the reconstruction.

The reconstruction has shown a good behavior on noisy pixels, while the model does not compete with state of the art classifiers such as Random Forest or SVM. One limitation comes from the independence hypothesis between spectral wavelengths, which is not true in practice. Therefore, our future work will be dedicated to the

definition of multivariate Gaussian Processes able to take into account the correlation between wavelengths.

Another work in progress concerns the estimation of the mean function. MIMGP is able to handle the irregular sampling caused by multiple tiles, but the mean function is assumed to be spatially stationary. Rather than using a linear combination of fixed basis functions, a more complex approximating function depending on the geographical location could be used.

Finally, this work can be extended in a number of directions that could be of interest in the remote sensing field. A first extension would be the use of the MIMGP model to perform classification in the unsupervised case thanks to an EM-like algorithm. A second direction of future research would be the joint use of both Sentinel-2 and Sentinel-1 time-series thanks to the MIMGP model which allows for arbitrary temporal sampling in each spectral band.

4.9 Appendix - Time-series reconstruction

Let us first consider the case where the class of the missing value is known to be c. In such a situation, the reconstruction can be achieved using the conditional expectation:

$$\hat{Y}^j_{b,c}(t^*) := \mathbb{E}[Y^j_b(t^*)|\mathbf{y}_j, Z_j = c].$$

The independence assumption A2 then yields

$$\mathbb{E}[Y^j_b(t^*)|\mathbf{y}_j, Z_j = c] = \mathbb{E}[Y^j_b(t^*)|\mathbf{y}_{j,b}, Z_j = c].$$

Besides, conditionally on $Z_i = c$, one has

$$\begin{pmatrix} Y^j_b(t^*) \\ \mathbf{y}_{j,b} \end{pmatrix} \sim \mathcal{N}_{T_j+1}\left(\begin{bmatrix} \mathbf{b}^*\boldsymbol{\alpha}_{b,c} \\ \mathbf{B}^j\boldsymbol{\alpha}_{b,c} \end{bmatrix}, \begin{bmatrix} K(t^*,t^*|\boldsymbol{\theta}_{b,c}) & \mathbf{k}(t^*, t^j_{1:T_j}|\boldsymbol{\theta}_{b,c})^\top \\ \mathbf{k}(t^*, t^j_{1:T_j}|\boldsymbol{\theta}_{b,c}) & \boldsymbol{\Sigma}^j(\boldsymbol{\theta}_{b,c}) \end{bmatrix}\right),$$

where $\mathbf{k}(t^*, t_{1:T_j}|\boldsymbol{\theta}_{b,c}) = [K(t^*,t_1|\boldsymbol{\theta}_{b,c}), \ldots, K(t^*,t_{T_j}|\boldsymbol{\theta}_{b,c})]^\top$, $\mathbf{b}^* = [\varphi_1(t^*), \ldots, \varphi_J(t^*)]$ and $\mathbf{B}^j$ is the $T_j \times J$ design matrix associated with the time-series $\mathbf{y}_j$ such that $(\mathbf{B}^j)_{\ell,k} = \varphi_k(t^j_\ell)$ for all $(\ell,k) \in \{1,\ldots,T_j\} \times \{1,\ldots,J\}$. From classical properties of conditional Gaussian distributions (see for instance [17, p.63]), it follows that

$$\mathbb{E}(Y^j_b(t^*)|\mathbf{y}_{j,b}; Z_j = c) = \mathbf{b}^*\boldsymbol{\alpha}_{b,c}, +\mathbf{k}(t^*, t^j_{1:T_j}|\boldsymbol{\theta}_{b,c})\boldsymbol{\Sigma}^j(\boldsymbol{\theta}_{b,c})^{-1}(\mathbf{y}_{i,b} - \mathbf{B}^j\boldsymbol{\alpha}_{b,c}),$$

$$\mathbb{V}(Y^j_b(t^*)|\mathbf{y}_{j,b}; Z_j = c) = K(t^*,t^*|\boldsymbol{\theta}_{b,c}) - \mathbf{k}(\boldsymbol{\theta}_{b,c}|t^*, t^j_{1:T_j})\boldsymbol{\Sigma}^j(\boldsymbol{\theta}_{b,c})^{-1}\mathbf{k}(t^*, t^j_{1:T_j}|t\boldsymbol{\theta}_{b,c})^\top.$$

Finally, replacing the unknown quantities by their estimated counterparts yields the desired results.

Second, when the class is unknown, the reconstruction rule is given by

$$\hat{Y}^j_b(t^*) = \mathbb{E}[Y^j_b(t^*)|\mathbf{y}_j].$$

Since

$$\mathbb{E}[Y^j_b(t^*)|\mathbf{y}_j] = \sum_{c=1}^{C} \mathbb{P}(Z_j = c|\mathbf{y}_j)\mathbb{E}[Y^j_b(t^*)|\mathbf{y}_j, Z_j = c],$$

it straightforwardly follows that

$$\hat{Y}^j_b(t^*) = \sum_{c=1}^{C} \mathbb{P}(Z_j = c|\mathbf{y}_j)\hat{Y}^j_{b,c}(t^*),$$

where the reconstructed value $\hat{Y}^j_{b,c}(t^*)$ is provided in (4.11) while the posterior probabilities are given by (4.10). The variance is obtained by a similar calculation:

$$\begin{aligned} \mathbb{V}(Y^j_b(t^*)) &= \mathbb{E}[Y^j_b(t^*)^2|\mathbf{y}_j] - \mathbb{E}[Y^j_b(t^*)|\mathbf{y}_j]^2, \\ &= \sum_{c=1}^{C} \mathbb{P}(Z_j = c|y_j)\mathbb{E}[Y^j_b(t^*)^2|\mathbf{y}_j, Z_j = c] - \hat{Y}^j_b(t^*)^2. \end{aligned}$$

Besides, remarking that

$$\begin{aligned} \mathbb{E}[Y^j_b(t^*)|\mathbf{y}_j, Z_j = c] &= \hat{Y}^j_{b,c}(t^*), \\ \mathbb{E}[Y^j_b(t^*)^2|\mathbf{y}_j, Z_j = c] &= \mathbb{V}(Y^j_{b,c}(t^*)|\mathbf{y}_j, Z_j = c) + \mathbb{E}[Y^j_{b,c}(t^*)|\mathbf{y}_j, Z_j = c]^2, \end{aligned}$$

it follows that

$$\mathbb{V}(\hat{Y}_b^j(t^*)) = \sum_{c=1}^{C} \mathbb{P}(Z_j = c|y_j)\mathbb{V}(\hat{Y}_{b,c}^j(t^*)|\mathbf{y}_j, Z_j = c) + \sum_{c=1}^{C} \mathbb{P}(Z_j = c|y_j)\hat{Y}_{b,c}^j(t^*)^2 - \hat{Y}_b^j(t^*)^2,$$

and the result is proved. Let us highlight that these derivations were conducted when a single time t^* is considered. Similar calculations can be achieved when reconstructed simultaneously several values. In such a case, this estimation procedure provides the covariance matrix of the reconstructed values at each time.

Acknowledgment

The authors thank the reviewers for their many constructive and helpful comments. The authors would like to thank S. Iovleff for his support and advices during the construction of the model. The authors would also like to thank Y. Tanguy for his help when using the CNES computational resources to run the experiments presented in this paper.

CHAPTER 5

MIXTURE OF MULTIVARIATE GAUSSIAN PROCESSES FOR CLASSIFICATION OF IRREGULARLY SAMPLED SATELLITE IMAGE TIME-SERIES

The following content has been submitted for publication:
A. Constantin, M. Fauvel, and S. Girard, "Mixture of multivariate gaussian processes for classification of irregularly sampled satellite image time-series", working paper or preprint, 2021
Appendices are reported in Section 5.8.

Outline

French abstract

Cet article étudie la classification des séries temporelles d'images satellitaires (SITS) échantillonnées de manière irrégulière. Un modèle de mélange de processus gaussiens multivariés est proposé pour prendre en compte l'échantillonnage irrégulier et la nature multivariée des séries temporelles. La corrélation spectrale et temporelle sont prises en compte en utilisant une structure de Kronecker pour l'opérateur de covariance du processus gaussien. Le modèle de mélange multivarié du processus gaussien permet à la fois la classification des séries temporelles et l'imputation des valeurs manquantes. Les résultats expérimentaux sur des données simulées et réelles (SITS) illustrent l'importance de prendre en compte la corrélation spectrale pour assurer un bon comportement en termes de précision de classification et d'erreurs de reconstruction.

Abstract

The classification of irregularly sampled Satellite image time-series (SITS) is investigated in this paper. A multivariate Gaussian process mixture model is proposed to address the irregular sampling and the multivariate nature of the time-series. The spectral and temporal correlation is handled using a Kronecker structure on the covariance operator of the Gaussian process. The multivariate Gaussian process mixture model allows both for the classification of time-series and the imputation of missing values. Experimental results on simulated and real SITS data illustrate the importance of taking into account the spectral correlation to ensure a good behavior in terms of classification accuracy and reconstruction errors.

Keywords: Multivariate Gaussian processes, Classification, Multivariate imputation of missing data, Irregular sampling, Satellite Image Time Series (SITS), remote sensing.

5.1 Introduction

Satellite images availability has exponentially grown in the last decade. Thanks to free data access policy, optical satellite image time-series (SITS) such as *Landsat* or *Sentinel-2*, offer an unique opportunity to monitor the state and evolution of our living planet. Therefore, SITS have found many applications in ecological monitoring [106, 57], meteorology [112, 14] or agricultural system mapping [180, 129, 58], among others.

SITS are characterized by their spatial and spectral resolutions, and their revisit cycle. The spatial resolution corresponds to the size of a pixel on the ground, *e.g.*, a square of 10 meters while the spectral resolution is related to the number of wavelengths collected by the sensor, ranging typically in the visible and near infra-red part of the spectrum [122]. The revisit cycle stands for the time between two acquisitions over the same location: SITS have constant and short (*e.g.* few days) revisit time. Hence, for a given temporal period, a pixel is the collection of spectral measurements made at different times over the same location.

These properties lead to an unprecedented amount of numerical data, for which statistical methods are used to extract meaningful information such as land cover, crops yields ... For a pixel-wise based analysis, the predictor variables are multivariate time-series and the output variables represent the information to be extracted. While spatial independence is usually assumed [103], temporal and spectral correlations are commonly taken into account in statistical models [115].

However, external random meteorological factors interfere with the availability of the acquired data at the pixel scale. Indeed, as displayed in Fig. 5.1, shadows and clouds result in missing data in the time-series. Furthermore, orbital trajectory generates an irregular temporal sampling: Even though the acquisition scheme is regular, acquisition days are different for pixels located at different places [93]. As such, each pixel of the SITS has its own size in the temporal domain: Fig. 5.2 illustrates the irregular temporal sampling on the data under consideration in this paper.

Specific models are thus required to properly analyze such time-series, as described in Section 5.2. Conventional approaches usually start by resampling the data onto a common temporal grid. In this work, we aim at analyzing irregularly sampled multidimensional SITS without any temporal resampling. In particular, the supervised pixel classification task is considered, *i.e.* the assignment of each pixel of the time-series to a predefined class.

To this end, a mixture of multivariate Gaussian Processes is proposed. A linear dependence model is assumed between the spectral variables leading to a separable covariance function in time and spectral domains. The resulting model provides statistical information on the underlying process for each class (mean and covariance functions) and scales linearly w.r.t. the number of samples.

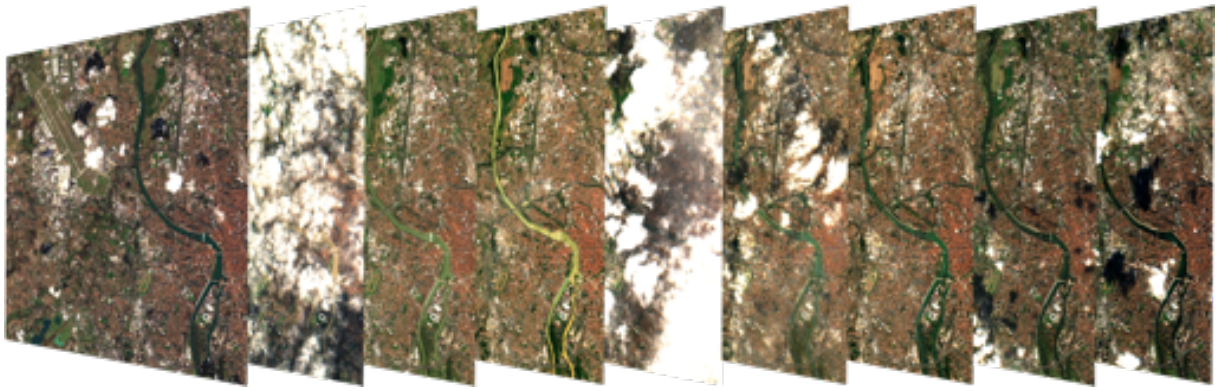


Figure 5.1: True color Sentinel-2 satellite image time-series. Data were acquired in 2018 at different time steps over the area of Toulouse, France (images were downloaded from *Theia Land Data Center*: `http://www.theia-land.fr/en/presentation/products`).

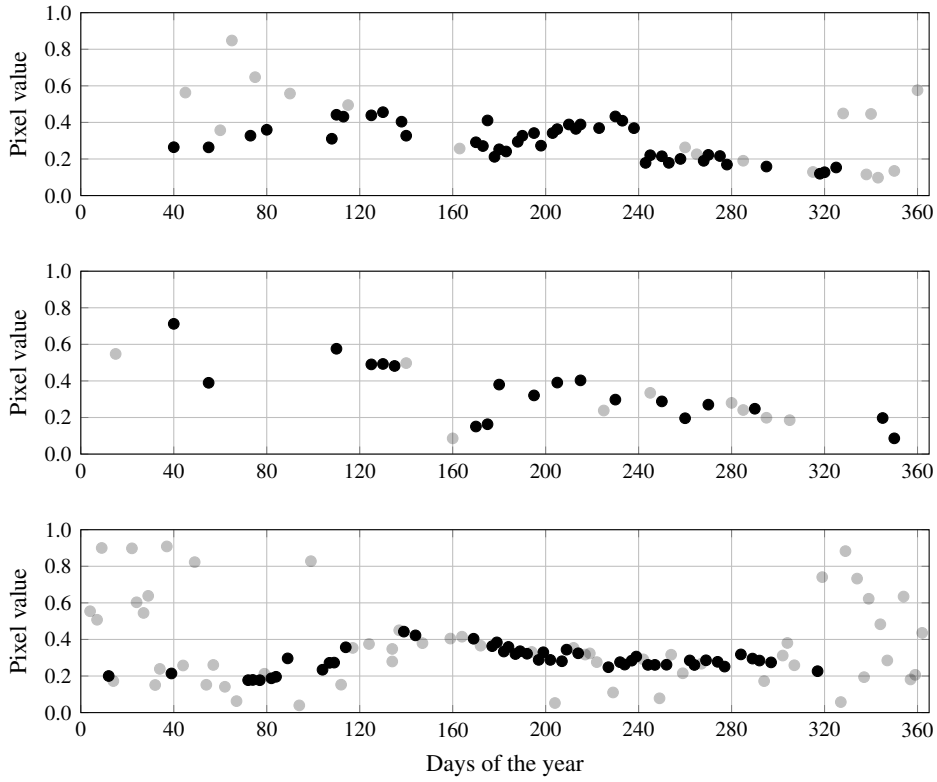


Figure 5.2: Illustration of the irregular temporal sampling for the SITS used in this work. Three time-series at different locations for one spectral band are reported: A black dot indicates that the pixel is clear (no shadow or cloud) at the considered time, and a light-gray dot indicates that the pixel has been tagged as clouds or shadows by the data provider.

The remainder of this paper is organized as follows. Section 5.2 reviews the state-of-the-art on classification with missing data and Gaussian processes. The statistical model is introduced in Section 5.3 while inference aspects are discussed in Section 5.4 including the estimation of the model parameters, the supervised classification, and the imputation of missing values. These statistical procedures are validated on simulated data in Section 5.5. Section 5.6 is dedicated to the application of our methodology to the classification of SITS from a Sentinel-2 data-set. Section 5.7 concludes with a discussion on possible extensions of this work.

5.2 Related Work

This section briefly reviews state-of-the-art methods for model-based classification, classification dealing with missing values and classification with Gaussian processes.

5.2.1 Supervised model-based classification

Supervised model-based classification (also referred to as model-based discriminant analysis) starts from a training set of n independent realizations from a random pair $(\mathbf{Y}, Z) \in E \times \{1, \dots, C\}$ and assumes that the conditional distribution of $\mathbf{y}|Z = c$ belongs to some parametric family: $p(\mathbf{y}|Z = c) = p_c(\mathbf{y}; \boldsymbol{\theta}_c)$, for all $c \in \{1, \dots, C\}$ and $\mathbf{y} \in E$, where E is an arbitrary space. Letting $\pi_c = \mathbb{P}(Z = c)$, the marginal distribution of $\mathbf{Y}$ can be written as a finite mixture

$$p(\mathbf{y}) = \sum_{c=1}^{C} \pi_c p_c(\mathbf{y}; \boldsymbol{\theta}_c),$$

whose parameters can be estimated by the maximum likelihood principle. A non-labeled observation can then be classified thanks to the Maximum a posteriori (MAP) criteria:

$$\hat{c} = \underset{c \in \{1,\dots,C\}}{\arg\max}\, p(Z = c|\mathbf{y}) = \underset{c \in \{1,\dots,C\}}{\arg\max}\, \pi_c p_c(\mathbf{y}; \boldsymbol{\theta}_c),$$

thanks to Bayes' rule. When $E = \mathbb{R}^q$, the multivariate Gaussian distribution is often adopted for $p_c(\mathbf{y}; \boldsymbol{\theta}_c)$ and gives rise to the well-known Quadratic discriminant analysis (QDA) method. We refer to [84, Section 4.3] for a discussion on the advantages and drawbacks of QDA and for possible extensions. Recent studies extend the model-based classification framework to non-Gaussian distributions such as the skew-normal distributions [175, 35] to deal with asymmetric data, or t-distributions [4, 132] to deal with outliers. We refer to [22, Chapter 9] for an in-depth review. The case $E = \mathbb{R}^q$ also encompasses the situation of discretized time-series on a common grid. Specific models can be then defined, as in [142] for temporal signatures.

If E is discrete, including for example the case of categorical data, extensions focus on the multinomial [34] or the Dirichlet [21] distributions. In the case of ordinal data, other extensions are proposed using a dedicated model of the process generating data [15]. Finally, when E is more complex, *e.g.* infinite dimensional, non-parametric techniques are used. Kernel methods are probably the most popular non-parametric techniques in this situation [88]. Recall that a kernel is a positive definite function that corresponds to a dot product in a feature space. It allows for the construction of non-linear and non-parametric classifiers on E without computing explicitly the feature space. Kernels can be defined, for instance, on strings [113], graphs [101], vector-valued functions [3, 61], or combinations of several data types [23].

5.2.2 Classification with missing data

When dealing with remote sensing data, *i.e.* spatial-spectro temporal data such as SITS, handling missing values [158] is a recurrent problem. Classification dealing with missing data occurs when some inputs in the training set are incomplete, *i.e.* the number of available coordinates in $\mathbf{y}$ can be different from one sample to another, see [153, 64, 9] for reviews.

Three main approaches can be found in the literature. A first solution is to impute missing values before the classification itself. The pre-processing gives rise to a training set with observations re-sampled on a common grid that can be considered as vectors in a finite space $E = \mathbb{R}^q$, opening the door to classical model-based classification methods. We refer to [109] for a review on imputation techniques. For SITS, [93] used such two-stage approaches on Sentinel-2 SITS where a linear interpolation was applied before performing the classification with a Random Forests classifier. Yet, by applying imputation techniques without any connection to the actual processing, propagated errors from the interpolation may degrade the results.

Alternative solutions are based on functional data analysis [146]. Each observation is interpreted as a sample from a random function. As such, it can be approximated by an expansion on some basis functions. The statistical analysis is then performed on the random vectors of coefficients, see [154] for an application to clustering. Nonparametric smoothing techniques may also adopted, see [60, Chapter 8] for an overview.

Finally, purely non-parametric methods can also be implemented by defining an appropriate dissimilarity measure between samples of varying size. In the context of time-series, Dynamic time warping (DTW) [40] is one of the most popular algorithms. It computes an optimal match between two vectors with different lengths. This map defines a dissimilarity that can be used for comparison in order to cluster samples into multiple groups.

5.2.3 Classification with Gaussian processes

A recent approach for supervised classification is based on the use of Gaussian processes (GPs) in a Bayesian framework. More specifically, Gaussian processes are used as prior distributions on the regression function linking the label Z to the explanatory variable $\mathbf{X}$. In the binary classification case, the conditional Bernoulli distribution of Z is defined through a logit transformation: $\text{logit}(p(Z = 1|\mathbf{X} = \mathbf{x})) := f(\mathbf{x})$ where $f(\mathbf{x})$ is a centered Gaussian process. The considered prior Gaussian process is, most of the time, one-dimensional. Extensions to the multi-dimensional case include the so-called multi-tasks or multi-outputs GP models, see [18, 3]. Finally, some recent works focus on non Gaussian processes such as Student-t processes which have gain attention over the past years [156, 37].

The discrete nature of Z makes the exact inference of model parameters infeasible. To overcome this difficulty, several techniques have been proposed, including the Laplace approximation, or through the expectation-propagation algorithm [133]. Such approaches rely on the inversion of a $n \times n$ covariance matrix and thus scale in $O(n^3)$ which makes the inference computationally demanding for large data sets. Scalable GPs were proposed to overcome this vexing effect, using for instance variational inference as in [86]. We refer to [111] for a review on this topic. In the next Section, we define a mixture of multivariate Gaussian processes which can be used for classification or imputation tasks without resort to approximate inference techniques.

5.3 Mixture of Multivariate Gaussian processes

The mixture of multivariate Gaussian processes model is introduced in Paragraph 5.3.1 and some associated properties are derived in Paragraph 5.3.2.

5.3.1 Model

Let $\mathcal{T}$ be a compact subset of $\mathbb{R}$, throughout this document, we denote by $\mathcal{GP}_1(0, K)$ a continuous univariate centered Gaussian process on $\mathcal{T}$ with covariance function $K : \mathcal{T}^2 \to \mathbb{R}$. Recall that, by definition, $W \sim \mathcal{GP}_1(0, K)$ implies that, for all $(t_1, \ldots, t_q) \in \mathcal{T}^q$, the random vector $(W(t_1), \ldots, W(t_q))^\top$ follows a multivariate centered Gaussian distribution $\mathcal{N}_q(\mathbf{0}, \mathbf{\Sigma})$ such that $\mathbf{\Sigma}_{i,j} = K(t_i, t_j)$, see for instance [190].

For all $p > 0$, let us similarly denote by $\mathcal{IGP}_p(0, K)$ a $p-$ dimensional, independent, centered Gaussian process defined as

$$\mathbf{W} = (W_1, \ldots, W_p)^\top \sim \mathcal{IGP}_p(0, K) \text{ if and only if } \begin{cases} W_b \sim \mathcal{GP}_1(0, K), \ \forall b \in \{1, \ldots, p\}, \\ W_b \perp\!\!\!\perp W_{b'}, \ \forall b \neq b' \in \{1, \ldots, p\}^2, \end{cases}$$

where $\perp\!\!\!\perp$ stands for independence. The above defined multivariate Gaussian processes are the building blocks to define more general multivariate Gaussian processes denoted by $\mathcal{MGP}_p(\mathbf{m}, K, \mathbf{A})$ where $\mathbf{m} : \mathcal{T} \to \mathbb{R}^p$ is the mean function, $K : \mathcal{T}^2 \to \mathbb{R}$ is the covariance operator and $\mathbf{A}$ a non-singular $p \times p$ matrix:

$$\mathbf{Y} \sim \mathcal{MGP}_p(\mathbf{m}, K, \mathbf{A}) \text{ if and only if } \mathbf{Y} = \mathbf{AW} + \mathbf{m} \text{ with } \mathbf{W} \sim \mathcal{IGP}_p(0, K). \tag{5.1}$$

Let us remark that model (5.1) is not identifiable without additional constraints. Indeed, $\mathcal{MGP}_p(\mathbf{m}, K, \mathbf{A})$ and $\mathcal{MGP}_p(\mathbf{m}, \lambda K, \mathbf{A}/\sqrt{\lambda})$ yield the same process for all $\lambda > 0$. This issue is discussed in further details in Section 5.4, see also the next paragraph for some basic properties of multivariate Gaussian processes defined in (5.1).

Introduce Z a discrete random variable taking its values in $c \in \{1, \ldots, C\}$ with $\pi_c = \mathbb{P}(Z = c)$. The mixture of multivariate Gaussian processes (M2GP) is defined by:

$$\text{Conditionally to } Z = c, \ \mathbf{Y} \sim \mathcal{MGP}_p(\mathbf{m}_c, K_c, \mathbf{A}_c), \tag{5.2}$$

where $\mathbf{m}_c : \mathcal{T} \to \mathbb{R}^p$, $K_c : \mathcal{T}^2 \to \mathbb{R}$ and $\mathbf{A}_c$ is a non-singular $p \times p$ matrix, for all $c \in \{1, \ldots, C\}$. In the context of SITS classification, $\mathbf{Y}$ represents the (unobserved) multidimensional process and p denotes the number of spectral

bands. The particular case $\mathbf{A}_c = \mathbf{I}_p$ yields a mixture of independent Gaussian processes (MIGP) whose applications to classification have been investigated in [41]. Let us also note that multivariate Gaussian processes have already been used in the machine learning community, without formal definition though, see for instance the so-called multi-task Gaussian process [18] or the multivariate Gaussian process regression [37].

5.3.2 First properties

Let $\mathbf{C}$ and $\mathbf{D}$ be two matrices of size $m \times n$ and $p \times q$ respectively. Recall that the Kronecker product $\mathbf{C} \otimes \mathbf{D}$ is the $mp \times nq$ matrix such that

$$\mathbf{C} \otimes \mathbf{D} = \begin{pmatrix} c_{11}\mathbf{D} & \dots & c_{n1}\mathbf{D} \\ \vdots & \ddots & \vdots \\ c_{m1}\mathbf{D} & \dots & c_{mn}\mathbf{D} \end{pmatrix}$$

and $\text{vec}(\mathbf{C}) \in \mathbb{R}^{mn}$ is the vector obtained by stacking the n columns of $\mathbf{C}$:

$$\text{vec}(\mathbf{C}) = (c_{11}, \dots, c_{m1}, c_{12}, \dots, c_{m2}, \dots, c_{1n}, \dots, c_{mn})^\top.$$

Keeping these definitions in mind, the matrix-variate normal distribution $\mathcal{MN}_{p,q}$ [48, 166] is defined for all $p \times q$ random matrix $\mathbf{Y}^\star$ as:

$$\mathbf{Y}^\star \sim \mathcal{MN}_{p,q}(\mathbf{M}, \mathbf{\Sigma}, \mathbf{\Lambda}) \text{ if and only if } \text{vec}(\mathbf{Y}^\star) \sim \mathcal{N}_{pq}(\text{vec}(\mathbf{M}), \mathbf{\Sigma} \otimes \mathbf{\Lambda}), \tag{5.3}$$

where $\mathbf{M}$ is a $p \times q$ matrix, $\mathbf{\Sigma}$ and $\mathbf{\Lambda}$ are symmetric positive definite matrices of size $q \times q$ and $p \times p$ respectively. We refer to [48] for an early definition of the matrix-variate normal distribution (as well as some of its derivatives), to [77] for a general account on matrix-variate distributions and to [1] for an application to missing data imputation. The associated density function is defined for all $p \times q$ matrix $\mathbf{y}$ by

$$p(\mathbf{y}) = (2\pi)^{-pq/2} \det(\mathbf{\Sigma})^{-p/2} \det(\mathbf{\Lambda})^{-q/2} \exp\left(-\frac{1}{2}\text{tr}\left[\mathbf{\Lambda}^{-1}(\mathbf{y} - \mathbf{M})\mathbf{\Sigma}^{-1}(\mathbf{y} - \mathbf{M})^\top\right]\right), \tag{5.4}$$

where $\text{tr}(\cdot)$ denotes the trace operator. The next Proposition establishes that the finite sized marginals of the multivariate Gaussian process (5.1) can be interpreted as random matrices from a matrix-variate normal distribution.

Proposition 5.1. *Let $\mathbf{Y} \sim \mathcal{MGP}_p(\mathbf{m}, K, \mathbf{A})$ and introduce $\mathbf{Y}^\star$ the $p \times q$ random matrix defined as $\mathbf{Y}^\star = (\mathbf{Y}(t_1), \dots, \mathbf{Y}(t_q))$ where $(t_1, \dots, t_q) \in \mathcal{T}^q$. Then,*

$$\mathbf{Y}^\star \sim \mathcal{MN}_{p,q}(\mathbf{M}, \mathbf{\Sigma}, \mathbf{A}\mathbf{A}^\top), \tag{5.5}$$

where $\mathbf{M} = (\mathbf{m}(t_1), \dots, \mathbf{m}(t_q))$ and $\mathbf{\Sigma}$ is the covariance matrix defined by $\mathbf{\Sigma}_{k,\ell} = K(t_k, t_\ell)$ for all $(k, \ell) \in \{1, \dots, q\}^2$. Equivalently,

$$vec(\mathbf{Y}^\star) \sim \mathcal{N}_{pq}(\boldsymbol{\mu}, \mathbf{\Sigma} \otimes \mathbf{A}\mathbf{A}^\top),$$

with $\boldsymbol{\mu} = vec(\mathbf{M})$.

In the SITS framework, $\mathbf{Y}^\star$ represents the observed $q-$dimensional SITS which is a discretized version of $\mathbf{Y}$ at q timestamps. An illustration is provided in Fig. 5.3 where $\mathcal{T} = [0, 1]$ and $p = q = 10$. Only the first two coordinates are represented for lack of space reasons. Let $\langle \cdot, \cdot \rangle$ denote the Euclidean scalar product on $\mathbb{R}^p$ and $\| \cdot \|$ be the associated norm. For all non zero vectors $(u, v) \in \mathbb{R}^p \times \mathbb{R}^p$, we also introduce $\cos(u, v) = \langle u, v \rangle / (\|u\| \, \|v\|)$. As a direct consequence of the covariance structure in (5.5), the correlation ρ between the elements of the random matrix $\mathbf{Y}^\star$ can be derived:

Corollary 5.0.1. *Suppose the assumptions of Proposition 5.1 hold.*

(i) For all $(b, b') \in \{1, \dots, p\}^2$ and $j \in \{1, \dots, q\}$, one has

$$\rho(Y^\star_{b,j}, Y^\star_{b',j}) = \cos(\mathbf{a}_b, \mathbf{a}_{b'}),$$

(with $\mathbf{a}_b$ the bth line of $\mathbf{A}$) and is thus independent of $j \in \{1, \dots, q\}$.

(ii) For all $(j, j') \in \{1, \dots, q\}^2$ and $b \in \{1, \dots, p\}$, one has

$$\rho(Y^\star_{b,j}, Y^\star_{b,j'}) = \Sigma_{j,j'} \Big/ \sqrt{\Sigma_{j,j}\Sigma_{j',j'}}\,, \tag{5.6}$$

and is thus independent of $b \in \{1, \dots, p\}$.

It appears that $\mathbf{A}$ tunes the dependence between the lines of $\mathbf{Y}^\star$ (*i.e.* the spectral bands in the SITS context) while $\mathbf{\Sigma}$ drives the dependence between the columns (*i.e.* the acquisition times of the SITS).

A likelihood ratio test is introduced in [116] to check whether the separability of the covariance (5.5) is adapted to the data in hand. However, this test has not been extended to irregularly sampled time-series. Let us also mention that, in [120], the same Kronecker product model is used to regularize the estimation of the covariance matrix in high dimension.

5.4 Inference

This section addresses several inference aspects associated with the M2GP model. Consider $\{(\mathbf{Y}^1, Z_1), \ldots, (\mathbf{Y}^n, Z_n)\}$ a set of n random pairs identically distributed from the M2GP model. Clearly, π_c can be estimated by its empirical counterpart $\hat{\pi}_c = n_c/n$ where $n_c = \sum_{i=1}^n \mathbb{I}\{Z_i = c\}$ is the number of samples in class c (and $\mathbb{I}\{\cdot\}$ is the indicator function). Besides, from (5.2), $\mathbf{Y}^i \sim \mathcal{MGP}_p(\mathbf{m}_c, K_c, \mathbf{A}_c)$ conditionally to $Z_i = c$, for all $i \in \{1, \ldots, n\}$. The unknown quantities to be estimated are $\mathbf{m}_c : \mathcal{T} \to \mathbb{R}^p$, $K_c : \mathcal{T}^2 \to \mathbb{R}$ and the matrix $\mathbf{A}_c$. The use of parametric models for mean and covariance functions is discussed in Subsection 5.4.1 and the Maximum likelihood estimation (MLE) of all resulting parameters is presented in Subsection 5.4.2. The associated classification method based on the Maximum a posteriori (MAP) rule and the imputation of missing values are described in Subsection 5.4.3 and Subsection 5.4.4 respectively.

5.4.1 Parametric mean and covariance functions

Let $J > 0$ and introduce $\{\varphi_1, \ldots, \varphi_J\}$ a subset of J basis functions of $L_2(\mathcal{T})$. For all $b \in \{1, \ldots, p\}$, the bth coordinate $(\mathbf{m}_c(t))_b$ of $\mathbf{m}_c(t)$ is expanded as

$$(\mathbf{m}_c(t))_b = \sum_{j=1}^{J} \alpha_{c,b,j}\, \varphi_j(t), \tag{5.7}$$

with $t \in \mathcal{T}$, and where $a_{c,b,j}$ is the projection coefficient of $(\mathbf{m}_c(\cdot))_b$ on $\varphi_j(\cdot)$. Denoting by $\boldsymbol{\alpha}_c$ the $p \times J$ matrix defined by:

$$\boldsymbol{\alpha}_c = \begin{pmatrix} \alpha_{c,1,1} & \alpha_{c,1,2} & \ldots & \alpha_{c,1,J} \\ \alpha_{c,2,1} & \ddots & \ldots & \alpha_{c,2,J} \\ \vdots & \vdots & \ddots & \ldots \\ \alpha_{c,p,1} & \ldots & \ldots & \alpha_{c,p,J} \end{pmatrix}$$

and letting $\mathbf{b} : t \in \mathcal{T} \mapsto (\varphi_1(t), \ldots, \varphi_J(t))^\top \in \mathbb{R}^J$, then (5.7) can rewritten matricially as $\mathbf{m}_c(t) = \boldsymbol{\alpha}_c \mathbf{b}(t)$.

The covariance operator K_c is assumed to belong to a family of symmetric positive-definite kernels, [190, Chapter 4]. A typical kernel is the squared exponential kernel (also known as Gaussian or RBF kernel) with an additive white noise covariance function:

$$K_c(t, t'|\boldsymbol{\theta}_c) = \gamma_c^2 \exp\left(-\frac{(t-t')^2}{2h_c^2}\right) + \sigma_c^2 \mathbb{I}\{t = t'\}, \tag{5.8}$$

where $(t, t') \in \mathcal{T}^2$.The parameters are collected in $\boldsymbol{\theta}_c$ with, in this case, $\boldsymbol{\theta}_c = \{\gamma_c, h_c, \sigma_c\}$.

5.4.2 Maximum likelihood estimation

Assume each multivariate Gaussian process $\mathbf{Y}^i$ is observed on its own finite grid of distinct q_i timestamps $(t_1^i, \ldots, t_{q_i}^i) \in \mathbb{R}^{q_i}$ and note $\mathbf{Y}^{i,\star} = (\mathbf{Y}^i(t_1^i), \ldots, \mathbf{Y}^i(t_{q_i}^i))^\top$ the associated $p \times q_i$ random matrix. Let us stress that this formalism naturally allows to deal with irregularly sampled SITS since the size of $\mathbf{Y}^{i,\star}$ may depend on i. From Proposition 5.1, one has

$$\text{Conditionally to } Z_i = c,\ \mathbf{Y}^{i,\star} \sim \mathcal{MN}_{p,q_i}(\boldsymbol{\alpha}_c \mathbf{B}^i, \mathbf{\Sigma}^{c,i}(\boldsymbol{\theta}_c), \mathbf{A}_c \mathbf{A}_c^\top), \tag{5.9}$$

where the covariance matrix $\mathbf{\Sigma}^{c,i}(\boldsymbol{\theta}_c)$ is defined for all $(j, j') \in \{1, \ldots, q_i\}^2$ by $\mathbf{\Sigma}^{c,i}(\boldsymbol{\theta}_c)_{j,j'} = K_c(t_j^i, t_{j'}^i|\boldsymbol{\theta}_c)$ and $\mathbf{B}^i = (\mathbf{b}(t_1^i), \ldots, \mathbf{b}(t_{q_i}^i))$ is a $J \times q_i$ design matrix. Parameters $\{\boldsymbol{\alpha}_c, \boldsymbol{\theta}_c, \mathbf{A}_c\}$ are estimated by minimizing the negative log-likelihood given hereafter.

Lemma 5.1. *The negative log-likelihood associated with (5.9) can be expanded as*

$$\mathcal{L} = \frac{1}{2}\sum_{c=1}^{C} \ell_c(\boldsymbol{\alpha}_c, \boldsymbol{\theta}_c, \mathbf{A}_c\mathbf{A}_c^\top),$$

(up to an additive constant) where, for all $c \in \{1, \ldots, C\}$,

$$\begin{aligned}\ell_c(\boldsymbol{\alpha}_c, \boldsymbol{\theta}_c, \mathbf{A}_c\mathbf{A}_c^\top) &= Q_c \log\det(\mathbf{A}_c\mathbf{A}_c^\top) + p\sum_{i|Z_i=c} \log\det \boldsymbol{\Sigma}^{c,i}(\boldsymbol{\theta}_c) \\ &+ \operatorname{tr}\left(\sum_{i|Z_i=c} (\mathbf{Y}^{i,\star} - \boldsymbol{\alpha}_c\mathbf{B}^i)\{\boldsymbol{\Sigma}^{c,i}(\boldsymbol{\theta}_c)\}^{-1}(\mathbf{Y}^{i,\star} - \boldsymbol{\alpha}_c\mathbf{B}^i)^\top\{\mathbf{A}_c\mathbf{A}_c^\top\}^{-1}\right), \end{aligned} \tag{5.10}$$

with $Q_c = \sum_{i|Z_i=c} q_i$.

It appears that the likelihood only involves the product of matrices $\mathbf{A}_c\mathbf{A}_c^\top$ and not the matrix $\mathbf{A}_c$ itself. This is a direct consequence of (5.5): The matrix-variate normal distribution of the sampled process $\mathbf{Y}^\star$ only depends on the above product. The parameters of interest are thus $\boldsymbol{\alpha}_c$, $\boldsymbol{\theta}_c$ and $\mathbf{A}_c\mathbf{A}_c^\top$ and the MLE is obtained by solving C independent optimization problems:

$$(\hat{\boldsymbol{\alpha}}_c, \hat{\boldsymbol{\theta}}_c, \widehat{\mathbf{A}_c\mathbf{A}_c^\top}) = \underset{\boldsymbol{\alpha}_c, \boldsymbol{\theta}_c, \mathbf{A}_c\mathbf{A}_c^\top}{\arg\min}\ \ell_c(\boldsymbol{\alpha}_c, \boldsymbol{\theta}_c, \mathbf{A}_c\mathbf{A}_c^\top), \tag{5.11}$$

for all $c \in \{1, \ldots, C\}$. The solution is partially explicit as explained in the next Proposition.

Proposition 5.2. *Let $c \in \{1, \ldots, C\}$.*

(i) Solutions of (5.11) satisfy the following two properties. Given $\hat{\boldsymbol{\theta}}_c$, one has:

$$\hat{\boldsymbol{\alpha}}_c = \left[\sum_{i|Z_i=c} \mathbf{Y}^{i,\star}\{\boldsymbol{\Sigma}^{c,i}(\hat{\boldsymbol{\theta}}_c)\}^{-1}(\mathbf{B}^i)^\top\right]\left[\sum_{i|Z_i=c} \mathbf{B}^i\{\boldsymbol{\Sigma}^{c,i}(\hat{\boldsymbol{\theta}}_c)\}^{-1}(\mathbf{B}^i)^\top\right]^{-1}, \tag{5.12}$$

$$\widehat{\mathbf{A}_c\mathbf{A}_c^\top} = \frac{1}{Q_c}\sum_{i|Z_i=c} (\mathbf{Y}^{i,\star} - \hat{\boldsymbol{\alpha}}_c\mathbf{B}^i)\{\boldsymbol{\Sigma}^{c,i}(\hat{\boldsymbol{\theta}}_c)\}^{-1}(\mathbf{Y}^{i,\star} - \hat{\boldsymbol{\alpha}}_c\mathbf{B}^i)^\top. \tag{5.13}$$

(ii) The partial derivative of (5.10), w.r.t. the kth coordinate of $\boldsymbol{\theta}_c$ is given by:

$$\frac{\partial \ell_c(\boldsymbol{\alpha}_c, \boldsymbol{\theta}_c, \mathbf{A}_c\mathbf{A}_c^\top)}{\partial(\boldsymbol{\theta}_c)_k} = \sum_{i|Z_i=c} \operatorname{tr}\left(\left[p\{\boldsymbol{\Sigma}^{c,i}(\boldsymbol{\theta}_c)\}^{-1} - \boldsymbol{\beta}^{c,i}(\boldsymbol{\alpha}_c, \boldsymbol{\theta}_c)^\top\{\mathbf{A}_c\mathbf{A}_c^\top\}^{-1}\boldsymbol{\beta}^{c,i}(\boldsymbol{\alpha}_c, \boldsymbol{\theta}_c)\right]\frac{\partial \boldsymbol{\Sigma}^{c,i}}{\partial(\boldsymbol{\theta}_c)_k}(\boldsymbol{\theta}_c)\right), \tag{5.14}$$

where $\boldsymbol{\beta}^{c,i}(\boldsymbol{\alpha}_c, \boldsymbol{\theta}_c) = (\mathbf{Y}^{i,\star} - \boldsymbol{\alpha}_c\mathbf{B}^i)\{\boldsymbol{\Sigma}^{c,i}(\boldsymbol{\theta}_c)\}^{-1}$.

In practice, the computation of the MLE is achieved thanks to an iterative procedure based on (5.12)–(5.14), described in Algorithm 3 and discussed in Paragraph 5.4.5.

5.4.3 Supervised classification

Starting from a training set from the M2GP model $\{(\mathbf{Y}^{1,\star}, Z_1), \ldots, (\mathbf{Y}^{n,\star}, Z_n)\}$ with $\mathbf{Y}^{i,\star} = (\mathbf{Y}^i(t_1^i), \ldots, \mathbf{Y}^i(t_{q_i}^i))^\top$, our goal is to assign a label $\tilde{c} \in \{1, \ldots, C\}$ to a new $p \times q$ random matrix $\tilde{\mathbf{Y}}^\star = (\mathbf{Y}(\tilde{t}_1), \ldots, \mathbf{Y}(\tilde{t}_q))^\top$. We focus on the MAP rule which consists in maximizing w.r.t. c the posterior probability

$$\mathbb{P}(Z = c|\tilde{\mathbf{Y}}^\star) \propto \pi_c p(\tilde{\mathbf{Y}}^\star|Z = c),$$

where $p(\tilde{\mathbf{Y}}^\star|Z = c)$ is matrix-variate normal density defined as

$$\begin{aligned} p(\tilde{\mathbf{Y}}^\star|Z = c) &= (2\pi)^{-pq/2}\det\left(\tilde{\boldsymbol{\Sigma}}^c(\boldsymbol{\theta}_c)\right)^{-p/2}\det\left(\mathbf{A}_c\mathbf{A}_c^\top\right)^{-q/2} \\ &\times \exp\left(-\frac{1}{2}\operatorname{tr}\left[\{\mathbf{A}_c\mathbf{A}_c^\top\}^{-1}(\tilde{\mathbf{Y}}^\star - \boldsymbol{\alpha}_c\tilde{\mathbf{B}})\tilde{\boldsymbol{\Sigma}}^c(\boldsymbol{\theta}_c)^{-1}(\tilde{\mathbf{Y}}^\star - \boldsymbol{\alpha}_c\tilde{\mathbf{B}})^\top\right]\right), \end{aligned} \tag{5.15}$$

see (5.22) in the proof of Lemma 5.1. Here, the covariance matrix $\tilde{\mathbf{\Sigma}}^c(\boldsymbol{\theta}_c)$ is defined for all $(j, j') \in \{1, \ldots, q\}^2$ by $\tilde{\mathbf{\Sigma}}^c(\boldsymbol{\theta}_c)_{j,j'} = K_c(\tilde{t}_j, \tilde{t}_{j'}|\boldsymbol{\theta}_c)$ and $\tilde{\mathbf{B}} = (\mathbf{b}(\tilde{t}_1), \ldots, \mathbf{b}(\tilde{t}_q))$ is a $J \times q$ design matrix. In practice, all parameters are replaced using their MLE counterparts and $\tilde{c}$ is selected by minimizing the negative log posterior probability, that is:

$$\tilde{c} = \arg\min_c \Big\{ p \log\det\left(\tilde{\mathbf{\Sigma}}^c(\hat{\boldsymbol{\theta}}_c)\right) + q \log\det\left(\widehat{\mathbf{A}_c\mathbf{A}_c^\top}\right) - 2\log(n_c/n) \\ + \operatorname{tr}\left[\left\{\widehat{\mathbf{A}_c\mathbf{A}_c^\top}\right\}^{-1}(\tilde{\mathbf{Y}}^\star - \hat{\boldsymbol{\alpha}}_c\tilde{\mathbf{B}})\tilde{\mathbf{\Sigma}}^c(\hat{\boldsymbol{\theta}}_c)^{-1}(\tilde{\mathbf{Y}}^\star - \hat{\boldsymbol{\alpha}}_c\tilde{\mathbf{B}})^\top\right]\Big\}.$$

In the SITS framework, the above formula provides a natural way to classify a new multivariate time-series even though it is not observed at the same timestamps as the examples from the training set.

5.4.4 Imputation of missing values

The next result provides the distribution of the MGP process at time $t^\dagger$ conditionally to its label and to observations at times $t_1, \ldots, t_q$.

Proposition 5.3. *Assume that, conditionally to $Z = c$, $\mathbf{Y} \sim \mathcal{MGP}_p(\boldsymbol{\alpha}_c\mathbf{b}, K_c, \mathbf{A}_c)$ and introduce $\mathbf{Y}^\star$ the $p \times q$ random matrix defined as $\mathbf{Y}^\star = (\mathbf{Y}(t_1), \ldots, \mathbf{Y}(t_q))$ where $(t_1, \ldots, t_q) \in \mathcal{T}^q$. Let $t^\dagger \in \mathcal{T}$ such that $t^\dagger \neq t_k$ for all $k \in \{1, \ldots, q\}$. Then,*

$$\text{conditionally to } Z = c \text{ and } \mathbf{Y}^\star = \mathbf{y}^\star,\ \mathbf{Y}(t^\dagger) \sim \mathcal{N}_p\left(\boldsymbol{\mu}_c(t^\dagger, \mathbf{y}^\star),\ \mathbf{\Lambda}_c(t^\dagger)\right),$$

with

$$\boldsymbol{\mu}_c(t^\dagger, \mathbf{y}^\star) = \boldsymbol{\alpha}_c\mathbf{b}(t^\dagger) + (\mathbf{y}^\star - \boldsymbol{\alpha}_c\mathbf{B})\{\mathbf{\Sigma}^c(\boldsymbol{\theta}_c)\}^{-1}\mathbf{k}_c(t^\dagger),$$
$$\mathbf{\Lambda}_c(t^\dagger) = \left[K_c(t^\dagger, t^\dagger|\boldsymbol{\theta}_c) - \mathbf{k}_c(t^\dagger)^\top\mathbf{\Sigma}^c(\boldsymbol{\theta}_c)^{-1}\mathbf{k}_c(t^\dagger)\right] \otimes \mathbf{A}_c\mathbf{A}_c^\top,$$

and where $\mathbf{k}_c(t^\dagger) = (K_c(t^\dagger, t_1|\boldsymbol{\theta}_c), \ldots, K_c(t^\dagger, t_q|\boldsymbol{\theta}_c))^\top$. Recall that the covariance matrix $\mathbf{\Sigma}^c(\boldsymbol{\theta}_c)$ is defined for all $(j, j') \in \{1, \ldots, q\}^2$ by $\mathbf{\Sigma}^c(\boldsymbol{\theta}_c)_{j,j'} = K_c(t_j, t_{j'}|\boldsymbol{\theta}_c)$ and $\mathbf{B} = (\mathbf{b}(t_1), \ldots, \mathbf{b}(t_q))$ is a $J \times q$ design matrix.

As a consequence, when $\mathbf{Y}(t^\dagger)$ is not observed (but its label is known to be c), this missing value can be imputed by the conditional expectation given in Proposition 5.3, where the unknown parameters are replaced by their associated MLE:

$$\hat{\mathbf{Y}}_c(t^\dagger) = \hat{\boldsymbol{\alpha}}_c\mathbf{b}(t^\dagger) + (\mathbf{Y}^\star - \hat{\boldsymbol{\alpha}}_c\mathbf{B})\{\mathbf{\Sigma}^c(\hat{\boldsymbol{\theta}}_c)\}^{-1}\hat{\mathbf{k}}_c(t^\dagger). \tag{5.16}$$

This allows for the reconstruction of SITS values at unobserved times. If the label of $\mathbf{Y}^\star$ is unknown, the distribution of the MGP process at time $t^\dagger$ conditionally to observations at times $t_1, \ldots, t_q$ can still be derived from Proposition 5.3:

$$\text{conditionally to } \mathbf{Y}^\star = \mathbf{y}^\star,\ \mathbf{Y}(t^\dagger) \sim \sum_{c=1}^{C} \mathbb{P}(Z = c|\mathbf{Y}^\star = \mathbf{y}^\star)\mathcal{N}_p\left(\boldsymbol{\mu}_c(t^\dagger, \mathbf{y}^\star),\ \mathbf{\Lambda}_c(t^\dagger)\right),$$

leading to

$$\boldsymbol{\mu}(t^\dagger, \mathbf{y}^\star) = \sum_{c=1}^{C} \mathbb{P}(Z = c|\mathbf{Y}^\star = \mathbf{y}^\star)\boldsymbol{\mu}_c(t^\dagger, \mathbf{y}^\star),$$
$$\mathbf{\Lambda}(t^\dagger) = \sum_{c=1}^{C} \mathbb{P}(Z = c|\mathbf{Y}^\star = \mathbf{y}^\star)\left(\mathbf{\Lambda}_c(t^\dagger) + \boldsymbol{\mu}_c(t^\dagger, \mathbf{y}^\star)^\top\boldsymbol{\mu}_c(t^\dagger, \mathbf{y}^\star)\right) - \boldsymbol{\mu}(t^\dagger, \mathbf{y}^\star)^\top\boldsymbol{\mu}(t^\dagger, \mathbf{y}^\star).$$

Thus, when both $\mathbf{Y}(t^\dagger)$ and its label are not observed, $\mathbf{Y}(t^\dagger)$ can be imputed by

$$\hat{\mathbf{Y}}(t^\dagger) = \sum_{c=1}^{C} \widehat{\mathbb{P}}(Z = c|\mathbf{Y}^\star)\hat{\mathbf{Y}}_c(t^\dagger), \tag{5.17}$$

where $\hat{\mathbf{Y}}_c(t^\dagger)$ is given in (5.16), $\mathbf{Y}^\star = (\mathbf{Y}(t_1), \ldots, \mathbf{Y}(t_q))$ and

$$\widehat{\mathbb{P}}(Z = c|\mathbf{Y}^\star) = \hat{\pi}_c\hat{p}(\mathbf{Y}^\star|Z = c) \Big/ \sum_{k=1}^{C} \hat{\pi}_k\hat{p}(\mathbf{Y}^\star|Z = k)\ ,$$

with $\hat{p}(\mathbf{Y}^\star|Z = k)$ the estimated matrix-variate density defined similarly to (5.15) by

$$\hat{p}(\mathbf{Y}^\star|Z = k) = (2\pi)^{-pq/2}\det\left(\mathbf{\Sigma}^k(\hat{\boldsymbol{\theta}}_k)\right)^{-p/2}\det\left(\widehat{\mathbf{A}_k\mathbf{A}_k^\top}\right)^{-q/2} \\ \times \exp\left(-\frac{1}{2}\operatorname{tr}\left[\widehat{\mathbf{A}_k\mathbf{A}_k^\top}\right\}^{-1}(\mathbf{Y}^\star - \hat{\boldsymbol{\alpha}}_k\mathbf{B})\mathbf{\Sigma}^c(\hat{\boldsymbol{\theta}}_k)^{-1}(\mathbf{Y}^\star - \hat{\boldsymbol{\alpha}}_k\mathbf{B})^\top\right]\right).$$

5.4.5 Numerical implementation

The computation of the MLE is implemented as detailed in Algorithm 3 using the results of Proposition 5.2. To deal with the idenfiability issue mentioned in Paragraph 5.3.1, $\mathbf{A}_c\mathbf{A}_c^\top$ is normalised by η_c such that $\|\mathbf{A}_c\mathbf{A}_c^\top\|_F = 1$ (where $\|\cdot\|_F$ denotes the Frobenius norm) and each covariance matrix $\mathbf{\Sigma}^{c,i}(\boldsymbol{\theta}_c)$ is modified accordingly so that the likelihood remains unaffected (step (d) of Algorithm 3). The gradient step (e) is performed using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm, see [197]. More specifically, the L-BFGS-B version is used which allows for box and positivity constraints. As described in [197], the gradient step is obtained by line search and the algorithm stops when: the objective function (*i.e.* the likelihood) does not change significantly, the (infinite) norm of the projected gradient is sufficiently small or when the maximum number of iterations is reached. Since the objective function is not convex, the optimization process is sensitive to the initialization. In practice, multiple random restarts are used and the best solution is retained. Let us highlight that, in practice, steps (a)-(e) are computed for all classes in parallel since the model parameters are decoupled w.r.t. the classes.

Algorithm 3: Computation of MLE of model parameters.

Input : Sample $\left\{(\mathbf{Y}^{i,\star}, Z_i) \in \mathbb{R}^{p\times q_i} \times \{1,\dots,C\},\ i = 1,\dots,n\right\}$ and initialization $(\boldsymbol{\theta}_1,\dots,\boldsymbol{\theta}_C)$.

Output: MLE $\left(\hat{\boldsymbol{\alpha}}_c, \widehat{\mathbf{A}_c\mathbf{A}_c^\top}, \hat{\boldsymbol{\theta}}_c\right)$, $c = 1,\dots,C$.

for $c = 1$ **to** C **do**
  repeat
    (a) Update $\boldsymbol{\alpha}_c$ using (5.12);
    (b) Update $\mathbf{A}_c\mathbf{A}_c^\top$ using (5.13);
    (c) Compute $\eta_c \leftarrow \|\mathbf{A}_c\mathbf{A}_c^\top\|_F$;
    (d) Update $\mathbf{A}_c\mathbf{A}_c^\top \leftarrow \mathbf{A}_c\mathbf{A}_c^\top/\eta_c$ and $\mathbf{\Sigma}^{c,i}(\boldsymbol{\theta}_c) \leftarrow \eta_c\mathbf{\Sigma}^{c,i}(\boldsymbol{\theta}_c)$, $i = 1,\dots,n$;
    (e) Update $\boldsymbol{\theta}_c$ with a gradient step using (5.14);
  until $\ell_c(\boldsymbol{\alpha}_c, \mathbf{A}_c\mathbf{A}_c^\top, \boldsymbol{\theta}_c)$ *has converged*;

The numerical complexity of one iteration for all classes of Algorithm 3 is $O(n(q_\infty^3 + p^3 + J^3))$ where n is the sample size and $q_\infty = \max\{q_i,\ i = 1,\dots,n\}$. The computation of the MLE thus scales linearly w.r.t. n. In constrast, the cost associated with standard classfication methods based on Gaussian processes is $O((C+1)n^3)$ [190, Algorithm 3.3]. Here, the computation of the MLE only relies on the inversion of $p \times p$ and $q_i \times q_i$ matrices whose sizes do not depend on the sample size.

Let us note that Algorithm 3 can be interpreted as an extension of the so-called Flip-flop method introduced independently by [123, 53]. This latter method in an iterative way to compute the MLE associated with the matrix-variate normal distribution. As such, it is limited to the situation where $q_1 = q_2 = \cdots = q_n$ which only occurs when all Gaussian processes are observed on a common grid. Identifiability issues are discussed in [166] and the method is extended to higher order tensor distributions in [121]. Applications of matrix-variate normal distribution are found in different contexts such as electro-encephalography [165] or remote sensing [71].

Finally, all the above estimations procedures have been implemented in Python using the Scikit-Lean API, see [29]. The Fourier basis $\{\varphi_1,\dots,\varphi_J\}$ was chosen to estimate the mean function (see [41] for other bases), while the family of symmetric positive-definite kernels was selected among the *Kernels* class in the Scikit-Learn library.

5.5 Validation on simulated data

The performance of the inference procedure associated with the M2GP model is illustrated on simulated data.[1] The simulated model is described in Paragraph 5.5.1. First, the influence of the dependence between coordinates as well as the influence of the number of observation times are investigated in Paragraph 5.5.2. Second; the consequences on the classification and imputation accuracy are discussed in Paragraph 5.5.3.

5.5.1 Experimental design

A binary classification problem is considered. Two classes are simulated from a 10-dimensional M2GP model on $\mathcal{T} = [0,1]$ with $1,000$ samples per class leading to $n = 2,000$ and $p = 10$. Mean functions are generated following (5.7) with a Fourier basis of size $J = 11$. Coefficients $\alpha_{c,b,j}$ are simulated independently from a $\mathcal{N}_1(0, 0.02)$

[1]The code and a notebook are available at `https://gitlab.inria.fr/aconstan/mixture-of-multivariate-gaussian-processes-for-classification-of-irregularly-sampled-satellite-image-time-series`.

distribution, $c \in \{1, 2\}$, $b \in \{1, \ldots, 10\}$ and $j \in \{1, \ldots, 11\}$. The covariance operator is identical for both classes: $K_1(\cdot, \cdot) = K_2(\cdot, \cdot)$. It is defined following (5.8) as the sum of a RBF kernel and a white noise covariance function. The associated parameters are $\boldsymbol{\theta}_1 = \{\gamma_1, h_1, \sigma_1\} = \{1.5, 150, 0.05\} = \boldsymbol{\theta}_2$. We also set $\mathbf{A}_1 = \mathbf{A}_2$ with

$$\mathbf{A}_1 \mathbf{A}_1^\top = \begin{pmatrix} 1 & \beta & \cdots & \beta \\ \beta & 1 & \cdots & \vdots \\ \vdots & \vdots & \ddots & \beta \\ \beta & \cdots & \beta & 1 \end{pmatrix}, \tag{5.18}$$

so that β tunes the pairwise correlation between the 10 coordinates of the Gaussian processes. In the following, we shall consider $\beta \in \{0, 1/4, 1/2\}$. In practice, M2GP processes are simulated on random grids of varying size $q \in \{10, 20, \ldots, 100\}$, see Fig. 5.3 for an illustration in the case $q = 10$ and $\beta = 0$.

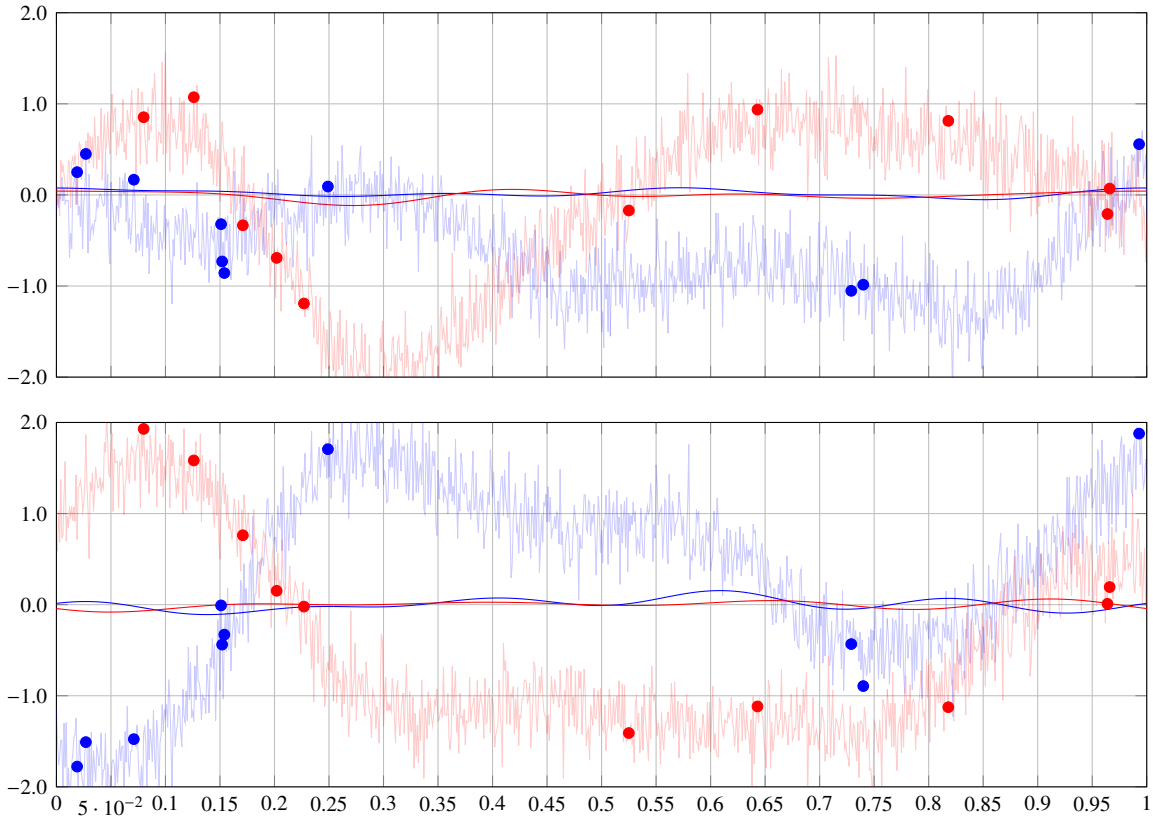


Figure 5.3: Two simulated M2GP processes (transparent lines) in dimension $p = 10$ observed at $q = 10$ timestamps (dots), from two classes ($c = 1$: blue, $c = 2$: red). The mean functions are depicted as continuous opaque lines. Top panel: first coordinates, bottom panel: second coordinates (only the first two coordinates p_1 and p_2 are represented).

5.5.2 Estimation results

All estimation procedures are evaluated on 100 replications of the above described simulation model. First, for all $c \in \{1, 2\}$, the quality of the reconstructed mean $\hat{\mathbf{M}}_c = \hat{\boldsymbol{\alpha}}_c \mathbf{B}$ is measured by the normalized Mean Squared Error (nMSE) defined as:

$$\text{nMSE}(\hat{\mathbf{M}}_c, \mathbf{M}_c) = \frac{\|\mathbf{M}_c - \hat{\mathbf{M}}_c\|_F^2}{\|\mathbf{M}_c - \overline{\mathbf{M}}_c\|_F^2}, \tag{5.19}$$

where $\overline{\mathbf{M}}_c$ is the empirical mean of the processes in class c. The lower this score is, the better the estimation. An example of reconstructed mean is presented on Fig. 5.4, for one replication. Second, the quality of the estimation

of the covariance structure $\mathbf{A}_c\mathbf{A}_c^\top$ (see 5.18) by $\widehat{\mathbf{A}_c\mathbf{A}_c^\top}$ is assessed by the cosine score defined as:

$$C(\widehat{\mathbf{A}_c\mathbf{A}_c^\top}, \mathbf{A}_c\mathbf{A}_c^\top) = 1 - \frac{\langle \widehat{\mathbf{A}_c\mathbf{A}_c^\top}, \mathbf{A}_c\mathbf{A}_c^\top \rangle_F}{\|\widehat{\mathbf{A}_c\mathbf{A}_c^\top}\|_F \|\mathbf{A}_c\mathbf{A}_c^\top\|_F}. \tag{5.20}$$

Let us note that $C(\widehat{\mathbf{A}_c\mathbf{A}_c^\top}, \mathbf{A}_c\mathbf{A}_c^\top) \in [0, 2]$ with $C(\widehat{\mathbf{A}_c\mathbf{A}_c^\top}, \mathbf{A}_c\mathbf{A}_c^\top) = 0$ when $\widehat{\mathbf{A}_c\mathbf{A}_c^\top}$ and $\mathbf{A}_c\mathbf{A}_c^\top$ are proportional. Finally, turning to the estimation of the kernel part (5.8) of the dependence structure, we focus on the estimation accuracy of the length-scale by computing the absolute difference between the true length-scale $h_1 = h_2 = 150$ and its estimated counterpart. The results are averaged over the 100 independent replications and are reported on Fig. 5.5 for the first class. Similar results are obtained for the second one. It appears that, unsurprisingly, the quality of the estimates increases with the number q of discretization times. At the opposite, the dependence parameter β does not seem to influence much the accuracy of the estimation. One can nevertheless note that, as expected, the variability of the estimators increases with β, as the information carried by correlated coordinates decreases. Besides, the estimated length-scales do not depend on β, this may be explained by the separability property exhibited in Corollary 5.0.1.

5.5.3 Classification and imputation results

Here, we focus on the comparison between results associated with M2GP and MIGP models. To assess the classification and imputation performances, $4,000$ samples are generated following the model described in Paragraph 5.5.1 and then split into two disjoint balanced sets. The first one is used as a training set (of size $n = 2,000$) to estimate model parameters. The second one is used as a test set where the accuracy of the classification and imputation steps associated with the two above methods are compared. The classification performance is assessed thanks to the Overall Accuracy (OA), that is the ratio of the number correctly classified test observations and the total number of test observations, while the nMSE is used for the imputation task. Similarly to (5.19), we let

$$\text{nMSE}(\hat{\mathbf{Y}}^\star, \mathbf{Y}^\star) = \frac{\|\hat{\mathbf{Y}}^\star - \mathbf{Y}^\star\|_F^2}{\|\mathbf{Y}^\star - \overline{\mathbf{Y}^\star}\|_F^2}, \tag{5.21}$$

where $\hat{\mathbf{Y}}^\star$ is the imputed discretized process when the class is unknown thanks to (5.17), given the observed discretized process on q points. $\overline{\mathbf{Y}^\star}$ is the empirical mean of discretized processes in the test set. The above Frobenius norms are computed on a fixed regular grid of $\mathcal{T}$ defined as $\{t_\ell = \ell/100,\ \ell = 1, \ldots, 100\}$. The results are reported in Fig. 5.6.

It appears that the classification scores associated with M2GP increase with the dependence coefficient β and the number q of discretization times. On the opposite, MIGP scores are decreasing with β, due to the independence assumption. When there is no dependence between coordinates ($\beta = 0$), both methods provide similar classification scores. Unsurprisingly, M2GP outperforms MIGP as soon as a dependence occurs.

In terms of reconstruction, both methods feature similar performances, increasing with q. The dependence strength only impacts the variance of the reconstructed processes: The larger β is, the larger the variability.

5.6 Time-series classification: Application to satellite data

This section is devoted to multivariate SITS classification using the M2GP model. The data were acquired by the Sentinel-2 satellite, and are presented in Paragraph 5.6.1, with a focus on the irregular temporal sampling. The estimated M2GP parameters are interpreted and discussed in Paragraph 5.6.2. Finally Paragraph 5.6.3 concludes this section with classification results and comparisons to state-of-the-art methods.

5.6.1 Sentinel-2 satellite image time-series

Since 2016, the Sentinel-2 mission [52] produces massive multispectral images,[2] around 1.6TBytes a day, with a spatial resolution of 10 m/pixel and 13 spectral bands (only 10 bands are used for the analysis). The frequency of revisit is 5 days and clouds as well as shadows are present in the data, at random locations. Most of the clouds and shadows positions are automatically extracted by the data provider. Yet, thin clouds may remain in the data. The selected images cover the area of Toulouse, France (Fig. 5.7) and all available acquisitions for the year 2018 were used. The image is of spatial size 10,000×10,000 pixels (10,000 km^2). Each extracted time-series i has a dimension of $p = 10$ channels (or bands) and its own number of timestamps q_i. The distribution of the q_is is represented in Fig. 5.8 for this area in 2018.

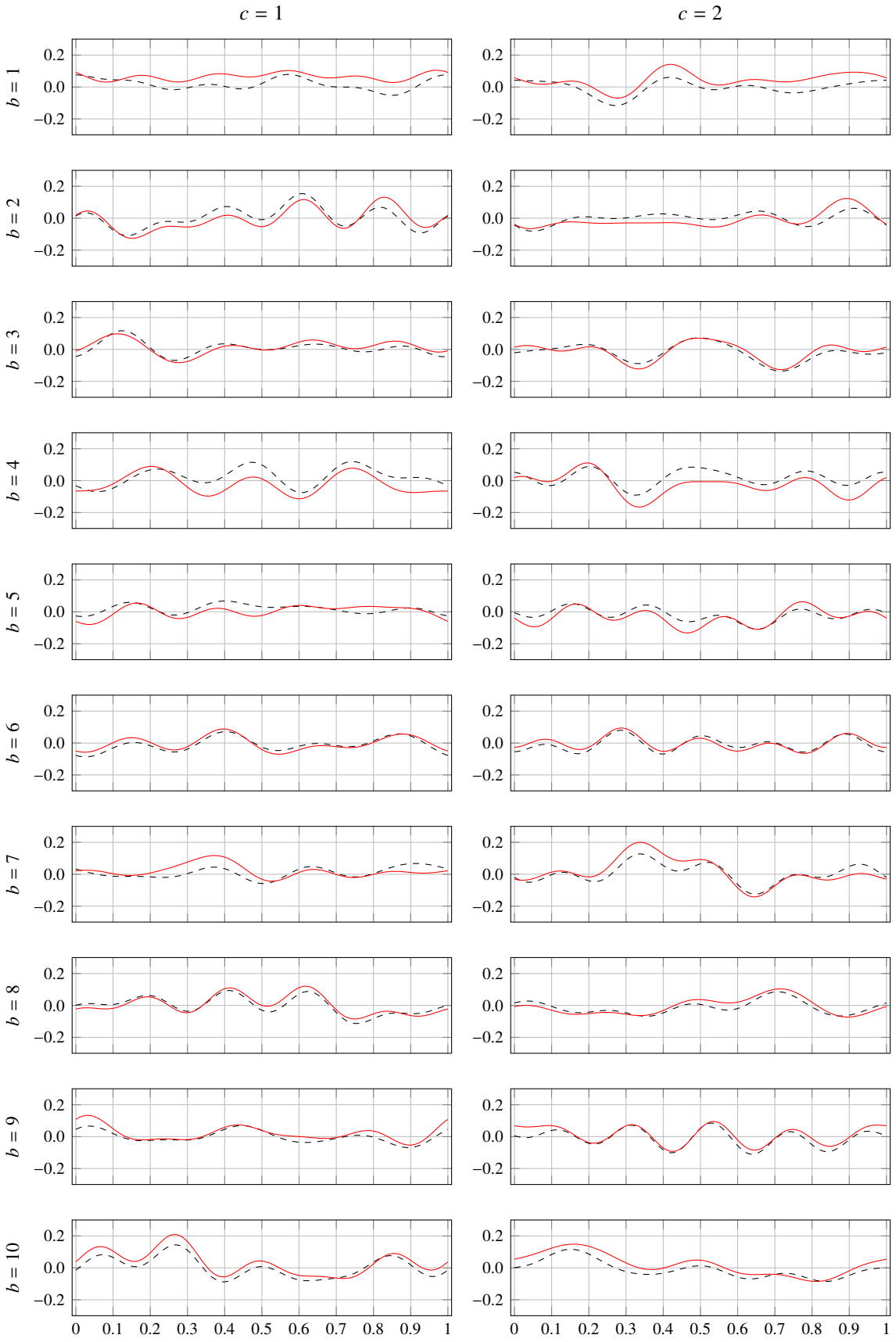


Figure 5.4: Estimation of mean functions by M2GP on simulated data for all coordinates $b \in \{1, \ldots, 10\}$, classes $c \in \{1, 2\}$ and $\beta = 0$ on one replication. The dashed line is the true mean, the red line is the estimated GP mean from a discretization on a grid of size $q = 10$.

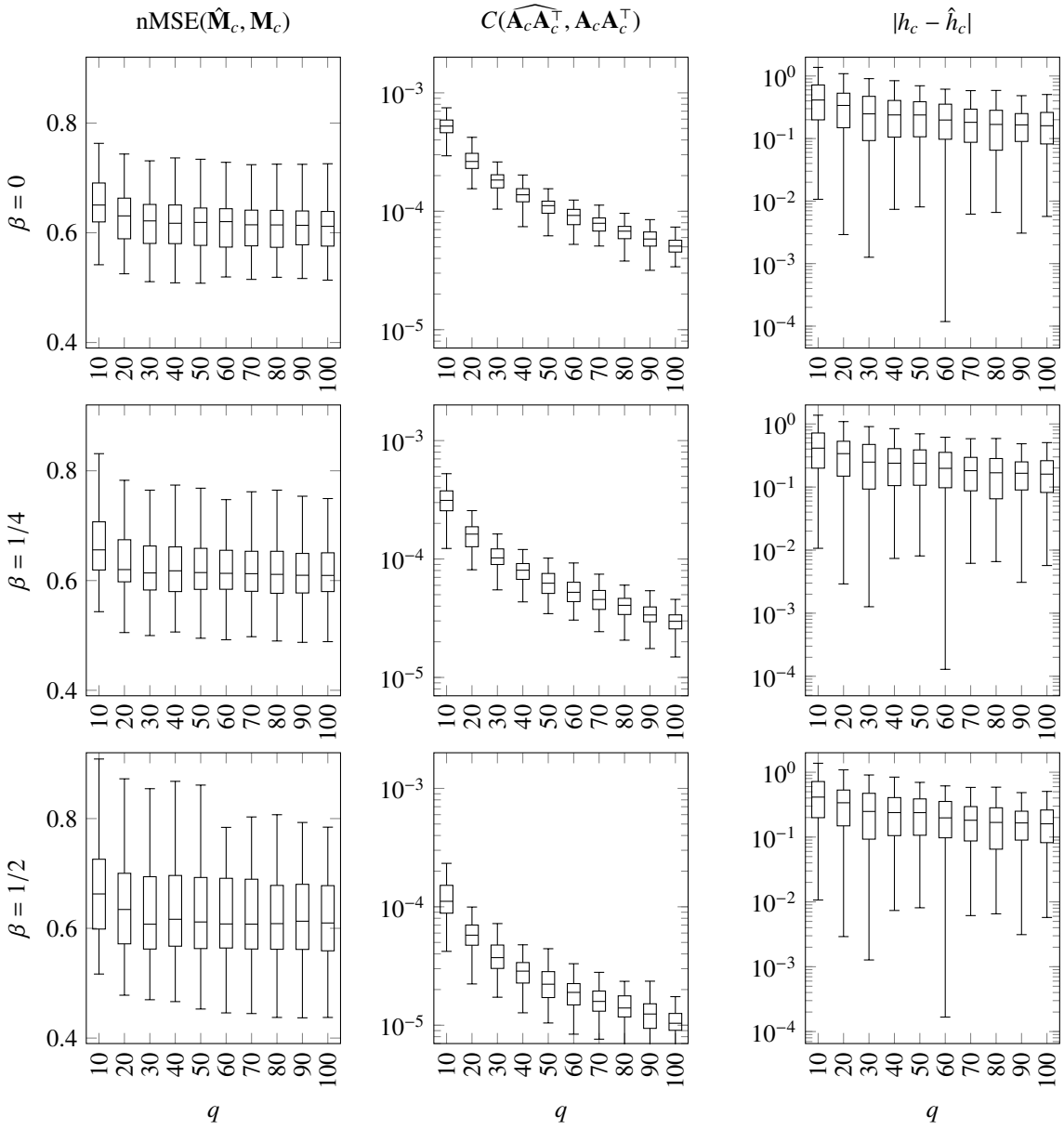


Figure 5.5: Estimation of M2GP parameters on simulated data as a function of the number q of discretization times on class $c = 1$. From left to right: normalized mean squared error (5.19), cosine score (5.20) and absolute difference of length-scales. From top to bottom: $\beta = 0$, $\beta = 1/4$ and $\beta = 1/2$.

The supervised classification task consists in assigning a pre-defined label to every pixel of the image. Fourteen classes were extracted from national data-bases and 10 pairs of training and validation data-sets are generated independently for the experiments by randomly selecting samples for the training and testing sets. Training and testing sets were carefully constructed to avoid spatial dependence between pixels.

Table 5.1 shows the number of extracted samples for each training and validation set. The number of samples per class is unbalanced but represents the actual proportion of land cover classes in the region.

5.6.2 Parameters estimation

M2GP is fitted to the satellite image time-series using the estimators described in Section 5.4. A Fourier basis is adopted for estimating the means using $J = 19$ functions while the time dependence structure is modeled by a RBF kernel combined with an additive white noise. The choice of the basis and the selection of the dimension J are

[2] `https://sentinel.esa.int/web/sentinel/missions/sentinel-2/data-products`.

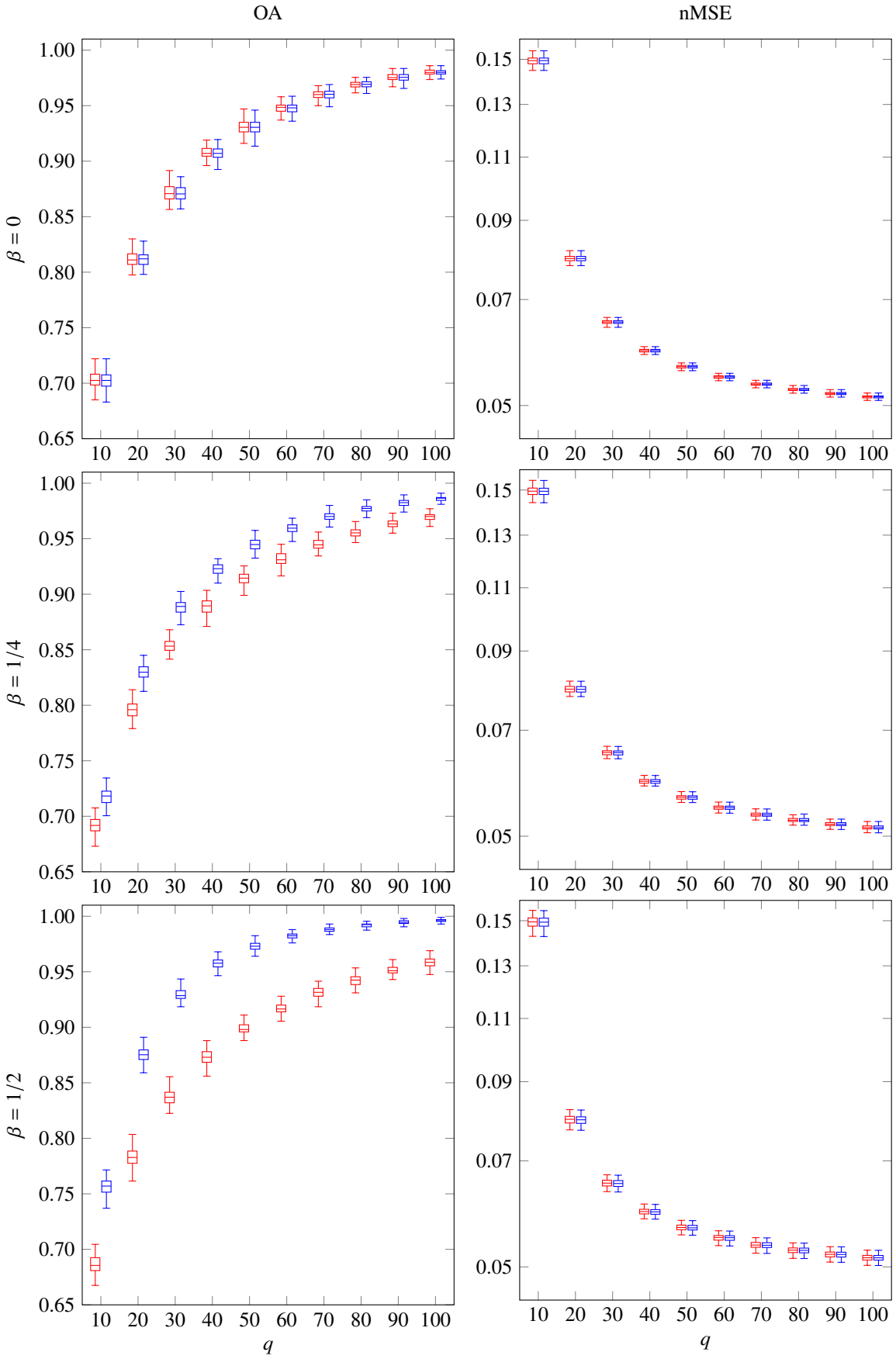


Figure 5.6: Classification overall accuracy (OA, left panel) and reconstruction normalized mean-squared error (nMSE, right panel in log scale) boxplots computed on simulated data. Comparison between M2GP (blue) and MIGP (red) results as functions of the number q of discretization times. From top to bottom: $\beta = 0$, $\beta = 1/4$ and $\beta = 1/2$.

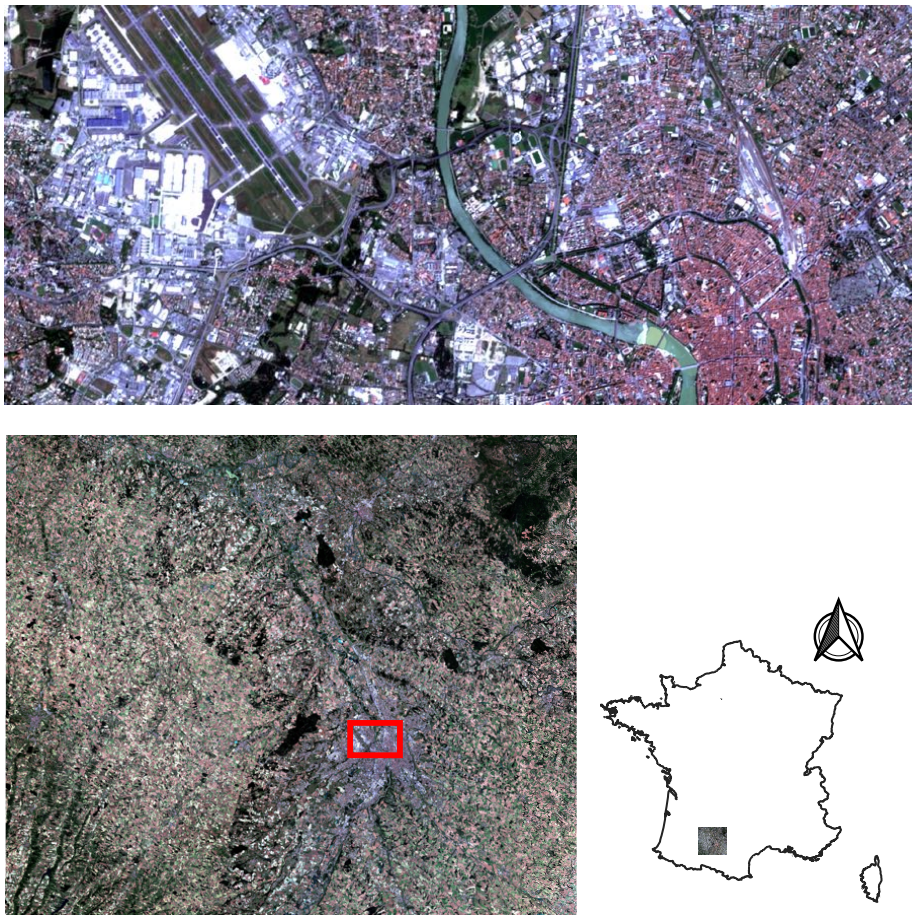


Figure 5.7: The study area is located in the south of France (right bottom image). The left bottom image corresponds to the entire area (100 km×100 km) and the upper image is a zoom over the red rectangle (11 km×5 km).

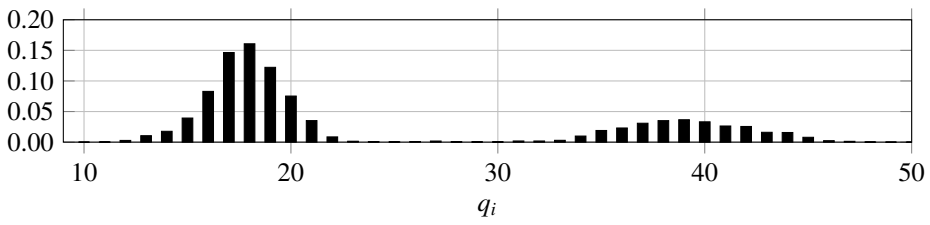


Figure 5.8: Normalized histogram of the q_is within the SITS data-set.

Table 5.1: Land cover classes and number of extracted samples n_c per class for each training and validation set.

Class	n_c
Summer crops	40,000
Winter crops	30,000
Broad-leaved forest	10,000
Continuous urban fabric	10,000
Discontinuous urban fabric	10,000
Industrial or commercial units	10,000
Meadow	10,000
Orchards	10,000
Road surfaces	10,000
Vines	10,000
Water bodies	10,000
Woody moorlands	9,972
Coniferous forest	9,957
Natural grasslands	9,939
Total	189,868

discussed in the MIGP framework by [41, Fig. 8, and Fig. 1 in the supp. mat.].

Estimated mean functions are reported in Fig. 5.9 for four selected channels: blue, green, red and near infrared (nIR) and four selected classes: continuous urban fabric, summer crops, broad-leaved forest and water bodies. In the context of remote sensing data, nIR is often correlated with the presence or absence of vegetation: Large values of nIR associated with small values of red, indicate that the vegetation is abundant. This behavior is observed in agricultural classes such as summer crops or broad-leaved forest during spring and summer.

The estimated covariance matrices between all 10 channels $\widehat{\mathbf{A}_c\mathbf{A}_c^\top}$ are reported in Fig. 5.10 for the same classes. Similar covariance matrices have already been observed on mono-temporal Sentinel-2 data, we refer to [185, Fig. 8] for similar results on crops classes.

Finally, the time covariance structure is illustrated on Fig. 5.11. The estimated RBF kernel on the same four classes is drawn when centered at day 180. The temporal correlation associated with natural elements, such as summer crops or broad-leaved forest, is short since their reflectance evolves along the year (*e.g.* because of the vegetation cycle, or anthropic events). In contrast, man-made materials, such as continuous urban fabrics, exhibit longer temporal correlation because their reflectance does not evolve along the time.[3]

5.6.3 Classification results

In this section, the classification performances of M2GP are compared to state-of-the-art methods. Four competitors are considered: Random forests (RF) [26], Quadratic discriminant analysis (QDA) which is based on a finite-dimensional Gaussian model, linear Support vector machine (SVM) classifier fitted with a Stochastic Gradient Descent [196], and, finally, Mixture of independent Gaussian processes (MIGP) [41]).

The time-series have been resampled on a common temporal grid of size 73 (every 5 days of year 2018) using a linear interpolation for RF, QDA and SVM methods since they require a fix vectorial representation of the sample. All the spectral bands have been stacked together to obtain a vector of dimension 73 dates $\times$ 10 spectral bands = 730 features. RF is trained with 100 trees of depth 25, and QDA is used with a regularized version of the estimated covariance matrix [63], $\tilde{\mathbf{\Sigma}} = (1-\epsilon)\hat{\mathbf{\Sigma}} + \epsilon\mathbf{I}$, with $\epsilon = 10^{-2}$.

The F_1-score is computed to assess numerically the classification accuracy. The F_1 is defined as the harmonic mean of the precision and recall scores [174]. Classification maps are also presented in order to qualitatively evaluate the spatial coherency of the results (despite a spatial pixel-wise independence assumption made by all considered methods).

Means F_1 scores and their standard deviations computed on 10 independent sets are reported in Table 5.2 for each class as well as the "average F_1 score" computed on all classes. Nnon-parametric methods (RF and SVM) provide the best classification results in terms of F1-score. The uni-modal assumption induced by Gaussian models

[3]This is true when the period of observation is not too long, few years, otherwise the material property might be altered and its reflectance could vary.

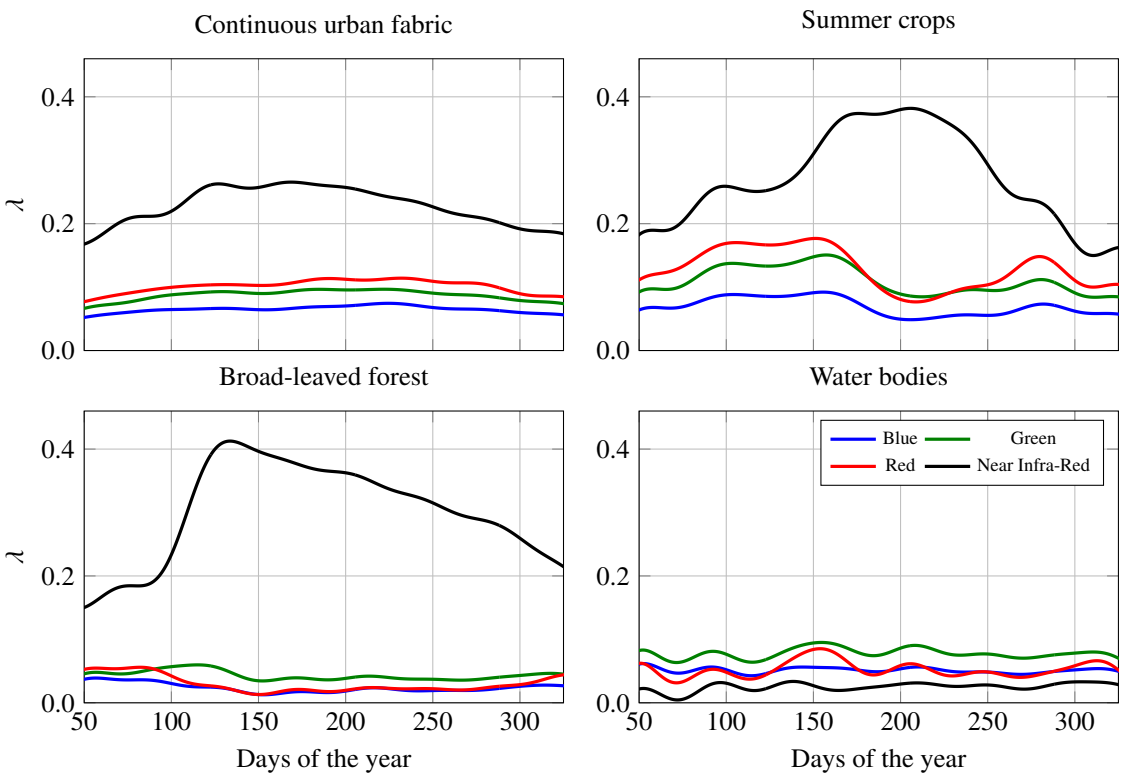


Figure 5.9: Estimated means for four channels and four classes (continuous urban fabric, summer crops, broad-leaved forest and water bodies). The horizontal axis represents the days of the year and the vertical axis represents the reflectance value.

may thus be ill-adapted to this data-set. M2GP and QDA provide lower and similar accuracy, even though M2GP is based on stronger assumptions on the covariance structure than QDA.

The obtained classification maps are reported in Fig. 5.12 for 3 different sites. Large differences are observed in these scenes. For the first column, corresponding to the airport zone, most of the inner vegetations are wrongly classified to natural grasslands with QDA, while RF, SVM and M2GP classify correctly them as meadow. Runway are mostly confused with industrial/commercial units using RF while runways are almost recovered by M2GP. Overall, strong differences between thematic maps are observed, but visual assessment from a mono-date color image is difficult. Yet, without taking into account the spatial dependence, M2GP recovers most of the spatial structure of the image, and the *salt and pepper* classification noise is limited, as for RF and SVM.

5.7 Discussion

A multivariate Gaussian process model has been introduced for the classification of irregularly sampling satellite image time-series. The multivariate model involves a specific structure of the covariance operator that exploits the data features and also reduces the number of parameters to estimate. Furthermore, the proposed formulation scales linearly w.r.t. the number of samples. Experimental results on simulated and real data sets show the importance of modeling the dependence between coordinates of the process, in particular for classification accuracy.

Current development concerns the use of two satellite sources. Sentinel-2 satellites are complemented with Sentinel-1 ones (which are not affected by clouds) which acquire radar data (with a different physical content): An extension of the proposed model will consist in combining these two time-series with irregular temporal and spectral sampling. Another possible extension would be to consider a non spatially stationary mean function, as in [44].

Finally, these models can be extended to non-Gaussian processes, *e.g.* Student-t as in [156, 37] and applied to the unsupervised classification problem.

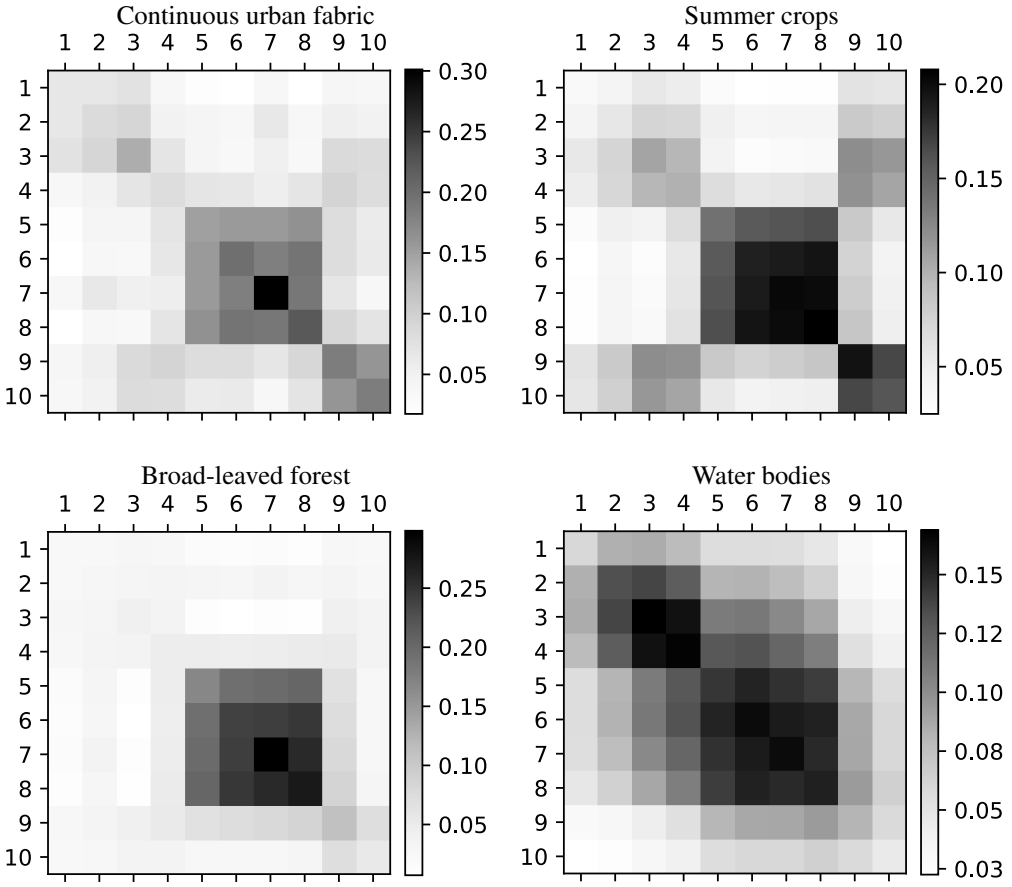


Figure 5.10: Estimated covariance matrices $\mathbf{A}_c\mathbf{A}_c^\top$ on four land-cover classes.

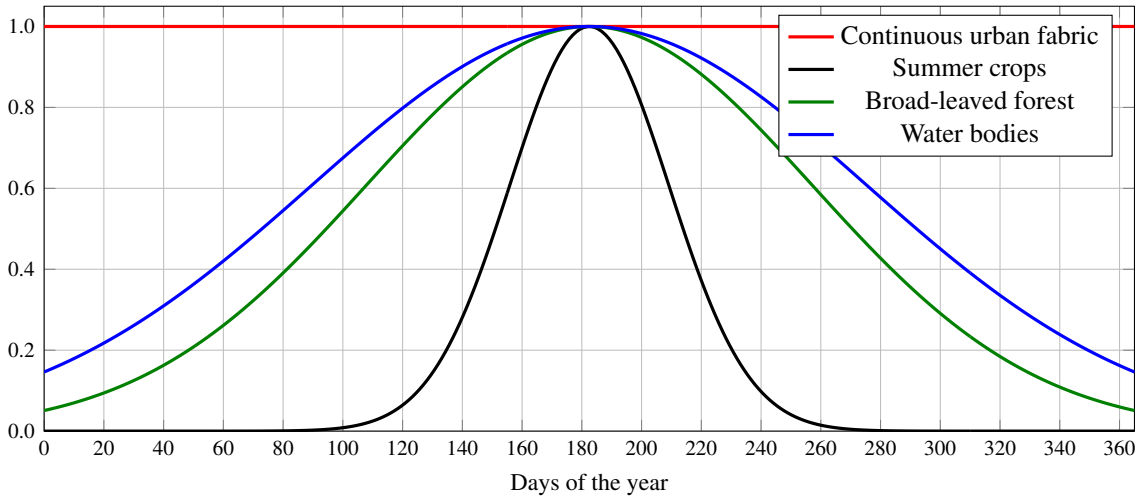


Figure 5.11: Normalized RBF kernels (5.8) centered at day 180: $K(t, 180) = \exp(-0.5(t-180)^2/h_c^2)$ computed on four classes.

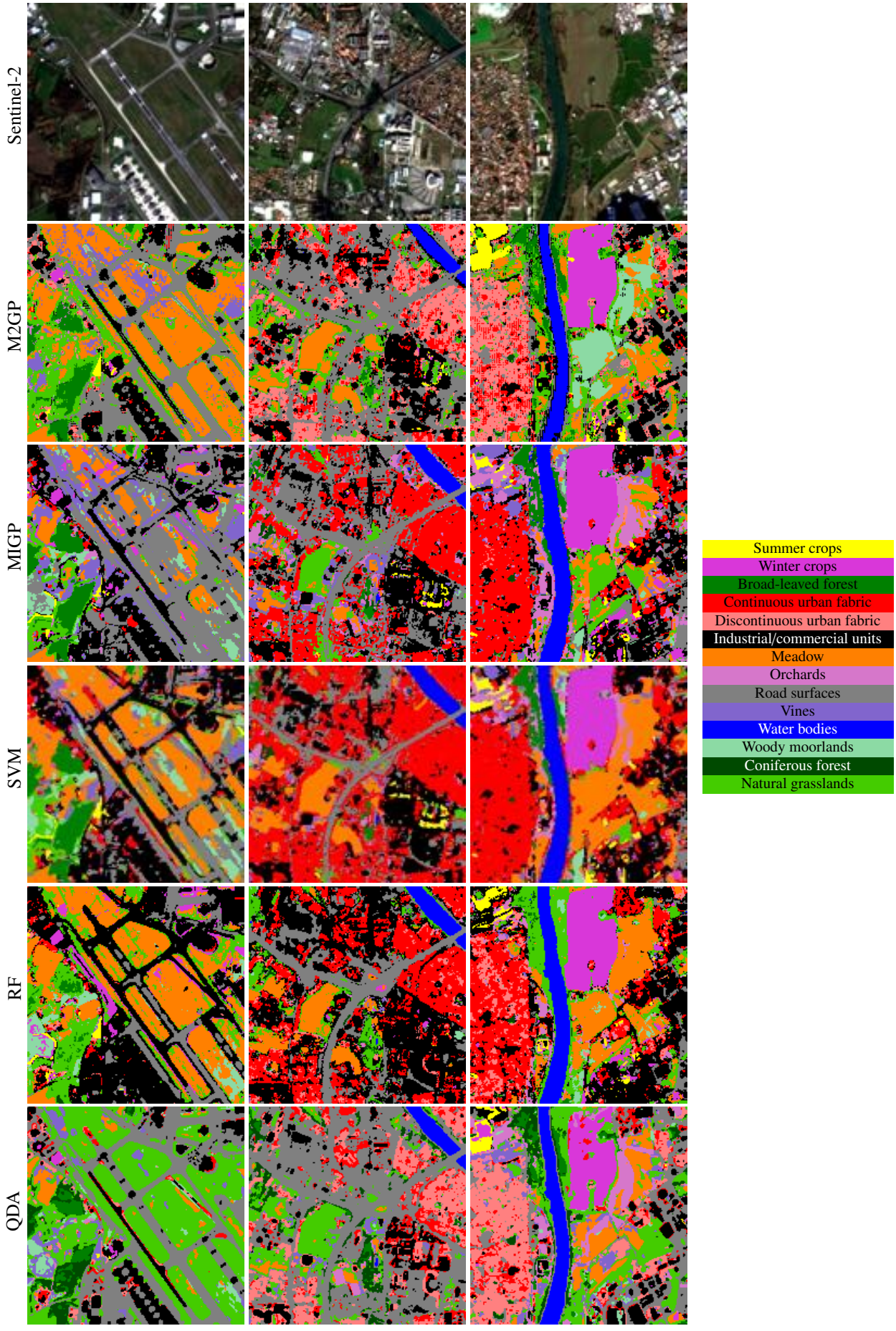


Figure 5.12: Three extracts of the classification maps obtained by QDA, RF, SVM, MIGP and M2PG methods.

Table 5.2: Mean F_1 score (mean(%) ± standard deviation) on the 10 independent data-sets.

	QDA	RF	SVM	MIGP	M2GP
Summer crops	96.5 ± 0.27	96.8 ± 0.45	95.6 ± 0.81	90.0 ± 0.83	95.9 ± 0.44
Winter crops	91.6 ± 0.48	94.0 ± 0.77	93.9 ± 0.66	80.2 ± 0.83	92.2 ± 0.64
Broad-leaved forest	77.4 ± 3.91	86.2 ± 2.35	85.3 ± 2.63	75.7 ± 5.03	81.5 ± 3.10
Cont. urban fabric	39.8 ± 6.18	58.0 ± 1.55	55.9 ± 2.49	21.4 ± 3.49	30.9 ± 5.51
Discont. urban fabric	58.5 ± 1.39	57.3 ± 3.44	40.2 ± 12.61	42.5 ± 3.17	54.5 ± 0.80
Ind. or commercial units	31.3 ± 2.14	60.3 ± 1.35	48.3 ± 4.05	27.4 ± 0.92	38.4 ± 2.34
Meadow	58.3 ± 4.14	64.8 ± 2.94	63.0 ± 3.17	43.3 ± 3.80	55.0 ± 4.19
Orchards	72.9 ± 4.05	81.0 ± 2.64	76.4 ± 3.11	51.9 ± 5.46	77.6 ± 3.58
Road surfaces	73.1 ± 1.92	87.1 ± 1.87	78.7 ± 2.79	54.2 ± 5.79	75.0 ± 2.06
Vines	71.1 ± 4.35	78.9 ± 6.86	78.5 ± 6.57	60.9 ± 7.61	71.7 ± 5.18
Water bodies	98.7 ± 0.35	99.4 ± 0.08	99.3 ± 0.10	84.9 ± 5.38	96.8 ± 0.84
Woody moorlands	23.9 ± 7.70	56.6 ± 3.50	56.1 ± 3.85	14.1 ± 5.52	10.6 ± 12.00
Coniferous forest	76.6 ± 7.24	86.9 ± 2.76	87.0 ± 2.56	61.2 ± 5.41	82.4 ± 6.61
Natural grasslands	29.8 ± 12.88	30.7 ± 16.90	19.4 ± 14.68	15.4 ± 7.86	20.6 ± 8.46
Average F_1 score	70.5 ± 0.75	78.2 ± 1.17	75.2 ± 1.11	57.4 ± 1.04	70.1 ± 0.43

5.8 Appendix - Proofs

Proof of Proposition 5.1 Let $\mathbf{Y} \sim \mathcal{MGP}_p(\mathbf{m}, K, \mathbf{A})$ and introduce $\mathbf{Y}^\star$ the $p \times q$ random matrix defined as $\mathbf{Y}^\star = (\mathbf{Y}(t_1), \ldots, \mathbf{Y}(t_q))$ where $(t_1, \ldots, t_q) \in \mathcal{T}^q$. From (5.1), we have $\mathbf{Y} = \mathbf{AW} + \mathbf{m}$ with $\mathbf{W} \sim \mathcal{IGP}_p(0, K)$. Let $\mathbf{W}^\star = (\mathbf{W}(t_1), \ldots, \mathbf{W}(t_q))$ be the associated $p \times q$ random matrix. Our first goal is to prove that $\mathbf{W}^\star \sim \mathcal{MN}_{p,q}(\mathbf{0}, \mathbf{\Sigma}, \mathbf{I}_p)$ or, equivalently, from (5.3), to prove that $\mathrm{vec}(\mathbf{W}^\star) \sim \mathcal{N}_{pq}(\mathbf{0}, \mathbf{\Sigma} \otimes \mathbf{I}_p)$. To this end, let us consider the random variable

$$S = \sum_{b=1}^{p} \sum_{j=1}^{q} \lambda_{b,j} \mathbf{W}^\star_{b,j},$$

and let us prove that S is a Gaussian random variable pour all $\lambda_{b,j} \in \mathbb{R}^{pq}$. Clear, one also has

$$S = \sum_{b=1}^{p} S_b, \text{ with } S_b := \sum_{j=1}^{q} \lambda_{b,j} \mathbf{W}^\star_{b,j} = \sum_{j=1}^{q} \lambda_{b,j} \mathbf{W}_b(t_j),$$

where $S_1, \ldots, S_p$ are independent centered Gaussian random variables with variance

$$\mathrm{var}(S_b) = \sum_{j=1}^{q} \sum_{j'=1}^{q} \lambda_{b,j} \lambda_{b,j'} \Sigma_{j,j'}.$$

As a consequence, S is a centered Gaussian random variable with variance

$$\mathrm{var}(S) = \sum_{b=1}^{p} \mathrm{var}(S_b) = \sum_{b=1}^{p} \sum_{b'=1}^{p} \sum_{j=1}^{q} \sum_{j'=1}^{q} \lambda_{b,j} \lambda_{b,j'} \Sigma_{j,j'} \times (\mathbf{I}_p)_{b,b'}.$$

As a conclusion, $\mathrm{vec}(\mathbf{W}^\star) \sim \mathcal{N}_{pq}(\mathbf{0}, \mathbf{\Sigma} \otimes \mathbf{I}_p)$ and thus $\mathbf{W}^\star \sim \mathcal{MN}_{p,q}(\mathbf{0}, \mathbf{\Sigma}, \mathbf{I}_p)$. Finally, $\mathbf{Y}^\star = \mathbf{AW}^\star + \mathbf{M} \sim \mathcal{MN}_{p,q}(\mathbf{M}, \mathbf{\Sigma}, \mathbf{AA}^\top)$, see [48, Example 1].

Proof of Lemma 5.1 Combining (5.4) and (5.9) yields that the density of $\mathbf{Y}^{i,\star}$ conditionally to $Z_i = c$ is given for all $i = 1, \ldots, n$ by

$$p_{i,c}(\mathbf{y}) = (2\pi)^{-pq_i/2} \det(\mathbf{\Sigma}^{c,i}(\boldsymbol{\theta}_c))^{-p/2} \det(\mathbf{A}_c \mathbf{A}_c^\top)^{-q_i/2} \times \exp\left(-\frac{1}{2} \mathrm{tr}\left[(\mathbf{A}_c \mathbf{A}_c^\top)^{-1} (\mathbf{y} - \boldsymbol{\alpha}_c \mathbf{B}^i)(\mathbf{\Sigma}^{c,i}(\boldsymbol{\theta}_c))^{-1} (\mathbf{y} - \boldsymbol{\alpha}_c \mathbf{B}^i)^\top \right] \right). \quad (5.22)$$

The likelihood is thus defined as

$$\prod_{c=1}^{C} \prod_{i|Z_i=c} p_{i,c}(\mathbf{Y}^{i,\star}),$$

and the negative log-likelihood can be written as

$$\mathcal{L} = -\sum_{c=1}^{C} \sum_{i|Z_i=c} \log p_{i,c}(\mathbf{Y}^{i,\star}) := \frac{1}{2} \sum_{c=1}^{C} \ell_c(\boldsymbol{\alpha}_c, \boldsymbol{\theta}_c, \mathbf{A}_c\mathbf{A}_c^\top) + \frac{p\log(2\pi)}{2} \sum_{c=1}^{C} \sum_{i|Z_i=c} q_i,$$

with, for all $c = 1, \ldots, C$,

$$\begin{aligned}\ell_c(\boldsymbol{\alpha}_c, \boldsymbol{\theta}_c, \mathbf{A}_c\mathbf{A}_c^\top) &= p \sum_{i|Z_i=c} \log\det(\boldsymbol{\Sigma}^{c,i}(\boldsymbol{\theta}_c)) + \sum_{i|Z_i=c} q_i \log\det(\mathbf{A}_c\mathbf{A}_c^\top) \\ &+ \sum_{i|Z_i=c} \operatorname{tr}\left[(\mathbf{A}_c\mathbf{A}_c^\top)^{-1}(\mathbf{Y}^{i,\star} - \boldsymbol{\alpha}_c\mathbf{B}^i)(\boldsymbol{\Sigma}^{c,i}(\boldsymbol{\theta}_c))^{-1}(\mathbf{Y}^{i,\star} - \boldsymbol{\alpha}_c\mathbf{B}^i)^\top\right].\end{aligned}$$

The conclusion follows.

Proof of Proposition 5.2 (*i*) Let us first consider the differential of $\ell_c(\boldsymbol{\alpha}_c, \boldsymbol{\theta}_c, \mathbf{A}_c\mathbf{A}_c^\top)$ w.r.t. $\boldsymbol{\alpha}_c$:

$$\begin{aligned}\mathrm{d}\ell_c(\boldsymbol{\alpha}_c) &:= \sum_{i|Z_i=c} \operatorname{tr}\left(\{\mathbf{A}_c\mathbf{A}_c^\top\}^{-1}\left[\mathrm{d}(\mathbf{Y}^{i,\star} - \boldsymbol{\alpha}_c\mathbf{B}^i)\{\boldsymbol{\Sigma}^{c,i}(\boldsymbol{\theta}_c)\}^{-1}(\mathbf{Y}^{i,\star} - \boldsymbol{\alpha}_c\mathbf{B}^i)^\top\right]\right) \\ &= -\sum_{i|Z_i=c} \operatorname{tr}\left(\{\mathbf{A}_c\mathbf{A}_c^\top\}^{-1}(\mathrm{d}\boldsymbol{\alpha}_c)\mathbf{B}^i\{\boldsymbol{\Sigma}^{c,i}(\boldsymbol{\theta}_c)\}^{-1}(\mathbf{Y}^{i,\star} - \boldsymbol{\alpha}_c\mathbf{B}^i)^\top\right) \\ &\quad - \sum_{i|Z_i=c} \operatorname{tr}\left(\{\mathbf{A}_c\mathbf{A}_c^\top\}^{-1}(\mathbf{Y}^{i,\star} - \boldsymbol{\alpha}_c\mathbf{B}^i)\{\boldsymbol{\Sigma}^{c,i}(\boldsymbol{\theta}_c)\}^{-1}(\mathbf{B}^i)^\top(\mathrm{d}\boldsymbol{\alpha}_c)^\top\right) \\ &= -2\sum_{i|Z_i=c} \operatorname{tr}\left(\{\mathbf{A}_c\mathbf{A}_c^\top\}^{-1}(\mathbf{Y}^{i,\star} - \boldsymbol{\alpha}_c\mathbf{B}^i)\{\boldsymbol{\Sigma}^{c,i}(\boldsymbol{\theta}_c)\}^{-1}(\mathbf{B}^i)^\top(\mathrm{d}\boldsymbol{\alpha}_c)^\top\right),\end{aligned}$$

by remarking that both terms are equal in view of the properties of the trace operator. Moreover, from Kronecker product properties [155, Theorem 8.12], one has

$$\begin{aligned}\mathrm{d}\ell_c(\boldsymbol{\alpha}_c) &= -2\sum_{i|Z_i=c} \operatorname{vec}(\mathrm{d}\boldsymbol{\alpha}_c)^\top\left(\mathbf{B}^i\{\boldsymbol{\Sigma}^{c,i}(\boldsymbol{\theta}_c)\}^{-1} \otimes \{\mathbf{A}_c\mathbf{A}_c^\top\}^{-1}\right)\operatorname{vec}(\mathbf{Y}^{i,\star} - \boldsymbol{\alpha}_c\mathbf{B}^i) \\ &= -2(\mathrm{dvec}(\boldsymbol{\alpha}_c))^\top \operatorname{vec}\left(\{\mathbf{A}_c\mathbf{A}_c^\top\}^{-1}\sum_{i|Z_i=c}(\mathbf{Y}^{i,\star} - \boldsymbol{\alpha}_c\mathbf{B}^i)\{\boldsymbol{\Sigma}^{c,i}(\boldsymbol{\theta}_c)\}^{-1}(\mathbf{B}^i)^\top\right).\end{aligned}$$

Interpreting the above result as a scalar product and using the "broad" definition of matrix derivative defined in [119], if follows:

$$\frac{\partial\ell_c(\boldsymbol{\alpha}_c, \boldsymbol{\theta}_c, \mathbf{A}_c\mathbf{A}_c^\top)}{\partial\boldsymbol{\alpha}_c} = -2\operatorname{vec}\left(\{\mathbf{A}_c\mathbf{A}_c^\top\}^{-1}\sum_{i|Z_i=c}(\mathbf{Y}^{i,\star} - \boldsymbol{\alpha}_c\mathbf{B}^i)\{\boldsymbol{\Sigma}^{c,i}(\boldsymbol{\theta}_c)\}^{-1}(\mathbf{B}^i)^\top\right).$$

Setting this partial derivative to zero yields

$$\sum_{i|Z_i=c}(\mathbf{Y}^{i,\star} - \boldsymbol{\alpha}_c\mathbf{B}^i)\{\boldsymbol{\Sigma}^{c,i}(\boldsymbol{\theta}_c)\}^{-1}(\mathbf{B}^i)^\top = 0,$$

or equivalently,

$$\boldsymbol{\alpha}_c = \left[\sum_{i|Z_i=c} \mathbf{Y}^{i,\star}\{\boldsymbol{\Sigma}^{c,i}(\boldsymbol{\theta}_c)\}^{-1}(\mathbf{B}^i)^\top\right]\left[\sum_{i|Z_i=c} \mathbf{B}^i\{\boldsymbol{\Sigma}^{c,i}(\boldsymbol{\theta}_c)\}^{-1}(\mathbf{B}^i)^\top\right]^{-1},$$

which is the desired result. Second, let us consider the differential of $\ell_c(\boldsymbol{\alpha}_c, \boldsymbol{\theta}_c, \mathbf{A}_c\mathbf{A}_c^\top)$ w.r.t. $\mathbf{A}_c\mathbf{A}_c^\top$:

$$\mathrm{d}\ell_c(\mathbf{A}_c\mathbf{A}_c^\top) = Q_c\mathrm{d}\log\det(\mathbf{A}_c\mathbf{A}_c^\top) + \mathrm{d}\operatorname{tr}\left(\mathbf{N}(\boldsymbol{\theta}_c)\{\mathbf{A}_c\mathbf{A}_c^\top\}^{-1}\right),$$

where $\mathbf{N}(\boldsymbol{\theta}_c) = \sum_{i|Z_i=c}(\mathbf{Y}^{i,\star} - \boldsymbol{\alpha}_c\mathbf{B}^i)\{\boldsymbol{\Sigma}^{c,i}(\boldsymbol{\theta}_c)\}^{-1}(\mathbf{Y}^{i,\star} - \boldsymbol{\alpha}_c\mathbf{B}^i)^\top$. From [155, Example 9.6], the associated partial derivative vanishes for

$$\mathbf{A}_c\mathbf{A}_c^\top = \frac{\mathbf{N}(\boldsymbol{\theta}_c)}{Q_c} = \frac{1}{Q_c}\sum_{i|Z_i=c}(\mathbf{Y}^{i,\star} - \boldsymbol{\alpha}_c\mathbf{B}^i)\{\boldsymbol{\Sigma}^{c,i}(\boldsymbol{\theta}_c)\}^{-1}(\mathbf{Y}^{i,\star} - \boldsymbol{\alpha}_c\mathbf{B}^i)^\top,$$

and the result is proved.

(*ii*) Introduce $\boldsymbol{\beta}^{c,i}(\boldsymbol{\alpha}_c, \boldsymbol{\theta}_c) = (\mathbf{Y}^{i,\star} - \boldsymbol{\alpha}_c\mathbf{B}^i)\{\boldsymbol{\Sigma}^{c,i}(\boldsymbol{\theta}_c)\}^{-1}$ and consider the kth coordinate of the gradient of $\ell_c(\boldsymbol{\alpha}_c, \boldsymbol{\theta}_c, \mathbf{A}_c\mathbf{A}_c^\top)$ w.r.t. $\boldsymbol{\theta}$:

$$\begin{aligned}
\frac{\partial \ell_c(\boldsymbol{\alpha}_c, \boldsymbol{\theta}_c, \mathbf{A}_c\mathbf{A}_c^\top)}{\partial \theta_k} &= p\sum_{i|Z_i=c}\frac{\partial}{\partial\theta_k}\log\det(\boldsymbol{\Sigma}^{c,i}(\boldsymbol{\theta}_c)) + \frac{\partial}{\partial\theta_k}\mathrm{tr}\left(\mathbf{N}(\boldsymbol{\theta}_c)\{\mathbf{A}_c\mathbf{A}_c^\top\}^{-1}\right)\\
&= p\sum_{i|Z_i=c}\mathrm{tr}\left(\{\boldsymbol{\Sigma}^{c,i}(\boldsymbol{\theta}_c)\}^{-1}\frac{\partial\boldsymbol{\Sigma}^{c,i}(\boldsymbol{\theta}_c)}{\partial\theta_k}\right)\\
&\quad - \sum_{i|Z_i=c}\mathrm{tr}\left((\mathbf{Y}^{i,\star} - \boldsymbol{\alpha}_c\mathbf{B}^i)\{\boldsymbol{\Sigma}^{c,i}(\boldsymbol{\theta}_c)\}^{-1}\frac{\partial\boldsymbol{\Sigma}^{c,i}(\boldsymbol{\theta}_c)}{\partial\theta_k}\{\boldsymbol{\Sigma}^{c,i}(\boldsymbol{\theta}_c)\}^{-1}(\mathbf{Y}^{i,\star} - \boldsymbol{\alpha}_c\mathbf{B}^i)^\top\{\mathbf{A}_c\mathbf{A}_c^\top\}^{-1}\right)\\
&= p\sum_{i|Z_i=c}\mathrm{tr}\left(\{\boldsymbol{\Sigma}^{c,i}(\boldsymbol{\theta}_c)\}^{-1}\frac{\partial\boldsymbol{\Sigma}^{c,i}(\boldsymbol{\theta}_c)}{\partial\theta_k}\right)\\
&\quad - \sum_{i|Z_i=c}\mathrm{tr}\left(\boldsymbol{\beta}^{c,i}(\boldsymbol{\alpha}_c, \boldsymbol{\theta}_c)^\top\{\mathbf{A}_c\mathbf{A}_c^\top\}^{-1}\boldsymbol{\beta}^{c,i}(\boldsymbol{\alpha}_c, \boldsymbol{\theta}_c)\frac{\partial\boldsymbol{\Sigma}^{c,i}(\boldsymbol{\theta})}{\partial\theta_k}\right)\\
&= \sum_{i|Z_i=c}\mathrm{tr}\left(\left[p\{\boldsymbol{\Sigma}^{c,i}(\boldsymbol{\theta}_c)\}^{-1} - \boldsymbol{\beta}^{c,i}(\boldsymbol{\alpha}_c, \boldsymbol{\theta}_c)^\top\{\mathbf{A}_c\mathbf{A}_c^\top\}^{-1}\boldsymbol{\beta}^{c,i}(\boldsymbol{\alpha}_c, \boldsymbol{\theta}_c)\right]\frac{\partial\boldsymbol{\Sigma}^{c,i}}{\partial\theta_k}(\boldsymbol{\theta}_c)\right).
\end{aligned}$$

The result is proved.

Proof of Proposition 5.3 Let $\mathbf{Y}^\star = (\mathbf{Y}(t_1), \ldots, \mathbf{Y}(t_q))$ be a $p\times q$ random matrix where, conditionally to $Z = c$, $\mathbf{Y} \sim \mathcal{MGP}_c(\boldsymbol{\alpha}_c\mathbf{b}, K_c, \mathbf{A}_c)$. Recall that Proposition 5.1 yields $\mathrm{vec}(\mathbf{Y}^\star) \sim \mathcal{N}_{pq}(\mathrm{vec}(\boldsymbol{\alpha}_c\mathbf{B}), \boldsymbol{\Sigma}^c(\boldsymbol{\theta}_c)\otimes\mathbf{A}_c\mathbf{A}_c^\top)$, where $\boldsymbol{\Sigma}^c(\boldsymbol{\theta}_c)$ is defined for all $(j, j') \in \{1, \ldots, q\}^2$ by $\boldsymbol{\Sigma}^c(\boldsymbol{\theta}_c)_{j,j'} = K_c(t_j, t_{j'}|\boldsymbol{\theta}_c)$ and $\mathbf{B} = (\mathbf{b}(t_1), \ldots, \mathbf{b}(t_q))$ is a $J\times q$ design matrix. Let $t^\dagger \in \mathcal{T}$ be an unobserved time, *i.e.* $t^\dagger \neq t_k$, for all $k \in \{1, \ldots, q\}$, and $\mathbf{k}_c(t^\dagger) = (K_c(t^\dagger, t_1|\boldsymbol{\theta}_c), \ldots, K_c(t^\dagger, t_q|\boldsymbol{\theta}_c))^\top$. Then, classical properties on conditional Gaussian random vectors (see for instance [17, p. 63]) entail that, conditionally to $Z = c$ and $\mathrm{vec}(\mathbf{Y}^\star) = \mathrm{vec}(\mathbf{y}^\star)$, $\mathbf{Y}(t^\dagger)$ follows the p-variate Gaussian distribution $\mathcal{N}_p(\boldsymbol{\mu}_c(t^\dagger, \mathbf{y}^\star), \boldsymbol{\Lambda}_c(t^\dagger))$ with, on the one hand

$$\begin{aligned}
\boldsymbol{\mu}_c(t^\dagger, \mathbf{y}^\star) &= \boldsymbol{\alpha}_c\mathbf{b}(t^\dagger) + [\mathbf{k}_c(t^\dagger)^\top\otimes\mathbf{A}_c\mathbf{A}_c^\top]\{\boldsymbol{\Sigma}^c(\boldsymbol{\theta}_c)\otimes\mathbf{A}_c\mathbf{A}_c^\top\}^{-1}\mathrm{vec}(\mathbf{y}^\star - \boldsymbol{\alpha}_c\mathbf{B})\\
&= \boldsymbol{\alpha}_c\mathbf{b}(t^\dagger) + [\mathbf{k}_c(t^\dagger)^\top\otimes\mathbf{A}_c\mathbf{A}_c^\top]\{\boldsymbol{\Sigma}^c(\boldsymbol{\theta}_c)^{-1}\otimes(\mathbf{A}_c\mathbf{A}_c^\top)^{-1}\}\mathrm{vec}(\mathbf{y}^\star - \boldsymbol{\alpha}_c\mathbf{B})\\
&= \boldsymbol{\alpha}_c\mathbf{b}(t^\dagger) + \left[\{\mathbf{k}_c(t^\dagger)^\top\boldsymbol{\Sigma}^c(\boldsymbol{\theta}_c)^{-1}\}\otimes\{(\mathbf{A}_c\mathbf{A}_c^\top)(\mathbf{A}_c\mathbf{A}_c^\top)^{-1}\}\right]\mathrm{vec}(\mathbf{y}^\star - \boldsymbol{\alpha}_c\mathbf{B})\\
&= \boldsymbol{\alpha}_c\mathbf{b}(t^\dagger) + \left[\{\mathbf{k}_c(t^\dagger)^\top\boldsymbol{\Sigma}^c(\boldsymbol{\theta}_c)^{-1}\}\otimes\mathbf{I}_p\right]\mathrm{vec}(\mathbf{y}^\star - \boldsymbol{\alpha}_c\mathbf{B})\\
&= \boldsymbol{\alpha}_c\mathbf{b}(t^\dagger) + \mathrm{vec}\left(\mathbf{I}_p(\mathbf{y}^\star - \hat{\boldsymbol{\alpha}}_c\mathbf{B})\{\mathbf{k}_c(t^\dagger)^\top\boldsymbol{\Sigma}^c(\boldsymbol{\theta}_c)^{-1}\}^\top\right)\\
&= \boldsymbol{\alpha}_c\mathbf{b}(t^\dagger) + (\mathbf{y}^\star - \boldsymbol{\alpha}_c\mathbf{B})\boldsymbol{\Sigma}^c(\boldsymbol{\theta}_c)^{-1}\mathbf{k}_c(t^\dagger),
\end{aligned}$$

and on the other hand,

$$\begin{aligned}
\boldsymbol{\Lambda}_c(t^\dagger) &= K_c(t^\dagger, t^\dagger|\boldsymbol{\theta}_c)\otimes\mathbf{A}_c\mathbf{A}_c^\top - [\mathbf{k}_c(t^\dagger)^\top\otimes\mathbf{A}_c\mathbf{A}_c^\top]\left\{\boldsymbol{\Sigma}^c\otimes\mathbf{A}_c\mathbf{A}_c^\top(\boldsymbol{\theta}_c)\right\}^{-1}[\mathbf{k}_c(t^\dagger)\otimes\mathbf{A}_c\mathbf{A}_c^\top]\\
&= K_c(t^\dagger, t^\dagger|\boldsymbol{\theta}_c)\otimes\mathbf{A}_c\mathbf{A}_c^\top - \left[(\mathbf{k}_c(t^\dagger)^\top\boldsymbol{\Sigma}^c(\boldsymbol{\theta}_c)^{-1})\otimes\mathbf{I}_p\right][\mathbf{k}_c(t^\dagger)\otimes\mathbf{A}_c\mathbf{A}_c^\top]\\
&= K_c(t^\dagger, t^\dagger|\boldsymbol{\theta}_c)\otimes\mathbf{A}_c\mathbf{A}_c^\top - \left(\mathbf{k}_c(t^\dagger)^\top\boldsymbol{\Sigma}^c(\boldsymbol{\theta}_c)^{-1}\mathbf{k}_c(t^\dagger)\right)\otimes(\mathbf{I}_p\mathbf{A}_c\mathbf{A}_c^\top)\\
&= \left[K_c(t^\dagger, t^\dagger|\boldsymbol{\theta}_c) - \mathbf{k}_c(t^\dagger)^\top\boldsymbol{\Sigma}^c(\boldsymbol{\theta}_c)^{-1}\mathbf{k}_c(t^\dagger)\right]\otimes\mathbf{A}_c\mathbf{A}_c^\top.
\end{aligned}$$

The result is proved.

Acknowledgment

The authors would like to thank S. Iovleff for his support and advices during the design of the model. The authors would also like to thank Y. Tanguy for his help when using the CNES computational resources to run the experiments presented in this paper.

CONCLUSION AND PERSPECTIVES

Summary

This thesis defines two statistical models for remote sensing data-sets, *i.e.* multi-spectral satellite images time-series (SITS) from Sentinel-2 constellation, to classify irregularly sampled time-series. To this end we defined a Gaussian process (GP) over functions. Any irregularly sampled time-series is a marginal from the process and is distributed according to a multi-variate Gaussian distribution.

We presented two models to overcome the classification problem or irregularly sampled SITS. The first model introduced in Chapter 4 assumes independence between spectral bands. It learns as many one-dimensional GP as the number of spectral bands. This model shows interesting behaviour, particularly in time-series reconstructions where the reconstructed reflectance value is as good as with non-parametric techniques, or sometimes better. The reconstruction also provides an uncertainty measure which is not always provided by other techniques. It has been applied to Sentinel-2 SITS and the provided information may be useful for the study of some phenomena (agricultural mowing period, *etc*). The classification scores are small when compared to state-of-the-art methods. The choice of various kernels and the combination of them have not shown significant improvements, the presented scores are the best we could obtain. Among others, we identified the independence assumption as too strong for the representation of SITS. The second model has been introduced in Chapter 5 to tackle the issue of independence between spectral bands. The model takes into account a linear dependency between the bands. The reconstruction quality has not been changed when compared to the previous model. However, the classification scores have been significantly improved. This model is slightly bellow state-of-the-art performances but the land use or land cover maps from irregularly sampled SITS is more reliable than the first approach.

To conclude, the issue of irregularly sampled time-series has been tackled. We were able to demonstrate the possibility to classify time-series without temporal re-sampling on large real-world remote sensing data-sets. The second model provides sufficient classification accuracy to rely on the produced maps. There are still some issues, in my opinion, some of them are linked to the Gaussian assumptions. Indeed, mislabelled data combined with variations within the same class linked to external factors (for example altitudes which modify the reflectance) yields an increasing noise learned by GP. This has been observed on areas with mountains where altitude changes the seasonality.

Future works

We identified three axes to extend the model and improve the performances of classification of irregularly sampled time-series. The first axis discusses non-Gaussian models, particularly Student-t processes. The second one discusses the three dimensional aspect of SITS by incorporating spatial variability. Finally the last axis concludes on more general covariance models with application to multi-sensor data fusion.

Non-Gaussian modelling

The Gaussian assumption done on the processes is, for SITS at least, limiting. The noise within SITS and the mislabelled data induce undesired outliers and it is known that the Gaussian distribution is very sensitive to outliers.

One usual solution within the past ten years is to assume a Student-t distribution. The Student-t distribution, or t-distribution, is more robust to outliers thanks to a heavier tail (see Figure 2.1). In [156] they proposed a definition of the Student-t process and showed interesting properties to model noisy functions. It may provide interesting reconstruction of SITS.

Additionally, the multi-variate Student-t processes has also been studied in [37] and is interesting to take into account spectral dependency of SITS.

Incorporating spatial variability

An other interesting work is to take into account the spatial information from SITS. Spatio-temporal [45] studies of SITS are of interest in Remote Sensing and already applied to Sentinel-2 data-set [66].

A perspective is to learn the spatial information within the mean of the Gaussian process as a function of time and space. It can by extended to a three-dimensional space by taking into account the altitude of the pixel location. It should scale to large data-sets as the pixels are seen independently and because the spatial information only changes the mean.

An other perspective is to use this information within the covariance operator (following the idea of three-order tensor distribution) but results in very complex estimation (M2GP is defined up to a multiplicative constant, a three-order tensor is defined up to two multiplicative constants). The use of large spatial areas (up to a complete Sentinel-2 tile) would result to a numerically non-invertible covariance matrix but restrictions to small areas may increase the classification accuracy.

More general covariance models

As introduced in Section 2.2.4, the LMC model (2.19) has been introduced in the context of Gaussian distributions over multi-outputs processes. LMC and its derived models issued from the geostatistics literature [75] and are used in a regression context, for example on time-series [38] (in this work, it is interesting to highlight that authors considered irregular time stamps but removed this specificity by considering one patient). LMC is a covariance model which assumes independent Gaussian distributions over latent processes resulting to a Gaussian posterior distribution [2, 172] over a linear combination over these latent processes. The number of latent processes may be different from the number of outputs. However, if all the latent processes share the same covariance operator, it yields the ICM model (2.20) which structure, based on the Kronecker product, generalizes M2GP. Inference on LMC, or ICM, is then also done using the *Maximum Likelihood Estimator* [2, Section 6.2] and is cubic w.r.t. the number of samples whereas M2GP scales linearly. LMC and ICM could be adapted to our context to scale to large data-sets.

An other idea is to consider a correlation between spectral bands which evolves with time input. The distribution is known as convolutional Gaussian processes [3, 33]. It may be seen as a two-dimensional Gaussian process with different applications. As an example, the M2GP model assumes that, despite an irregular sampling, the spectral bands are observed at the same time stamps. However, within the remote sensing literature, one key application is to use jointly multi-sensors to classify SITS, as an example one may consider the Sentinel-1 (SAR images) and Sentinel-2 (optical images) data fusion [136]. Indeed, Sentinel-1 provides complementary information as the information is not modified by the presence of clouds or shadows. The joint use implies an irregular sampling between spectral bands at the same time stamps. In [25] they studied the Dependent GP where the idea is to build latent features to retrieve the same assumptions as M2GP model or LMC.

In spite of that, estimation of convolutional GP is challenging. In [2, Section 3] the spectral covariance has been simplified to a Dirac delta function, resulting to the LMC. In [33] the authors considered a Gaussian kernel between spectral features and between temporal features and simplified the resulting kernel. In both cases, the numerical complexity is as complex as in LMC, adaptation to our context may require additional assumptions particularly for large data-sets.

Appendix A

About the estimation of the mean

This appendix describes the estimation of the mean. Section A.1 presents the study of the two matrix derivatives to assess the optimum found in Chapter 5. Section A.2 highlights computational issues encountered with random signals (irregularly and unevenly sampled).

A.1 On the broad and the narrow definitions of matrix derivative

This section discusses the two matrix derivatives defined in [119] for the computation of optimal mean in Chapter 5. Let $f : \mathcal{M}_{p,q}(\mathbb{R}) \to \mathbb{R}$ be a function which returns a scalar from a $p \times q$ matrix $\mathbf{X} \in \mathcal{M}_{p,q}(\mathbb{R})$, then the broad definition is defined as:

$$\mathrm{d}_{\mathbf{X}} f(\mathbf{X}) = \frac{\mathrm{d}f(\mathbf{X})}{\mathrm{d}\mathbf{X}}, \tag{A.1}$$

and the narrow definition as:

$$\mathrm{d}_{\mathbf{X}} f(\mathbf{X}) = \frac{\mathrm{d}f(\mathbf{X})}{\mathrm{dvec}(\mathbf{X})^\top}. \tag{A.2}$$

The first equation defines the derivative as a matrix of size $p \times q$ and the second definition returns a pq-dimensional vector.

It has been shown in [119] that the so-called *narrow* definition of the matrix derivative generalizes better the derivative on vectors. However, what follows proves that the two definitions give us the same result for computing the partial derivative with respect to the mean coefficient $\boldsymbol{\alpha}_c \in \mathcal{M}_{p,J}(\mathbb{R})$ in Proposition 5.2.

Proof. Firstly, recall that the differential w.r.t. $\boldsymbol{\alpha}_c$ is given by the Proposition 5.2, on one hand, as:

$$\mathrm{d}\ell_c(\boldsymbol{\alpha}_c) = -2 \sum_{i|Z_i=c} \mathrm{tr}\Big(\{\mathbf{A}_c\mathbf{A}_c^\top\}^{-1}(\mathbf{Y}^{i,\star} - \boldsymbol{\alpha}_c\mathbf{B}^i)\{\boldsymbol{\Sigma}^{c,i}(\boldsymbol{\theta}_c)\}^{-1}(\mathbf{B}^i)^\top(\mathrm{d}\boldsymbol{\alpha}_c)^\top\Big), \tag{A.3}$$

And, on the second hand, using Kronecker product properties [155, Theorem 8.12] gives:

$$\mathrm{d}\ell_c(\boldsymbol{\alpha}_c) = -2(\mathrm{dvec}(\boldsymbol{\alpha}_c))^\top \mathrm{vec}\left(\{\mathbf{A}_c\mathbf{A}_c^\top\}^{-1} \sum_{i|Z_i=c} (\mathbf{Y}^{i,\star} - \boldsymbol{\alpha}_c\mathbf{B}^i)\{\boldsymbol{\Sigma}^{c,i}(\boldsymbol{\theta}_c)\}^{-1}(\mathbf{B}^i)^\top\right). \tag{A.4}$$

Then:

1. Using the *broad* definition (A.1), and the matrix inner product ($\langle A, B\rangle \mapsto \mathrm{Tr}(B^\top A)$), from (A.3), $\mathrm{d}_{\boldsymbol{\alpha}_c}\ell_c(\boldsymbol{\alpha}_c) = 0$ is equivalent to:

$$\begin{aligned} &-2 \sum_{i|Z_i=c} \mathbf{B}^i\{\boldsymbol{\Sigma}^{c,i}(\boldsymbol{\theta}_c)\}^{-1}(\mathbf{Y}^{i,\star} - \hat{\boldsymbol{\alpha}}_c\mathbf{B}^i)^\top\{\mathbf{A}_c\mathbf{A}_c^\top\}^{-1} = 0 \\ \Leftrightarrow &\sum_{i|Z_i=c} \mathbf{B}^i\{\boldsymbol{\Sigma}^{c,i}(\boldsymbol{\theta}_c)\}^{-1}(\mathbf{Y}^{i,\star})^\top = \sum_{i|Z_i=c} \mathbf{B}^i\{\boldsymbol{\Sigma}^{c,i}(\boldsymbol{\theta}_c)\}^{-1}(\hat{\boldsymbol{\alpha}}_c\mathbf{B}^i)^\top. \end{aligned}$$

   Finally,

$$\hat{\boldsymbol{\alpha}}_c^\top = \left[\sum_{i|Z_i=c} \mathbf{B}^i\{\boldsymbol{\Sigma}^i(\boldsymbol{\theta}_c)\}^{-1}\mathbf{B}^{i\top}\right]^{-1} \sum_{i|Z_i=c} \mathbf{B}^i\{\boldsymbol{\Sigma}^i(\boldsymbol{\theta}_c)\}^{-1}(\mathbf{Y}^i)^\top. \tag{A.5}$$

2. Using the *narrow* definition (A.2), the derivative w.r.t. $\text{vec}(\boldsymbol{\alpha}_c)$ from (A.4) is:

$$\begin{aligned}\frac{\partial \ell_c(\boldsymbol{\alpha}_c)}{\partial \text{vec}(\boldsymbol{\alpha}_c)^\top} &= -2 \sum_{i|Z_i=c} \left(\mathbf{B}^i \{\boldsymbol{\Sigma}^i(\boldsymbol{\theta}_c)\}^{-1} \otimes \{\mathbf{A}_c \mathbf{A}_c^\top\}^{-1}\right) (\mathbf{y}_i - \text{vec}(\boldsymbol{\alpha}_c \mathbf{B}^i)) \\ &= \text{vec}\left(-2\{\mathbf{A}_c \mathbf{A}_c^\top\}^{-1} \sum_{i|Z_i=c} (\mathbf{Y}^{i,\star} - \boldsymbol{\alpha}_c \mathbf{B}^i)\{\boldsymbol{\Sigma}^i(\boldsymbol{\theta}_c)\}^{-1} \mathbf{B}^{i^\top}\right)\end{aligned}$$

Then we have

$$\begin{aligned}&\frac{\partial \ell_c(\boldsymbol{\alpha}_c)}{\partial \text{vec}(\boldsymbol{\alpha}_c)^\top} = 0 \\ \Leftrightarrow\ & -2\{\mathbf{A}_c \mathbf{A}_c^\top\}^{-1} \sum_{i|Z_i=c} (\mathbf{Y}^{i,\star} - \boldsymbol{\alpha}_c \mathbf{B}^i)\{\boldsymbol{\Sigma}^i(\boldsymbol{\theta}_c)\}^{-1} \mathbf{B}^{i^\top} = 0 \\ \Leftrightarrow\ & \sum_{i|Z_i=c} \mathbf{B}^i \{\boldsymbol{\Sigma}^i(\boldsymbol{\theta}_c)\}^{-1} (\mathbf{Y}^{i,\star})^\top = \sum_{i|Z_i=c} \mathbf{B}^i \{\boldsymbol{\Sigma}^i(\boldsymbol{\theta}_c)\}^{-1} (\boldsymbol{\alpha}_c \mathbf{B}^i)^\top,\end{aligned}$$

and yields to the same result as (A.5).

□

As a conclusion, considering one definition or the other returns the same optima in the *maximum likelihood estimation*.

A.2 On the conditioning of the design matrix

In what follow, we discuss the problem of time-series reconstruction. Whereas the reconstruction of regularly sampled time-series is well-known (for example using the Fourier decomposition, it is known that the frequency of the associated basis must not be too high w.r.t. the Shannon criteria), the theoretical study of irregularly sampled time-series involves fewer theoretical criteria to use properly the two models.

We defined in Appendix B (Section I) a necessary condition based on the injectivity of the design matrix $\mathbf{B}$ to compute the optimal mean. The matrix $\sum \mathbf{B}\{\boldsymbol{\Sigma}^{-1}\}\mathbf{B}$ is invertible if $\mathbf{B}$ is injective. This section goes further by defining a sufficient criteria to obtain a numerically stable inversion in Section A.2.1. This criteria involves the matrix $\mathbf{B}^\top\mathbf{B}$. Indeed if matrix $\mathbf{B}$ is injective, numerical issues occur when a temporal window is not observed by any sample, this is ilustrated in Section A.2.2.

A.2.1 Theoretical study

Let $\mathcal{S}_c = \{\mathbf{y}_i\}_{i=1}^{n_c}$ a set of *n i.i.d.* irregularly sampled time-series from a Gaussian process within the same class $c \in \{1, \ldots, C\}$. For any given i, following Appendix B, $\mathbb{E}(\mathbf{y}_i | z_i = c) = \mathbf{B}^i \boldsymbol{\alpha}_c = \mathbf{N}_i \mathbf{B}_c$ where $\mathbf{B}_c$ is the design matrix of size $T^c \times J$, with T^c the total number of unique time stamps within $\mathcal{S}_c$, and $\mathbf{N}_i$ of size $T_i \times T^c$ a matrix which selects the observed time stamps in $\mathbf{y}_i$.

From (4.8) (or (A.5)), $\hat{\boldsymbol{\alpha}}_c$ is obtained by inverting the matrix $\sum_{i|z_i=c} \mathbf{B}^{i^\top} \boldsymbol{\Sigma}^{i^{-1}} \mathbf{B}^i = \mathbf{B}_c^\top \mathbf{M}_c \mathbf{B}_c$, with the previous notations and $\mathbf{M}_c = \sum_{i|z_i=c} \mathbf{N}_i^\top \boldsymbol{\Sigma}^{i^{-1}} \mathbf{N}^i$ a non-singular matrix as presented in Appendix B, equation (2) (the number corresponds to the one of the included supplementary materials file). What follows aim to study the conditioning of this matrix to be numerically stable.

Proposition A.1. *Let* $\mathbf{B}$ *be a* $m \times n$ *injective matrix and* $\mathbf{M}$ *a* $m \times m$ *non-singular matrix. Then*

$$\text{cond}(\mathbf{B}^\top \mathbf{M} \mathbf{B}) \leq \text{cond}(\mathbf{M}) \text{cond}(\mathbf{B}^\top \mathbf{B}).$$

Proof of Proposition A.1. The proof is in two steps, firstly the upper bound of the largest eigenvalue is presented, then the lower bound for the smallest eigenvalue. Recall that the largest and the smallest eigenvalues correpond to

the upper and lower bounds of the Rayleigh quotient, we have:

$$
\begin{aligned}
\lambda_{\max}^{\mathbf{B}^\top\mathbf{M}\mathbf{B}} &= \sup_{\mathbf{x}} \frac{\mathbf{x}^\top\mathbf{B}^\top\mathbf{M}\mathbf{B}\mathbf{x}}{\mathbf{x}^\top\mathbf{x}} \\
&= \sup_{\mathbf{x}} \frac{\mathbf{x}^\top\mathbf{B}^\top\mathbf{M}\mathbf{B}\mathbf{x}}{\mathbf{x}^\top\mathbf{x}} \frac{\mathbf{x}^\top\mathbf{B}^\top\mathbf{B}\mathbf{x}}{\mathbf{x}^\top\mathbf{B}^\top\mathbf{B}\mathbf{x}}, \qquad \text{because } \mathbf{B} \text{ is injective,} \\
&= \sup_{\mathbf{x}} \frac{\mathbf{x}^\top\mathbf{B}^\top\mathbf{M}\mathbf{B}\mathbf{x}}{\mathbf{x}^\top\mathbf{B}^\top\mathbf{B}\mathbf{x}} \frac{\mathbf{x}^\top\mathbf{B}^\top\mathbf{B}\mathbf{x}}{\mathbf{x}^\top\mathbf{x}} \\
&= \sup_{\mathbf{x}} \frac{(\mathbf{B}\mathbf{x})^\top\mathbf{M}\mathbf{B}\mathbf{x}}{(\mathbf{B}\mathbf{x})^\top\mathbf{B}\mathbf{x}} \frac{\mathbf{x}^\top\mathbf{B}^\top\mathbf{B}\mathbf{x}}{\mathbf{x}^\top\mathbf{x}} \\
&\leq \sup_{\mathbf{y}} \frac{\mathbf{y}^\top\mathbf{M}\mathbf{y}}{\mathbf{y}^\top\mathbf{y}} \sup_{\mathbf{x}} \frac{\mathbf{x}^\top\mathbf{B}^\top\mathbf{B}\mathbf{x}}{\mathbf{x}^\top\mathbf{x}} \\
&\leq \lambda_{\max}^{\mathbf{M}} \lambda_{\max}^{\mathbf{B}^\top\mathbf{B}}.
\end{aligned}
$$

Finally,

$$\lambda_{\max}^{\mathbf{B}^\top\mathbf{M}\mathbf{B}} \leq \lambda_{\max}^{\mathbf{M}} \lambda_{\max}^{\mathbf{B}^\top\mathbf{B}}.$$

With the lower bound of the Rayleigh quotient, it is straightforward that:

$$\lambda_{\min}^{\mathbf{B}^\top\mathbf{M}\mathbf{B}} \geq \lambda_{\min}^{\mathbf{M}} \lambda_{\min}^{\mathbf{B}^\top\mathbf{B}},$$

and yields to the expected upper bound. □

In (4.8), a sufficient condition to compute $\boldsymbol{\alpha}_c$ numerically is to consider both a kernel and the design matrix such that $\mathbf{B}_c^\top\mathbf{B}_c$ that are simple to invert.

A.2.2 Illustration on a toy data-set

This toy example aims to present the estimation of the mean for an ill-conditioned design matrix. To this end we define three means, one for each class ($p = 1$ spectral band) for the Gaussian processes with different behaviors: one constant line, one generated using Fourier functions and one generated using polynomial functions. The three means are illustrated in Figure A.1. Besides, we define a RBF kernel (with a length scale of 63 days and a multiplicative constant of 1) with an additive white noise (set to 0.5). The noise may is not representative of the Sentinel-2 SITS, neither the range of the mean values, but our aim is to observe the estimation of the mean.

Afterwards, a total of $n_c = 350$ GP samples are generated from the distribution for each class, and are marginalized such as a temporal window remains unobserved. The size of each marginal is randomly selected according to a discrete uniform distribution and ranges between 3 and 13. The simulation is replicated 100 times by generating new samples/

Finally, five simulations settings are computed with an increasing size of the unobserved temporal window, starting from day 100 and ending at 100 + N day with N such as: $N_1 = 10$ days, $N_2 = 20$ days, $N_3 = 50$ days, $N_4 = 100$ days and $N_5 = 170$ days.

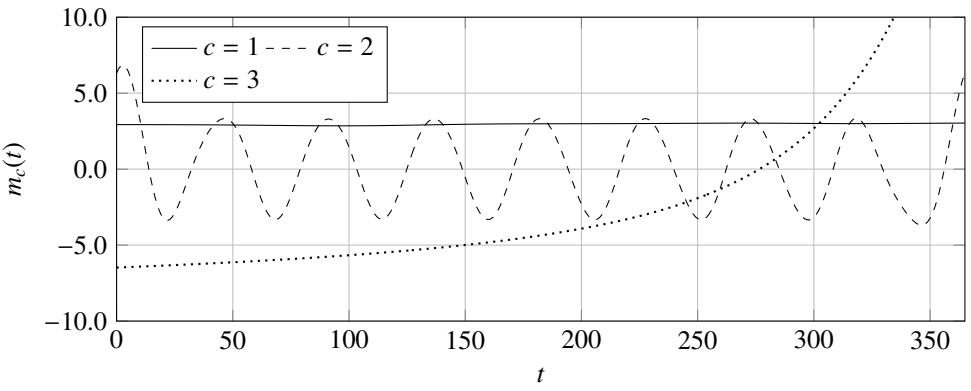


Figure A.1: Means of the 3 classes.

Model settings and estimation

The MIMGP model is learned using a Fourier base with 19 basis functions, a Matérn kernel (smoother than RBF [67] with parameter $\nu = 2.5$) and an additive white noise. The estimated means for one run are reported in Figures A.2, A.3 and A.4 (one figure per class).

The first class and the second class are well estimated for a size of N sufficiently small. Indeed, these two classes are generated according to the model. The third class is more complex, the reconstruction is poor as the basis functions are not chosen correctly. However, the three classes present a huge mis-estimation of mean for large unobserved windows which can sometimes shift the estimated mean where the data are observed (see class 3 on Figure A.4).

The same experience holds for a fixed unobserved temporal window and an increasing number of basis functions.

Condition number within class 1

The condition numbers for the quantities in Proposition A.1 are computed within the first class ($c = 1$) on 100 replications. The mean condition number values are reported in Table A.1.

Table A.1: Averaged log-condition numbers for the computation of the mean within class 1.

N	$\text{cond}(\mathbf{B}_c^\top \mathbf{M}_c \mathbf{B}_c)$	$\text{cond}(\mathbf{B}_c^\top \mathbf{B}_c)$	$\text{cond}(\mathbf{M}_c)$
10	4.20	1.32	5.96
20	4.14	2.25	5.93
50	5.96	6.27	5.78
100	14.18	14.33	5.81
170	27.71	27.55	6.10

The inequality in Proposition A.1 is verified. No lower bound has been found but the upper bound becomes broad with the condition number of matrix $\mathbf{B}_c^\top \mathbf{B}_c$, where $\mathbf{B}_c$ is the design matrix within class c.

Despite no improvements on the classification accuracy, a regularized approach can improve the conditioning of the matrix to be inverted (see [169, Section 3.1]). Nevertheless, a regularization on the mean has to take into account the complete temporal window of interest ($\mathcal{T}$) with an increasing number of unobserved time-stamps as the number of basis functions increases to avoid mis-estimation.

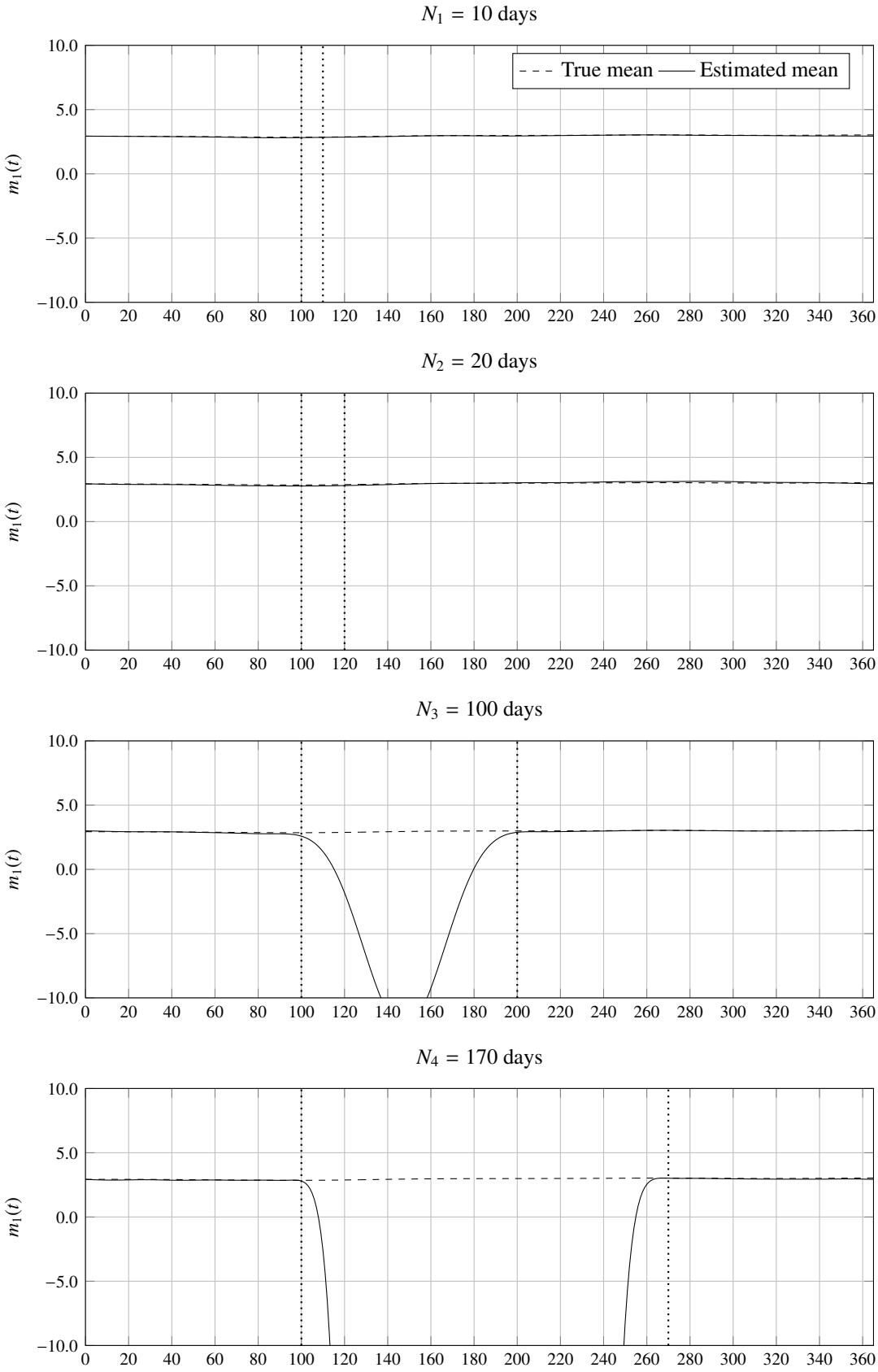


Figure A.2: Estimated mean function for the 1st class when the average number of time stamps and the number of basis functions J are identical but the size of the unobserved window is increasing ($N_4 > N_3 > N_2 > N_1$), delimited by the dotted vertical lines.

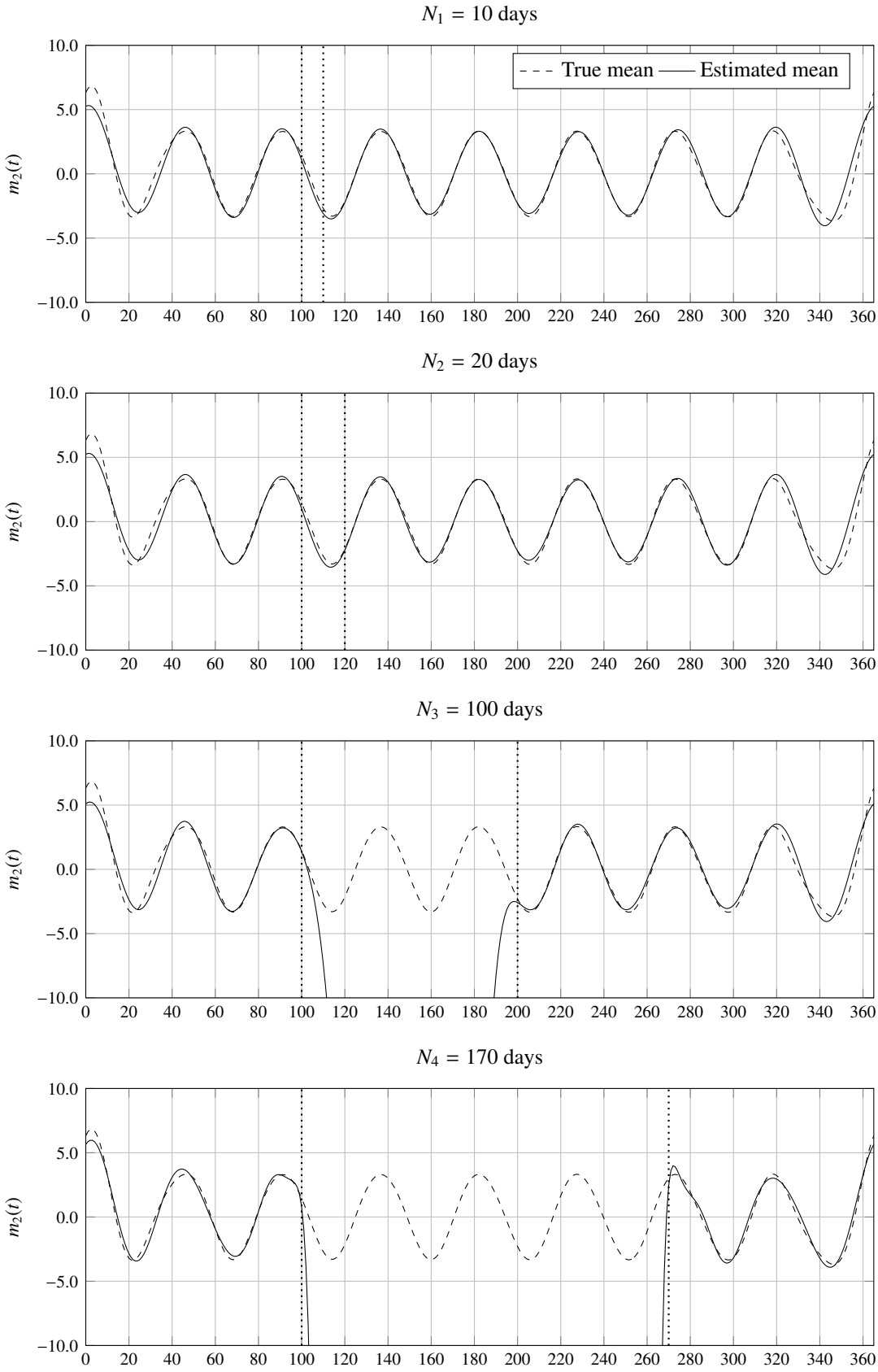


Figure A.3: Estimated mean function for the 2nd class when the average number of time stamps and the number of basis functions J are identical but the size of the unobserved window is increasing ($N_4 > N_3 > N_2 > N_1$), delimited by the dotted vertical lines.

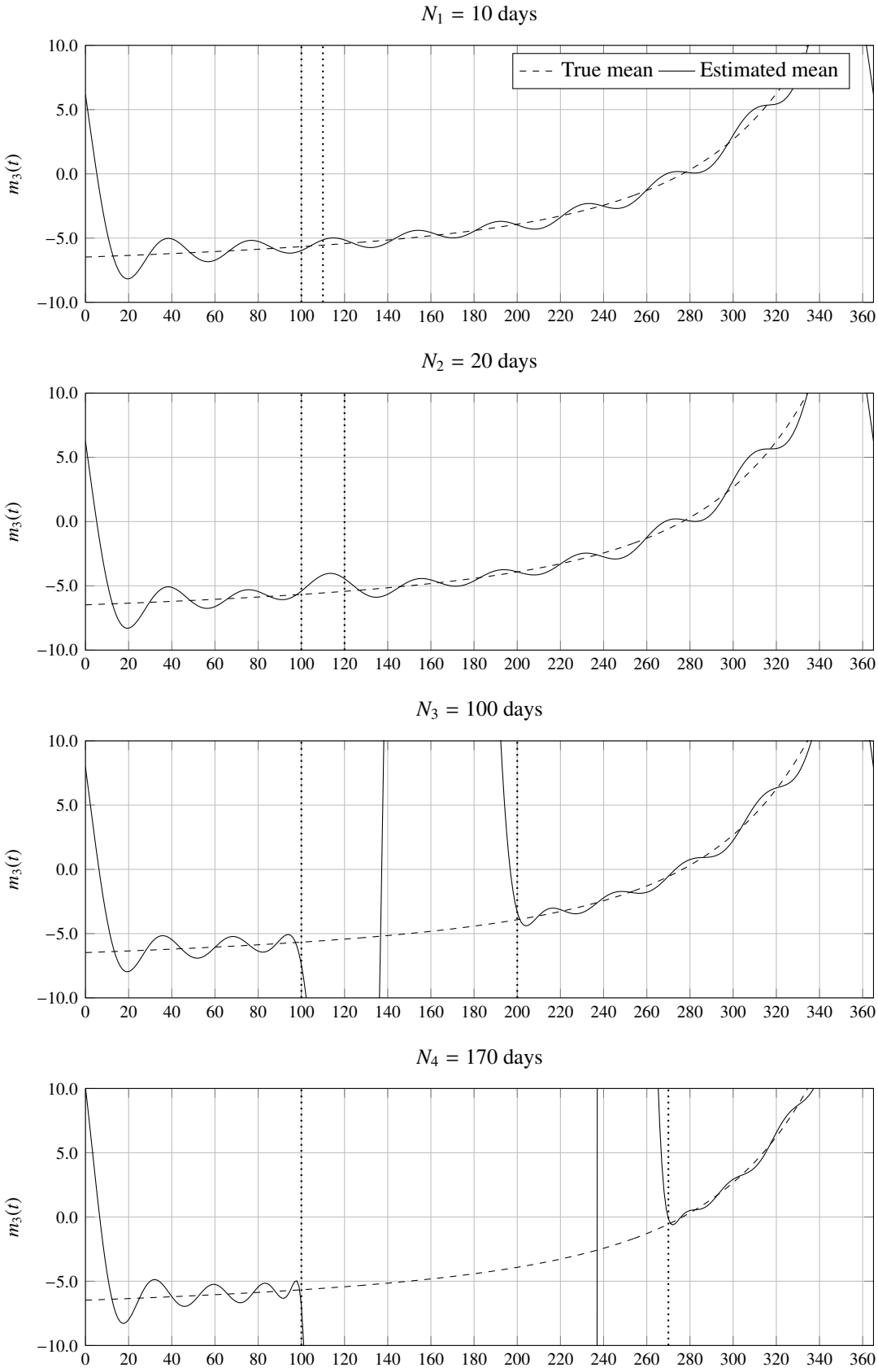


Figure A.4: Estimated mean function for the 3rd class when the average number of time stamps and the number of basis functions J are identical but the size of the unobserved window is increasing ($N_4 > N_3 > N_2 > N_1$), delimited by the dotted vertical lines.

APPENDIX B

SUPPLEMENTARY MATERIALS OF CHAPTER 4

This appendix describes the supplementary materials of [41] (compiled before inclusion, pagination does not follow the one of this manuscript neither the references). They contain additional details related to the main paper.

Supplementary Materials
Joint Supervised Classification and Reconstruction of Irregularly Sampled Satellite Image Times Series

Alexandre Constantin, *Student Member, IEEE,* Mathieu Fauvel, *Senior Member, IEEE,* and Stéphane Girard

CONTENTS

I. ESTIMATION OF THE PARAMETERS

The identifiability of the model depends on the ability to compute $\boldsymbol{\alpha}_{b,c}$. In (1) below, we remind that the optimal parameter is computed using the inverse of the matrix $\mathbf{G}_c$, defined as the sum of terms involving the design and precision matrices:

$$\boldsymbol{\alpha}_{b,c} = \mathbf{G}_c^{-1}\left[\sum_{i|Z_i=c} \mathbf{B}^{i^\top}\boldsymbol{\Sigma}^i(\boldsymbol{\theta}_{b,c})^{-1}\mathbf{y}_{i,b}\right], \qquad \text{where } \mathbf{G}_c = \sum_{i|Z_i=c} \mathbf{B}^{i^\top}\boldsymbol{\Sigma}^i(\boldsymbol{\theta}_{b,c})^{-1}\mathbf{B}^i. \tag{1}$$

The goal of this Section is to establish a necessary and sufficient condition for $\mathbf{G}_c$ to be non-singular. Let T^c be the total number of unique temporal acquisitions for a given class c. Let us also introduce the $T_i \times T^c$ matrix $\mathbf{N}_i$ (with $T_i \leq T_c$ for all i such as $z_i = c$) and the global design $T^c \times J$ matrix $\mathbf{B}_c$ such that $\mathbf{B}^i = \mathbf{N}_i\mathbf{B}_c$. The matrix $\mathbf{N}_i$ is composed of ones and zeros to select the observed samples in signal i from the global design matrix. Then, the matrix $\mathbf{G}_c$ can be rewritten as

$$\mathbf{G}_c = \sum_{i|Z_i=c} \mathbf{B}_c^\top\mathbf{N}_i^\top\boldsymbol{\Sigma}^i(\boldsymbol{\theta}_{b,c})^{-1}\mathbf{N}_i\mathbf{B}_c = \mathbf{B}_c^\top\left[\sum_{i|Z_i=c} \mathbf{N}_i^\top\boldsymbol{\Sigma}^i(\boldsymbol{\theta}_{b,c})^{-1}\mathbf{N}_i\right]\mathbf{B}_c = \mathbf{B}_c^\top\mathbf{M}_c\mathbf{B}_c, \tag{2}$$

where we set $\mathbf{M}_c = \sum_{i|Z_i=c}\mathbf{N}_i^\top\boldsymbol{\Sigma}^i(\boldsymbol{\theta}_{b,c})^{-1}\mathbf{N}_i$. In view of (2), $\mathbf{G}_c$ is non-singular if and only if $\mathbf{M}_c$ is non-singular and $\mathbf{B}_c$ is injective.

1) Let us first prove that $\mathbf{M}_c$ is non-singular. To this end, consider $\mathbf{v}^* \in \mathbb{R}^{T^c}$ such that $\mathbf{M}_c\mathbf{v}^* = \mathbf{0}$. This implies that $\mathbf{v}^{*\top}\mathbf{M}_c\mathbf{v}^* = 0$. From the definition of $\mathbf{M}_c$, and since $\boldsymbol{\Sigma}^i(\boldsymbol{\theta}_{b,c})^{-1}$ is a positive definite matrix, $\mathbf{v}^{*\top}\mathbf{M}_c\mathbf{v}^*$ is a sum of non-negative terms. Consequently, this entails that $(\mathbf{N}_i\mathbf{v})^{*\top}\boldsymbol{\Sigma}^i(\boldsymbol{\theta}_{b,c})^{-1}(\mathbf{N}_i\mathbf{v}^*) = 0$ for all i such that $z_i = c$. Using again the positive definiteness of $\boldsymbol{\Sigma}^i(\boldsymbol{\theta}_{b,c})^{-1}$ yields $\mathbf{N}_i\mathbf{v}^* = \mathbf{0}$ for all i such that $z_i = c$ or equivalently $\mathbf{N}\mathbf{v}^* = \mathbf{0}$ where $\mathbf{N} := \left[\mathbf{N}_1^\top, \ldots, \mathbf{N}_{n_c}^\top\right]^\top$ and n_c is the number of signals in class c. Up to a lines permutation, $\mathbf{N}$ can be rewritten as $\mathbf{N} = \left[\mathbf{I}_{T^c}, \tilde{\mathbf{N}}\right]^\top$ where $\mathbf{I}_{T^c}$ denotes the $T^c \times T^c$ identity matrix. Then, $\mathbf{N}\mathbf{v}^* = \mathbf{0}$ implies $\mathbf{v}^* = \mathbf{0}$. Hence, the result.
2) Second, $\mathbf{B}_c$ is injective if and only if $\mathbf{B}_c$ has full rank, *i.e.* $\text{rank}(\mathbf{B}_c) \geq J$.

As a conclusion, a necessary and sufficient condition for the existence of the inverse in (1) is $\text{rank}(\mathbf{B}_c) \geq J$. This condition involves constraints both on the orthogonal basis and on the temporal acquisition points. It can be checked numerically on each experiment. Finally, let us note that, in all situations, $T^c \geq J$ is a necessary condition.

This work is supported by the French National Research Agency in the framework of the Investissements dAvenir program (ANR-15-IDEX-02) and by the Centre National dEtudes Spatiales (CNES).

A. Constantin and S. Girard are with Université Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK, 38000 Grenoble, France (e-mail: alexandre.constantin@inria.fr; stephane.girard@inria.fr).

M. Fauvel is with CESBIO, Université de Toulouse, CNES/CNRS/INRAe/IRD/UPS, 31000 Toulouse, France (e-mail: mathieu.fauvel@inrae.fr).

II. Numerical implementation and Convergence

The model has been implemented with Python and inherits from Scikit Learn the Gaussian Process and Kernel classes [1]. The optimization problem is solved using L-BGFS-B subroutine [2] that allows for additional constraints or bounds (*i.e.* positivity or minimal/maximal values) on the covariance function parameters.

Minimization of (7) in the main document is known to converge to a local minimum, which depends on initial values of the hyperparameters. Figure 2 illustrates the evolution of the parameter h of the RBF kernel as well as the marginal log-likelihood with respect to the number of iterations. Here, a Fourier basis of dimension 19 was used to estimate the mean function.

III. Mean and Covariance functions

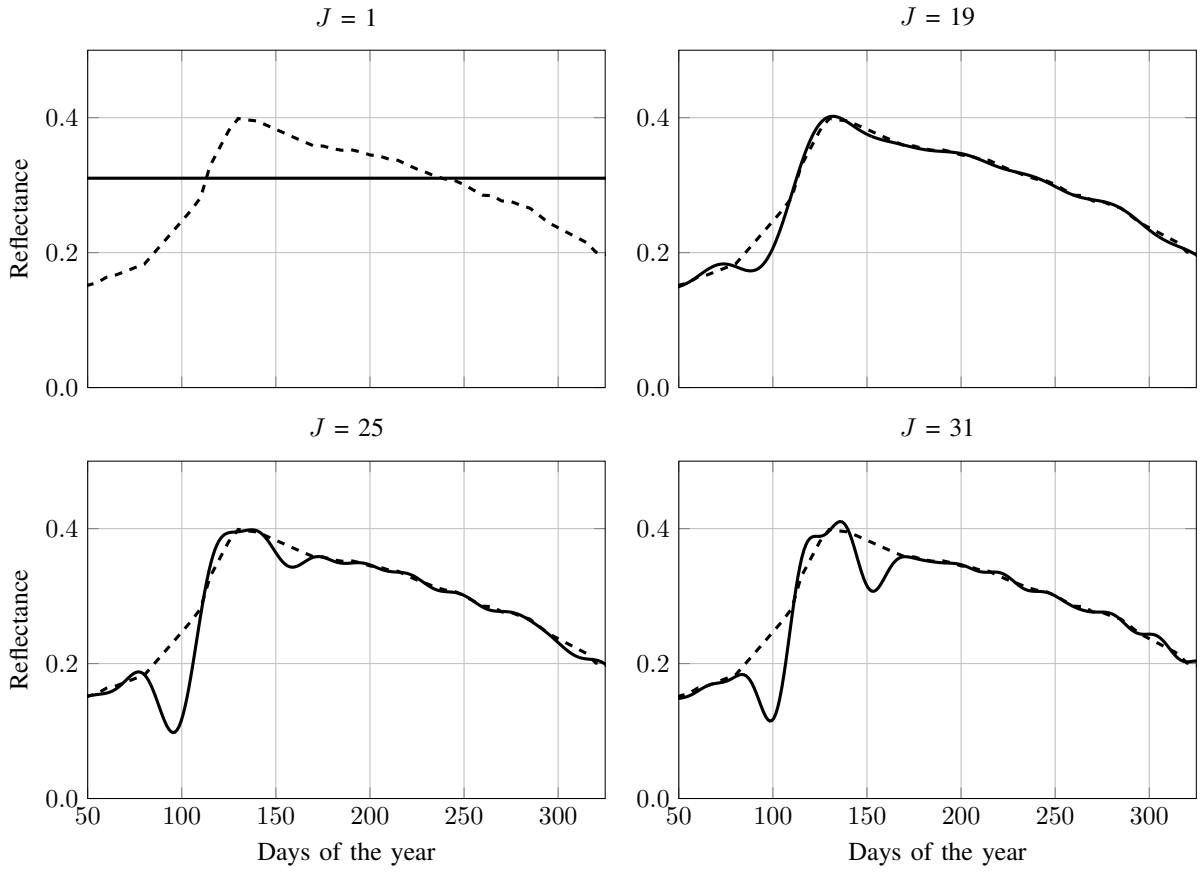


Fig. 1: Reconstructed mean near-infrared (IR) time-series from broad-leaved forest class with parameters $\boldsymbol{\alpha}$ (continuous line) estimated on the raw data, for different dimensions of basis. The dashed line corresponds to the gap-filled data mean from QDA on the resampled grid.

Figure 1 represents the estimated mean function in the infra-red for class "broad-leaved forest" for different dimensions of the Fourier basis. For $J > 25$, some oscillations appear in the reconstruction, due to a possible overfit. $J = 19$ seems to be the most appropriate dimension, even though, in terms of classification, $J = 25$ yields a slightly better accuracy score. In practice, a compromise should be found between a good reconstruction and a better classification score (see Figure 8 in the main document), depending on the final objective.

Figure 3 presents the RBF kernel computed as $\exp\left\{-\frac{(t-180)^2}{2h^2}\right\}$, $t \in \{1, ...365\}$ on the infrared wavelength from three different classes. It appears that the continuous urban fabric has a higher temporal correlation than Broad-Leaved forests and Water bodies. Indeed, natural elements reflectance such as vegetation evolve along the year (e.g. phenology) while man-made material's reflectance does not evolve along the year and thus exhibits longer temporal correlation.

IV. Consequences of the independence assumption

To illustrate the impact of the independence assumption on the classification accuracy, two versions of Quadratic Discriminant Analysis (QDA) have been implemented and compared. The first version, referred as u-QDA is based on the assumption that all the wavelengths are uncorrelated. More specifically, a QDA model is fitted independently to each wavelength. The likelihood

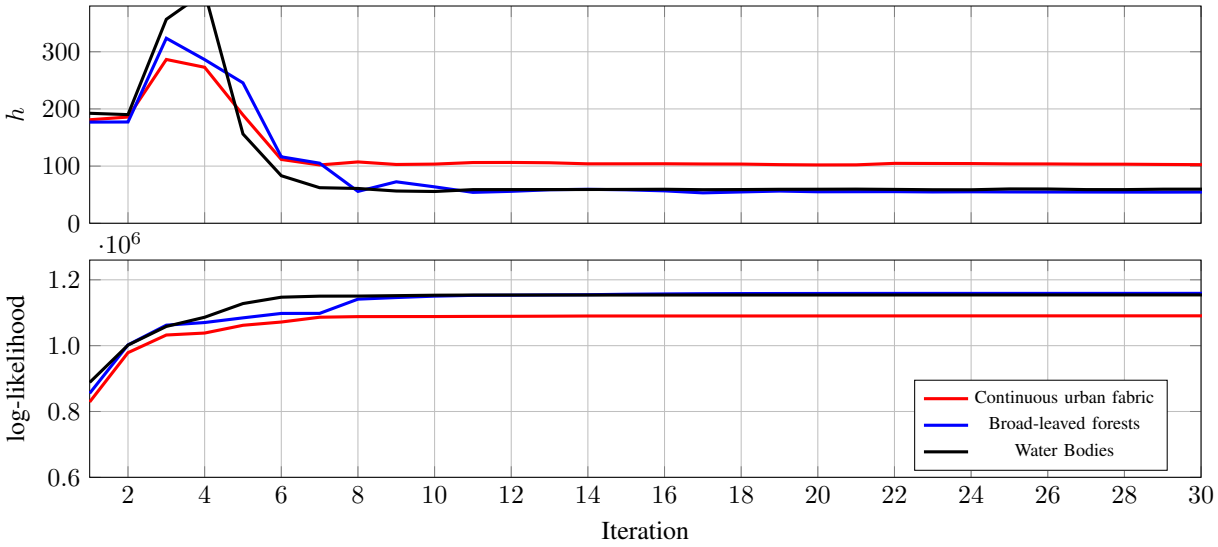


Fig. 2: Evolution of the length-scale $h \in \boldsymbol{\theta}$, given in (4) in the main document, and associated log-likelihood for the IR wavelength and three different areas: artificial (continuous urban fabric), semi-natural (broad-leaved forests) and water (water bodies) areas.

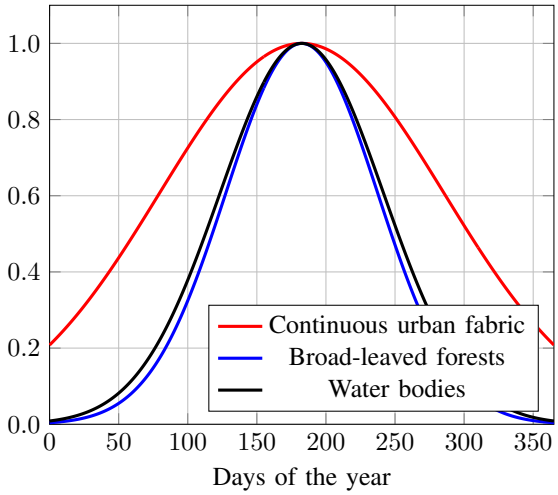


Fig. 3: Estimated RBF kernels for the IR wavelength and three different areas: artificial (continuous urban fabric), semi-natural (broad-leaved forests) and water (water bodies) areas It shows the correlation between day of the year 180 and the other days, from day 1 to day 360.

is computed as the product of the likelihoods associated with each wavelength, and the posterior probability is obtained using Bayes' rule as usual. The second version is the classical QDA method presented in the main document. The results are reported in Table I. It appears that the classification accuracy of u-QDA is significantly smaller than the one of the usual QDA method. Therefore, one can conclude that the independence assumption has a significant impact on the performance of the classifiers.

TABLE I: Averaged mean F_1 score (mean(%) $\pm$ standard deviation) associated with u-QDA and QDA computed on three tiles.

	u-QDA	QDA
T31TCJ	26.2 ± 3.1	36.2 ± 2.5
T31TDN	21.3 ± 3.6	30.5 ± 2.8
T31TGK	29.2 ± 2.1	38.9 ± 2.1

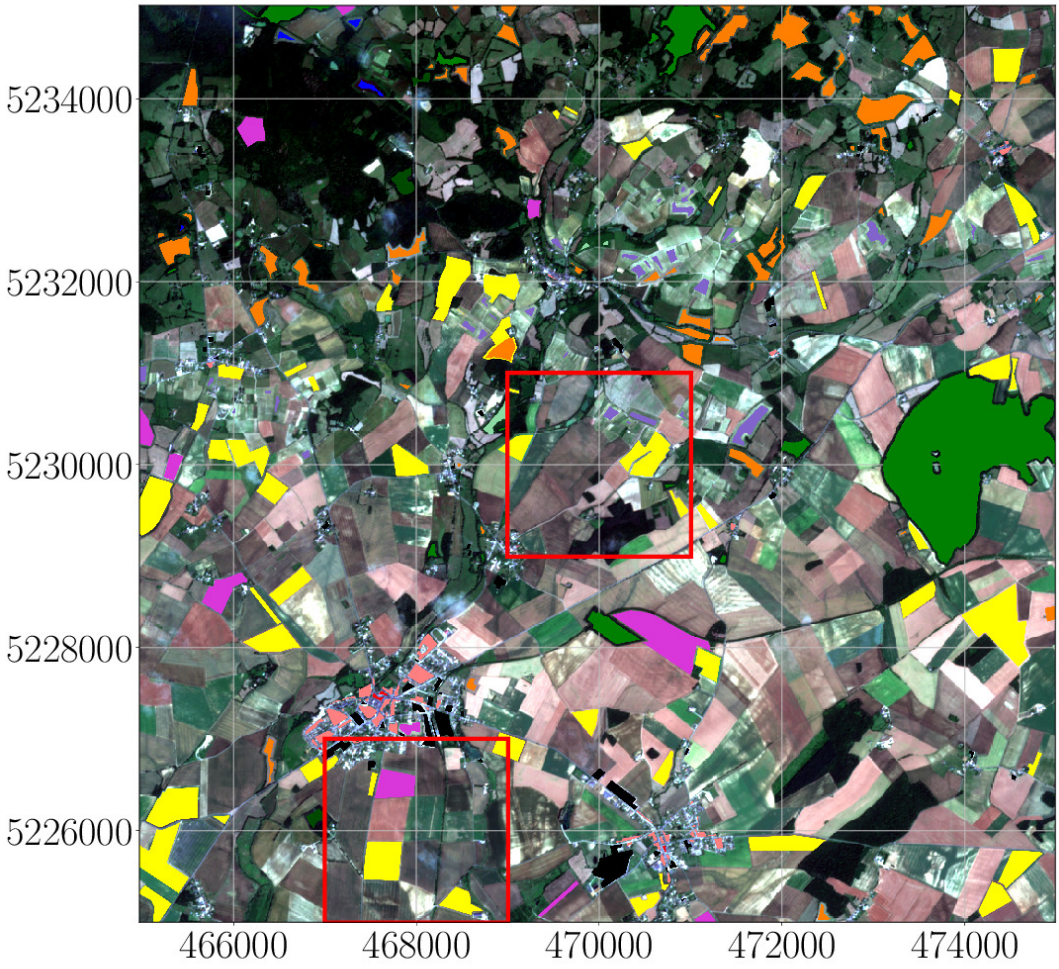


Fig. 4: Reference data extraction where each color represents an extract of land cover classes. The color code is described in Figure 7. The axes correspond to the geographical coordinates.The red continuous squares correspond to zoomed areas of the classification thematic, and reconstruction, maps provided in Figure 7 and Figure 15 respectively. The background image is an extract from Sentinel-2 optical images.

V. Supervised Classification

This section provides an extract of the thematic classification maps obtained with RF, MIMGP and SVM on tiles T31TCJ (Figure 6), T31DN (Figure 7) and T31TGK (Figure 8). The original tiles are too large to be correctly displayed, we thus choose to extract the center of each tile. All extracts have 1000×1000 pixels. We also provide a zoom (200×200 pixels) on several extracts.

The first extract from T31TCJ is presented in Figure 7 of the main document. The second and third extracts from, respectively, T31TDN and T31TGK, are presented in Figures 4 and 5.

The F_1 scores for each class and for each tiles, averaged over the 10 independent runs, are provided in Tables II, III and IV.

VI. Time-series reconstruction

This Section displays additional temporal reconstructions.

Figures 9 and 10 present the reconstructed time-series for all days in year 2018 for two pixels on four wavelengths of different classes, summer crops and broad-leaved forests.

Figure 11 compares reconstructions obtained with MIMGP model and Whittaker smoother on the 10 wavelengths. The more intense the colors are (yellow) on the line of equation $y = x$, the better the reconstruction is. Both methods have higher density

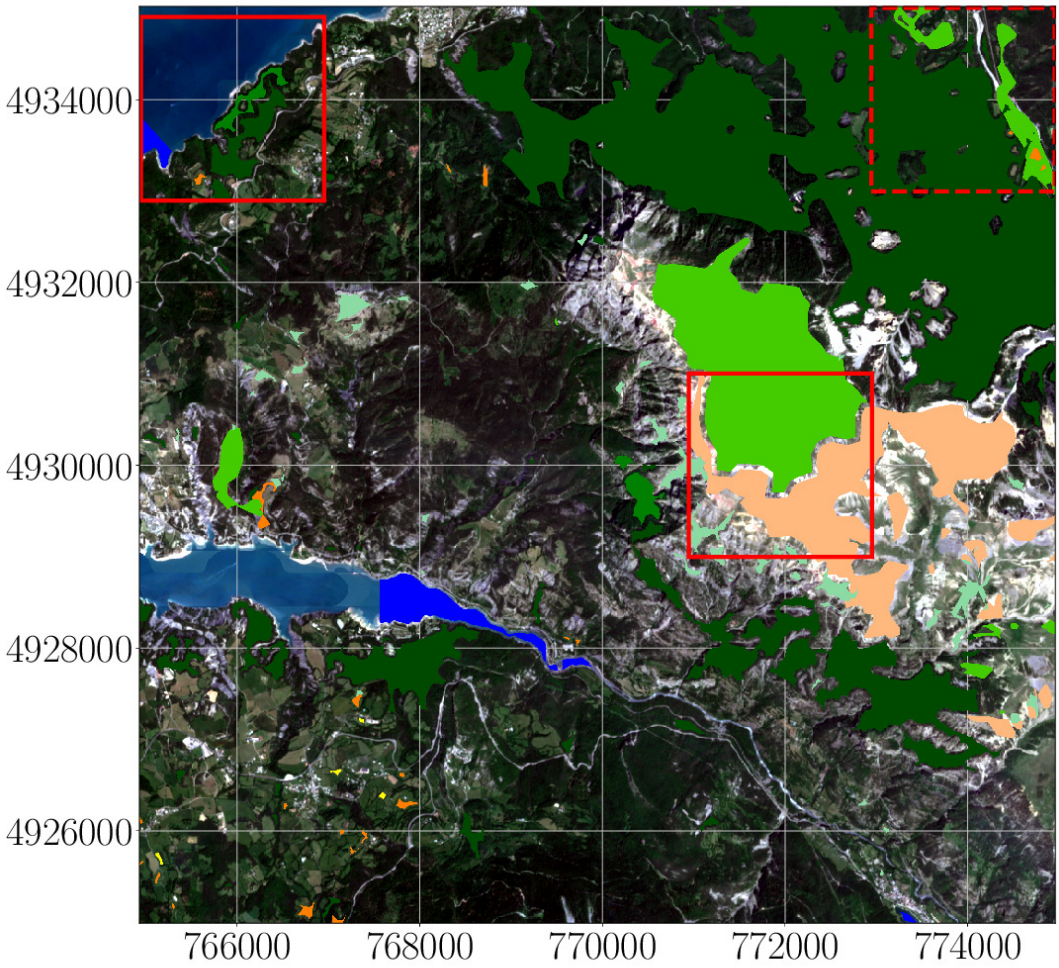


Fig. 5: Reference data extraction where each color represents an extract of land cover classes. The color code is described in Figure 8. The axes correspond to the geographical coordinates. The red continuous squares correspond to zoomed areas of the classification thematic maps provided in Figure 8 and the dashed square corresponds to the reconstruction map given in Figure 16. The background image is an extract from Sentinel-2 optical images.

TABLE II: F_1 scores for tile T31TCJ.

Class	QDA	RF	SVM	MIMGP
Continuous urban fabric	23.0 ± 11.8	53.6 ± 01.1	**56.9 ± 01.2**	27.8 ± 03.6
Discontinuous urban fabric	24.2 ± 05.6	**53.7 ± 02.7**	38.9 ± 10.1	51.2 ± 01.1
Industrial or commercial units	13.1 ± 03.7	**55.5 ± 00.9**	47.8 ± 03.8	27.7 ± 01.6
Road surfaces	36.7 ± 06.1	**85.4 ± 01.7**	78.1 ± 04.0	65.7 ± 03.2
Winter crops	40.7 ± 07.9	93.2 ± 00.7	**93.4 ± 01.1**	86.4 ± 02.0
Summer crops	66.5 ± 08.4	**96.6 ± 00.3**	95.5 ± 00.9	92.1 ± 01.0
Meadow	25.4 ± 07.7	**63.4 ± 02.9**	63.2 ± 03.7	53.7 ± 06.2
Orchards	48.8 ± 05.9	**79.2 ± 02.7**	76.4 ± 03.3	53.1 ± 04.4
Vines	21.6 ± 13.9	74.4 ± 06.6	**78.7 ± 07.0**	64.6 ± 09.7
Broad-leaved forest	58.6 ± 11.0	**85.3 ± 02.4**	84.6 ± 02.9	71.7 ± 03.6
Coniferous forest	28.4 ± 16.5	86.1 ± 02.5	**86.4 ± 02.4**	73.4 ± 03.6
Natural grasslands	06.0 ± 07.1	**29.3 ± 15.3**	22.5 ± 12.2	21.8 ± 13.0
Woody moorlands	25.9 ± 03.6	54.6 ± 03.5	**55.4 ± 04.0**	20.6 ± 05.9
Water bodies	87.7 ± 07.7	**99.3 ± 00.0**	99.2 ± 00.1	95.1 ± 01.9

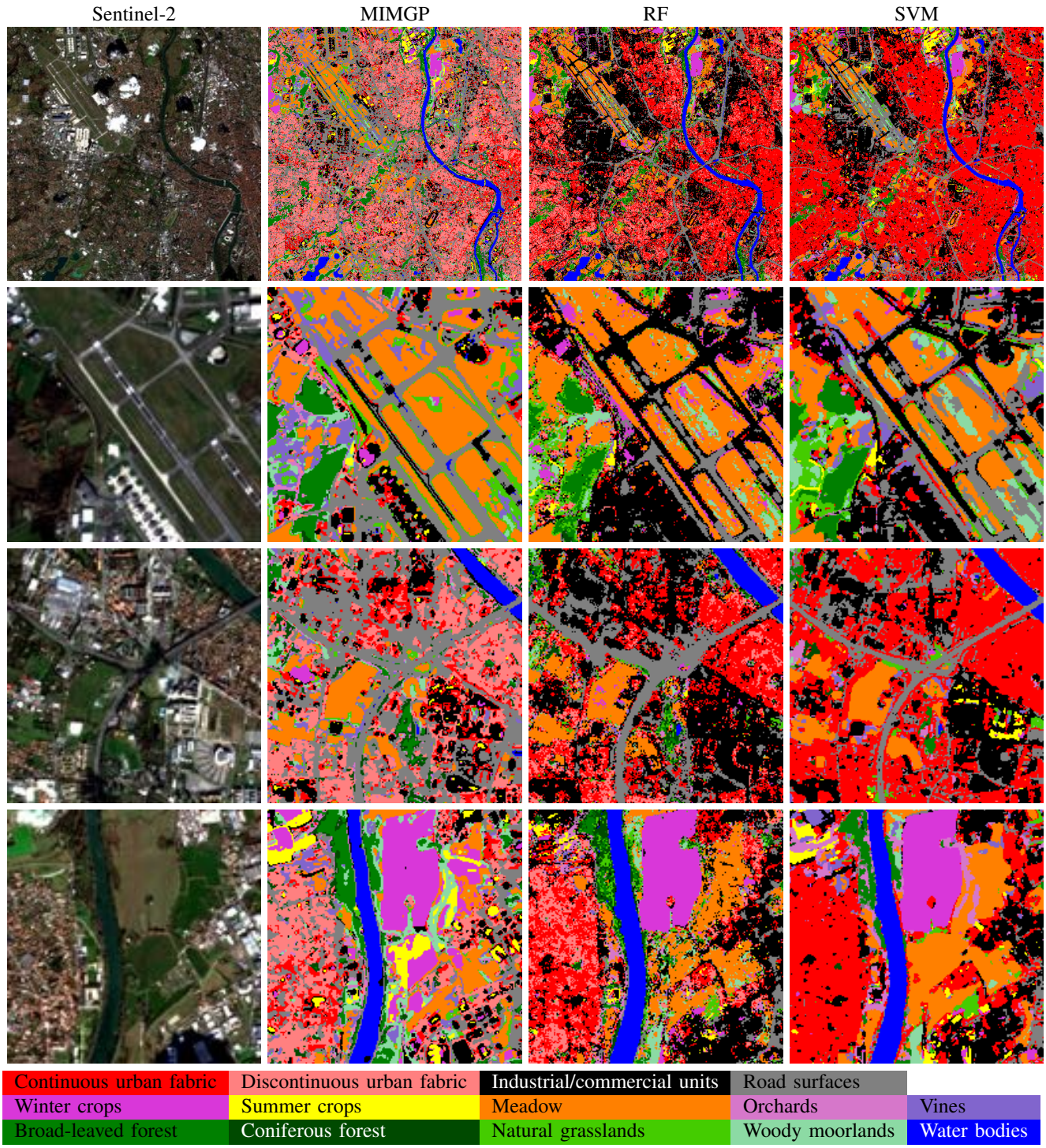


Fig. 6: Thematic maps for 3 sites from tile T31TCJ. The first line is the whole 10km long side square while the other lines are enlargements (2km length) of some extracts. The associated coordinates are presented in Figure 7 of the main document.

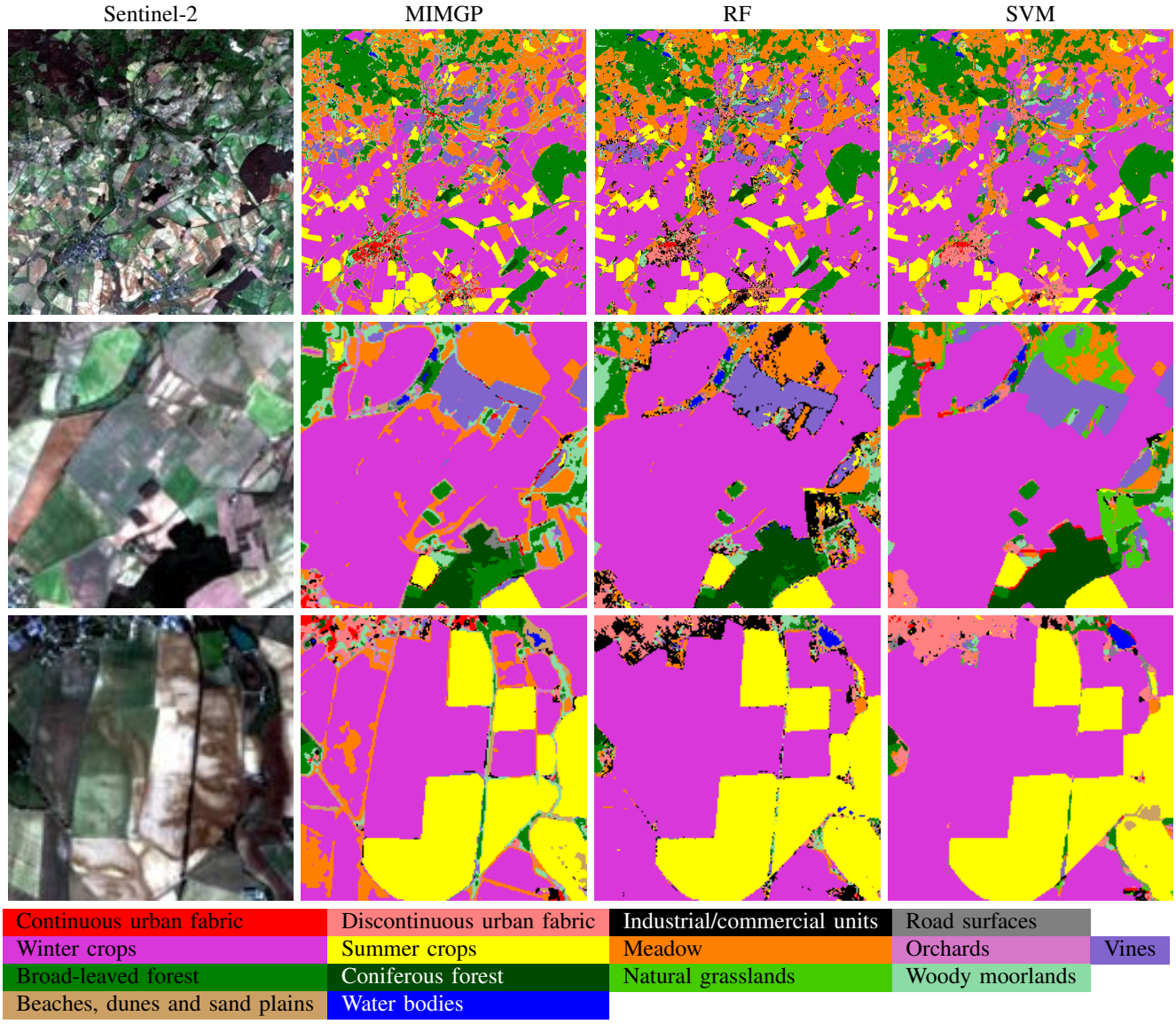


Fig. 7: Thematic maps for 3 sites from tile T31TDN. The first line is the whole 10km long side square while the other lines are enlargements (2km length) of some extracts. The associated coordinates are presented in Figure 4.

TABLE III: F_1 scores for tile T31TDN.

Class	QDA	RF	SVM	MIMGP
Continuous urban fabric	20.3 ± 7.7	**67.7 ± 2.3**	66.6 ± 4.0	58.5 ± 2.8
Discontinuous urban fabric	14.0 ± 6.5	**64.7 ± 3.6**	54.8 ± 11.0	51.3 ± 3.2
Industrial or commercial units	13.4 ± 2.2	**55.1 ± 2.6**	48.3 ± 4.3	38.3 ± 1.6
Road surfaces	34.9 ± 6.4	**83.3 ± 1.5**	75.6 ± 2.9	65.2 ± 2.6
Winter crops	23.0 ± 5.0	**93.4 ± 0.7**	91.4 ± 1.2	85.3 ± 1.6
Summer crops	26.5 ± 9.9	**95.5 ± 0.7**	94.2 ± 1.0	92.9 ± 0.8
Meadow	32.1 ± 3.0	85.0 ± 2.1	**85.4 ± 2.2**	71.2 ± 1.8
Orchards	24.4 ± 12.9	64.1 ± 9.6	**66.0 ± 20.1**	41.8 ± 6.8
Vines	42.8 ± 13.9	**89.6 ± 2.1**	88.4 ± 3.0	85.8 ± 2.0
Broad-leaved forest	7.6 ± 10.1	**86.0 ± 1.2**	83.5 ± 2.0	70.9 ± 2.8
Coniferous forest	61.6 ± 7.6	**91.9 ± 0.3**	91.2 ± 0.5	80.3 ± 1.5
Natural grasslands	24.3 ± 11.6	**43.1 ± 11.3**	36.7 ± 14.3	39.3 ± 10.0
Woody moorlands	4.9 ± 4.7	**69.2 ± 2.2**	68.3 ± 3.2	49.8 ± 2.5
Beaches, dunes and sand plains	43.1 ± 16.5	**80.6 ± 5.6**	78.7 ± 4.4	59.7 ± 6.8
Water bodies	84.1 ± 3.3	**94.0 ± 1.6**	92.3 ± 2.6	86.4 ± 2.6

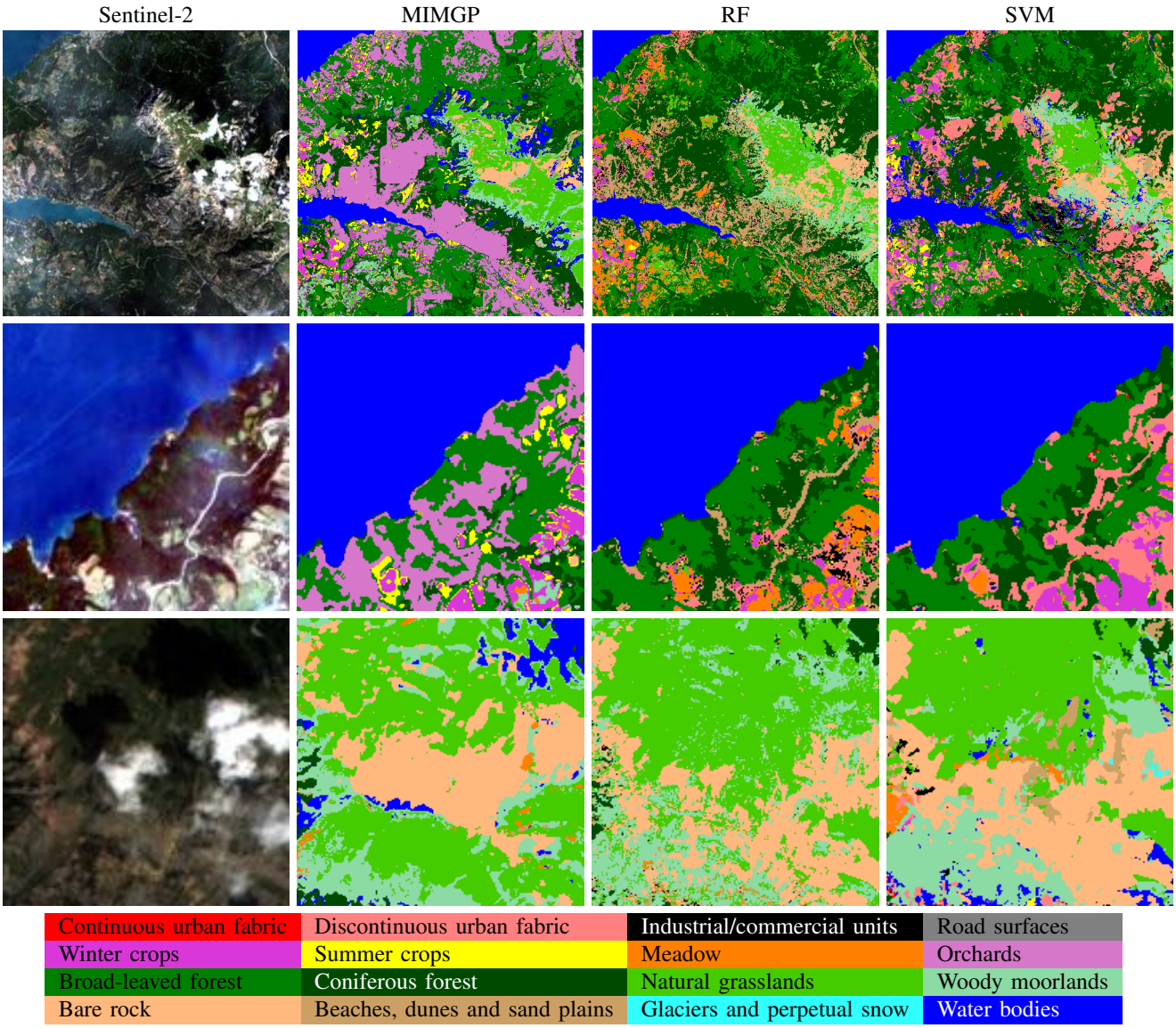


Fig. 8: Thematic maps for 3 sites from tile T31TGK. The first line is the whole 10km long side square while the other lines are enlargements (2km length) of some extracts. The associated coordinates are presented in Figure 5.

TABLE IV: F_1 scores for tile T31TGK.

Class	QDA	RF	SVM	MIMGP
Continuous urban fabric	0.3 ± 0.9	**9.1 ± 3.6**	1.7 ± 1.0	8.7 ± 4.2
Discontinuous urban fabric	23.2 ± 18.3	**54.6 ± 4.5**	45.6 ± 11.0	27.7 ± 15.5
Industrial or commercial units	27.0 ± 6.1	**55.5 ± 3.5**	50.3 ± 4.1	28.2 ± 4.7
Road surfaces4	33.3 ± 12.2	**63.2 ± 8.5**	61.3 ± 9.0	42.1 ± 5.8
Winter crops	61.6 ± 9.2	**91.3 ± 1.9**	88.0 ± 3.7	72.4 ± 3.2
Summer crops	80.8 ± 4.7	**94.3 ± 1.4**	92.8 ± 2.2	81.9 ± 3.2
Meadow	25.2 ± 4.7	48.1 ± 2.6	**53.1 ± 2.2**	19.2 ± 3.2
Orchards	84.6 ± 3.6	**91.5 ± 1.2**	88.9 ± 2.0	73.5 ± 2.3
Broad-leaved forest	20.9 ± 15.8	**79.9 ± 5.2**	78.5 ± 5.2	56.0 ± 17.0
Coniferous forest	43.8 ± 4.6	**79.4 ± 2.5**	79.2 ± 2.6	45.4 ± 4.6
Natural grasslands	35.2 ± 5.6	**49.3 ± 3.2**	45.5 ± 9.3	37.9 ± 4.4
Woody moorlands	46.7 ± 5.3	**50.8 ± 2.9**	48.4 ± 5.4	33.2 ± 2.3
Bare rock	40.7 ± 7.0	72.3 ± 4.9	**73.2 ± 4.4**	46.8 ± 4.2
Beaches, dunes and sand plains	47.3 ± 14.3	56.8 ± 9.3	**61.3 ± 10.7**	33.8 ± 7.5
Glaciers and perpetual snow	76.6 ± 10.2	89.3 ± 7.3	**90.9 ± 2.7**	78.7 ± 10.8
Water bodies	14.5 ± 11.1	**86.3 ± 2.4**	83.3 ± 4.1	43.2 ± 13.1

on the $y = x$ line, confirming that there is no bias in the reconstructions, but Whittaker exhibits higher dispersion. This is especially true for tiles T31TDN and T31TGK.

Finally, MIMGP and Whittaker image reconstructions are compared visually on the three tiles. The four wavelengths natively at a resolution of 10 meters per pixel (Blue, Green, Red and Infrared) are reported in Figures 14, 15 and 16 for, respectively, tile T31TCJ, T31TDN and T31TGK. For the visualization sake, each image of a given tile has been displayed using the same histogram stretching function and identical parameters. The mask is a binary image, with white color for valid pixels and black color for invalid pixels (clouds, saturation, ...).

REFERENCES

[1] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[2] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, "Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization," *ACM Transactions on Mathematical Software (TOMS)*, vol. 23, no. 4, pp. 550–560, 1997.

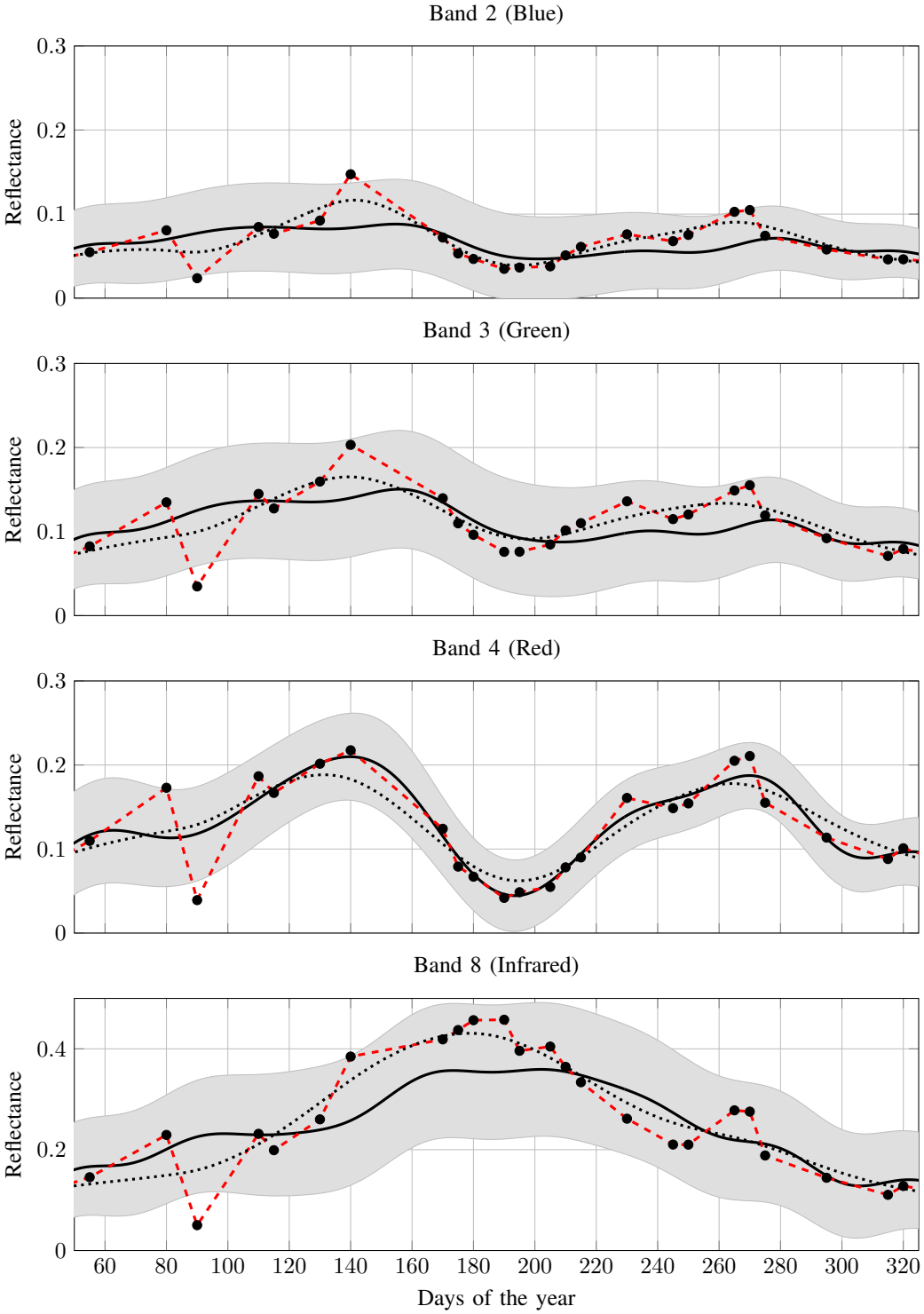


Fig. 9: Blue, Green, Red and IR wavelengths reconstruction for the Summer crops class. The red dashed line represents the linear interpolation, the black dotted line the Whittaker reconstruction and the black continuous line the MIMGP reconstruction.

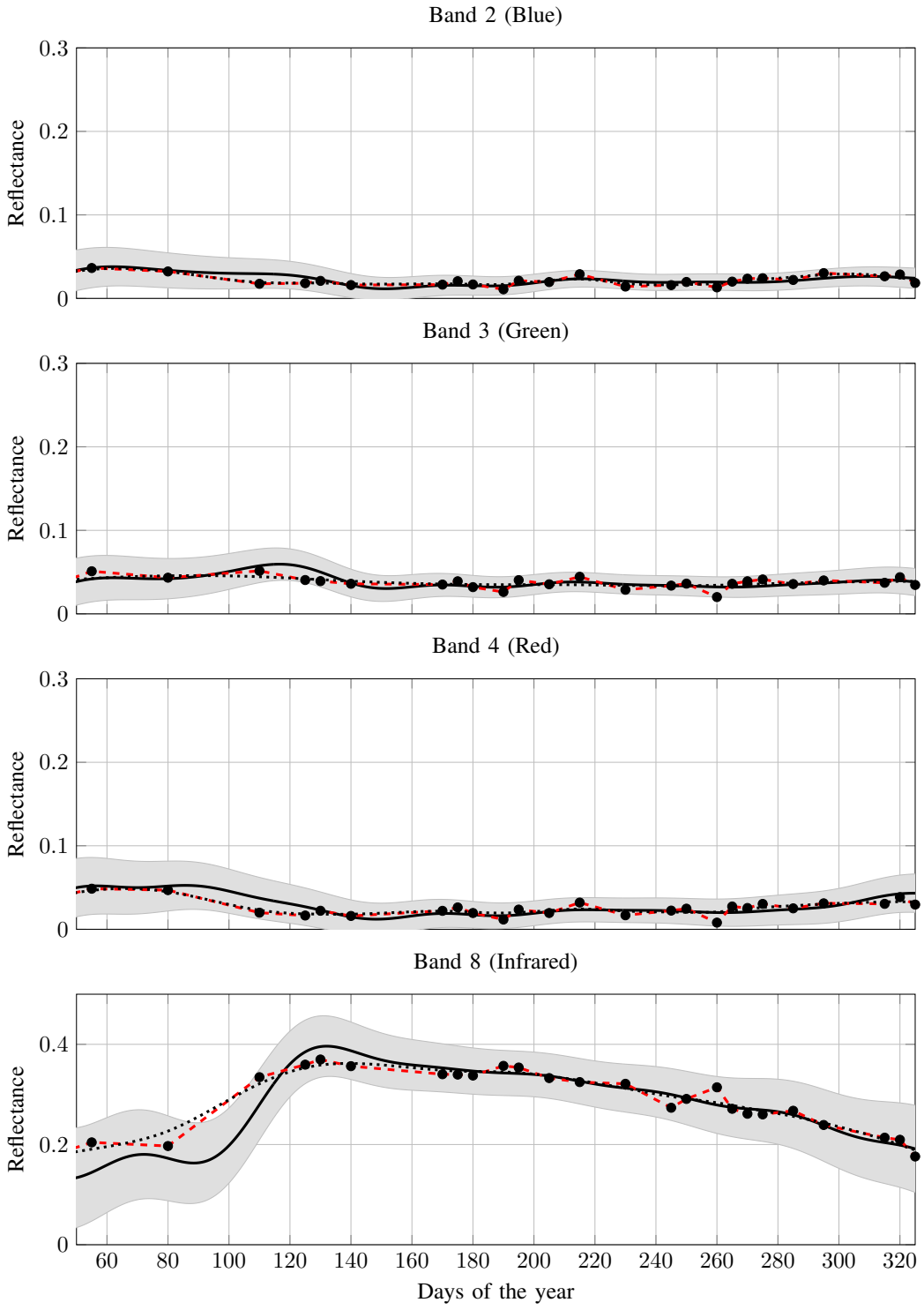


Fig. 10: Blue, Green, Red and IR wavelengths reconstruction for the broad-leaved forests class. The red dashed line represents the linear interpolation, the black dotted line the Whittaker reconstruction and the black continuous line the MIMGP reconstruction.

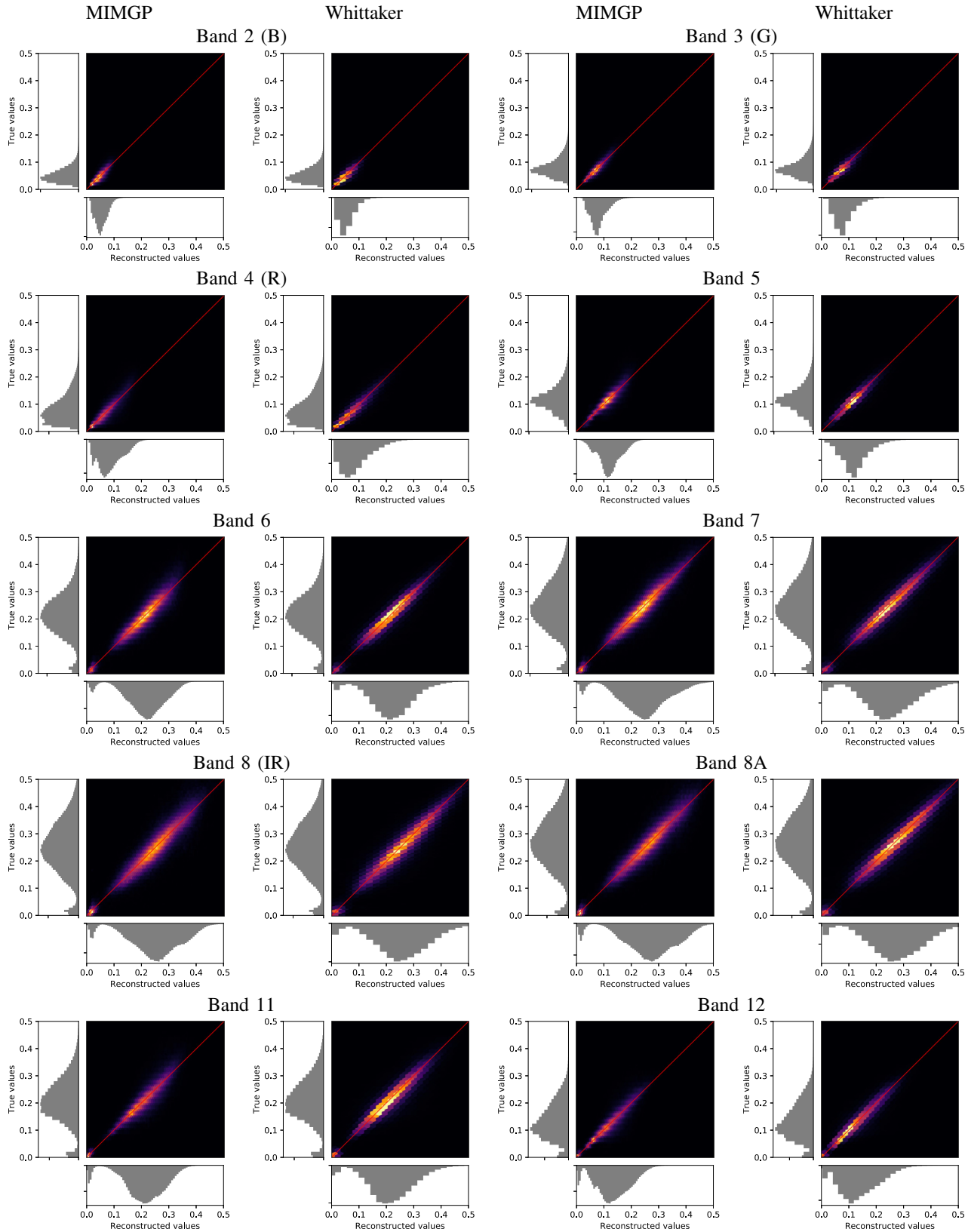


Fig. 11: Scatter plots from tile T31TCJ. They illustrate the link between the true values (x-axis) and the reconstructed ones (y-axis) with MIMGP (left) and with the Whittaker smoother (right).

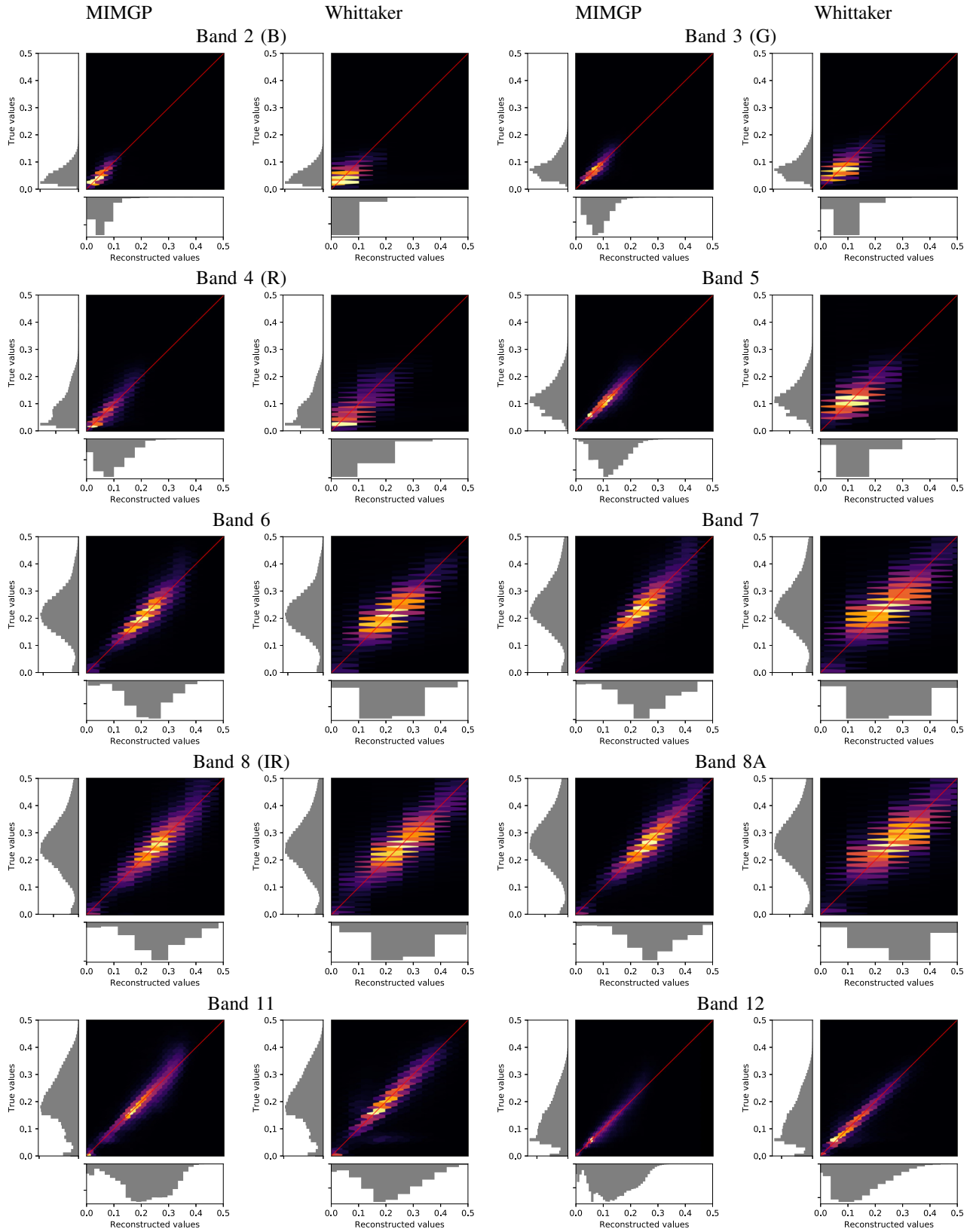


Fig. 12: Scatter plots from tile T31TDN. They illustrate the link between the true values (x-axis) and the reconstructed ones (y-axis) with MIMGP (left) and with the Whittaker smoother (right).

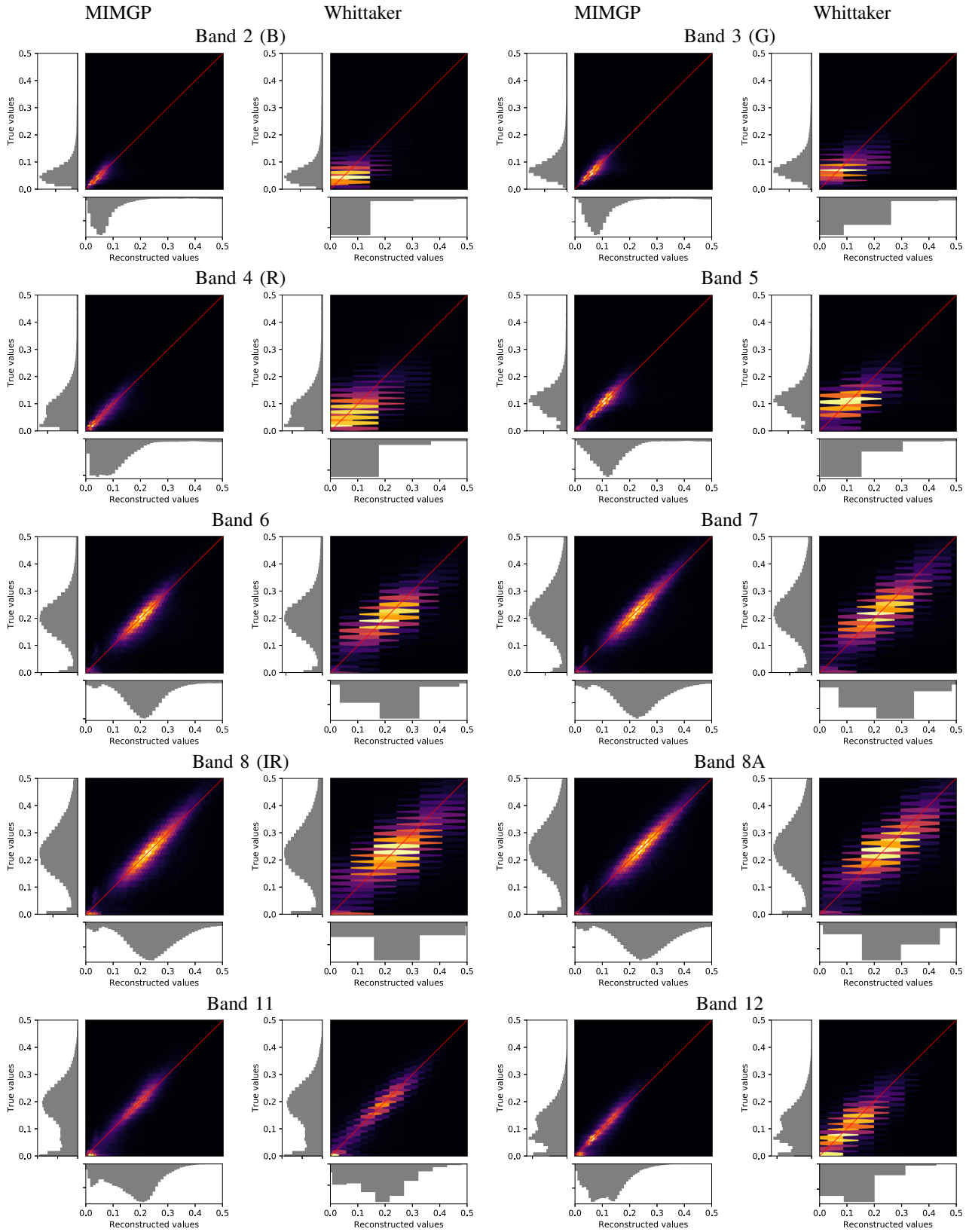


Fig. 13: Scatter plots from tile T31TGK. They illustrate the link between the true values (x-axis) and the reconstructed ones (y-axis) with MIMGP (left) and with the Whittaker smoother (right).

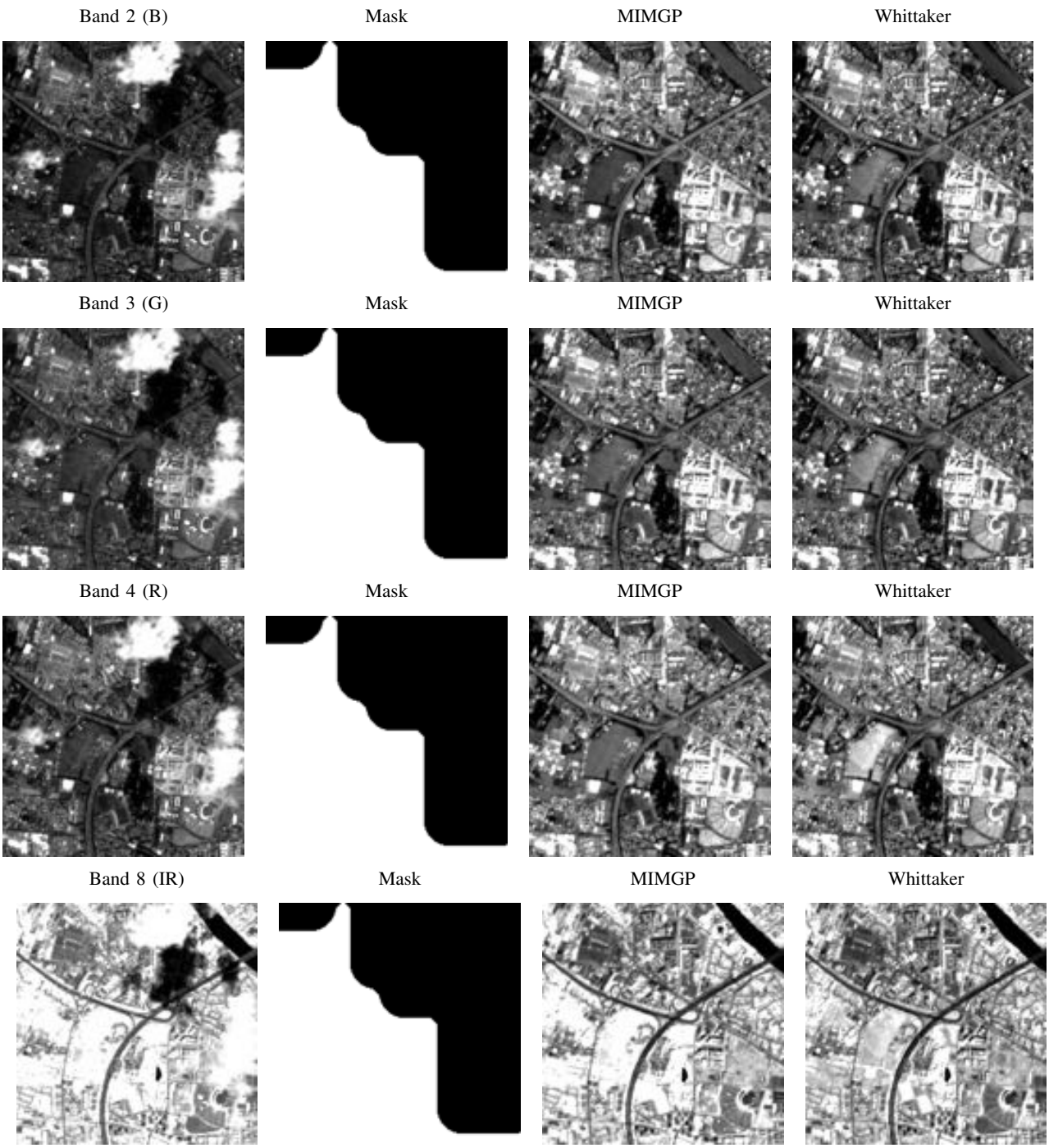


Fig. 14: Comparison of reconstructions on a 2km long side square from T31TCJ tile on July 12, 2018. The associated site is presented in Figure 7 of the main document.

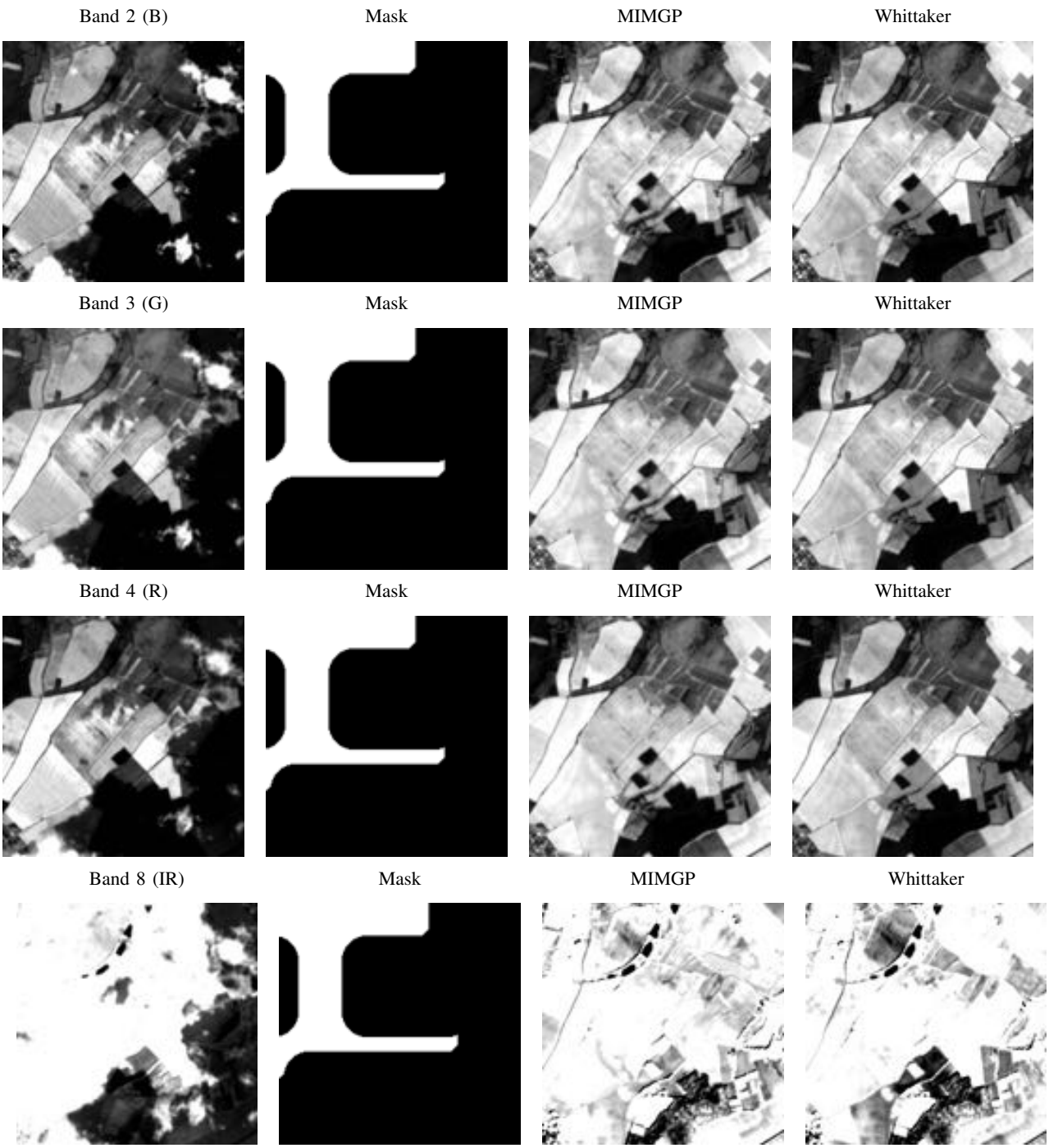


Fig. 15: Comparison of reconstructions on a 2km long side square from T31TDN tile on July 14, 2018. The associated site is presented in Figure 4.

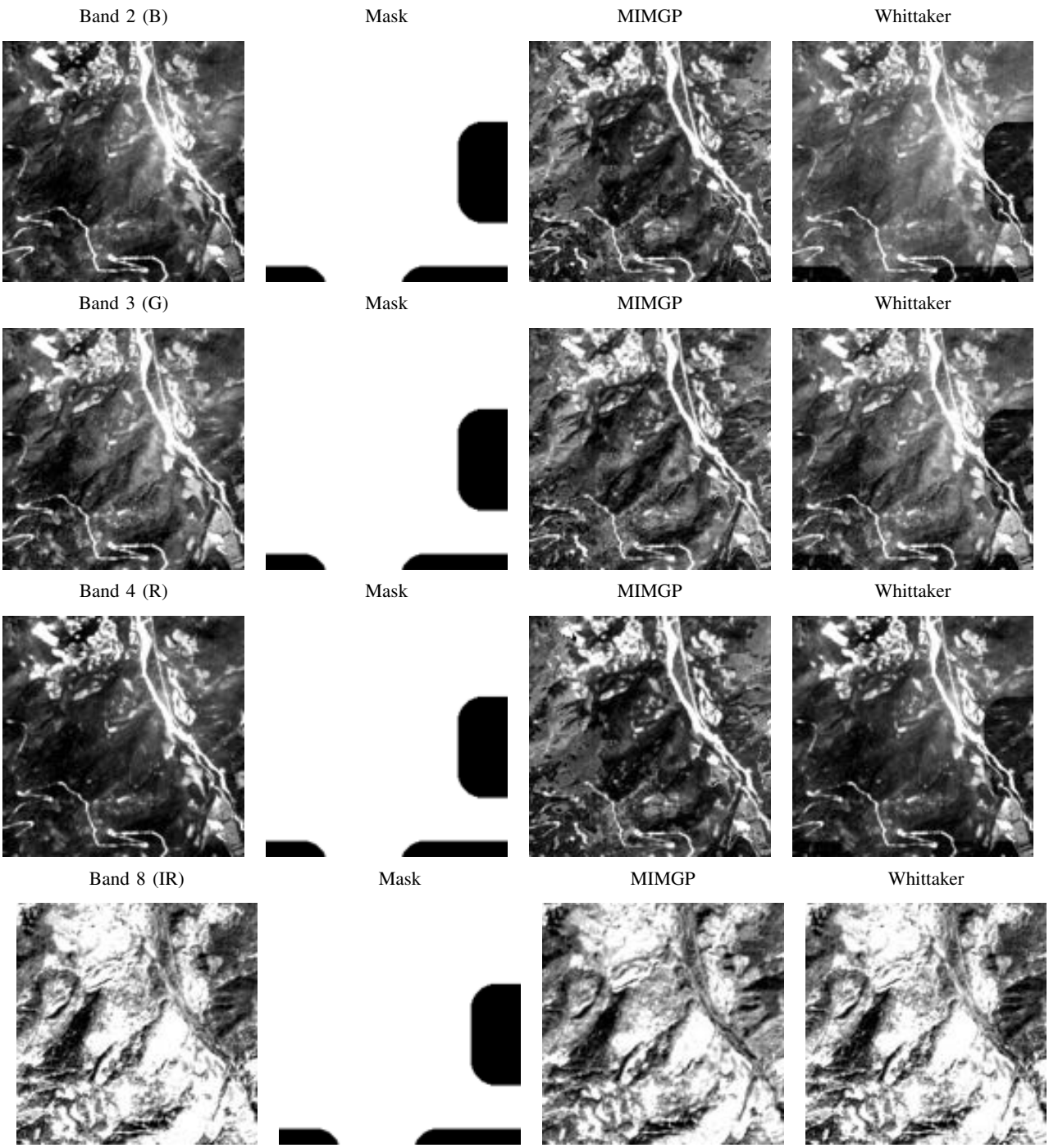


Fig. 16: Comparison of reconstructions on a 2km long side square from T31TGK tile on August 6, 2018. The associated site is presented in Figure 5.

APPENDIX C

COMPLEMENTS ABOUT THE CODE AND OPTIMIZATION

This appendix provides additional details about the code. Recall that the codes are written in python and available online, MIMGP from Chapter 4 is available at:
`https://gitlab.inria.fr/aconstan/supervised-classification-and-reconstruction-of-irregularly-sampled-times-series`,
and M2GP from Chapter 5 at:
`https://gitlab.inria.fr/aconstan/mixture-of-multivariate-gaussian-processes-for-classification-of-irregularly-sampled-satellite-image-time-series`.

Section C.1 presents additional details on implementation and identify key elements to optimize and Section C.2 presents the optimization done.

C.1 Implementation of the code

Firstly, the code has been designed to inherit scikit-learn API [137]. To do so, some classical functions are provided as "*score*" or "*fit*" and "*predict*". It allows the use of scikit-learn's functions as GridSearch.

The model is a python class. The use of the model is standard. However the main problem with python is the use of "*for*" loops. The language is well optimized in a vectorized approach of numerical objects. In out context two loops are identified as heavy from a computational point of view:

1. A loop on each sample for each class (and each spectral band for MIMGP) is hardly vectorized.
2. A loop on each sample when computing the **log-marginal likelihood** and the **prediction of posterior probabilities** functions.

Additionally the code has been implemented in the CNES computational resources in order to use both the HPC clusters and the simplified Sentinel-2 data requests.

C.2 Optimization of the code

An important part of this work is to optimize the code to compute the large number of Sentinel-2 SITS, the two optimizations are described briefly.

The first loop has been easily parallelized using multi-processing techniques. It consists of $p \times C$ independent loops for MIMGP and C independent loops for M2GP, where C is the number of classes and p the number of spectral bands.

The second loop does not allow parallelization. Indeed the loop computes the sum of the likelihood and each element of the sum is heavy to compute: computation of kernel operators, inversion of matrices, *etc*. The latter part also has to be computed to predict posterior probabilities. The latter one uses cython, a compiled language, to process the functions. Algorithms 4 and 5 in Section C.3 are an example of the cython code to compute $\hat{\boldsymbol{\alpha}}$. This code is then compiled using the cython compiler. Figure C.1 shows the computation time saved to compute the log-likelihood (about half to a third of the python computation time) for different numbers of samples and Figure C.2 shows the computation time to predict the probabilities (about 10 to 20 times faster). Both times are averaged on 10 runs - One run constructs n toy samples with an average of 10 time-stamps and computes each function once.

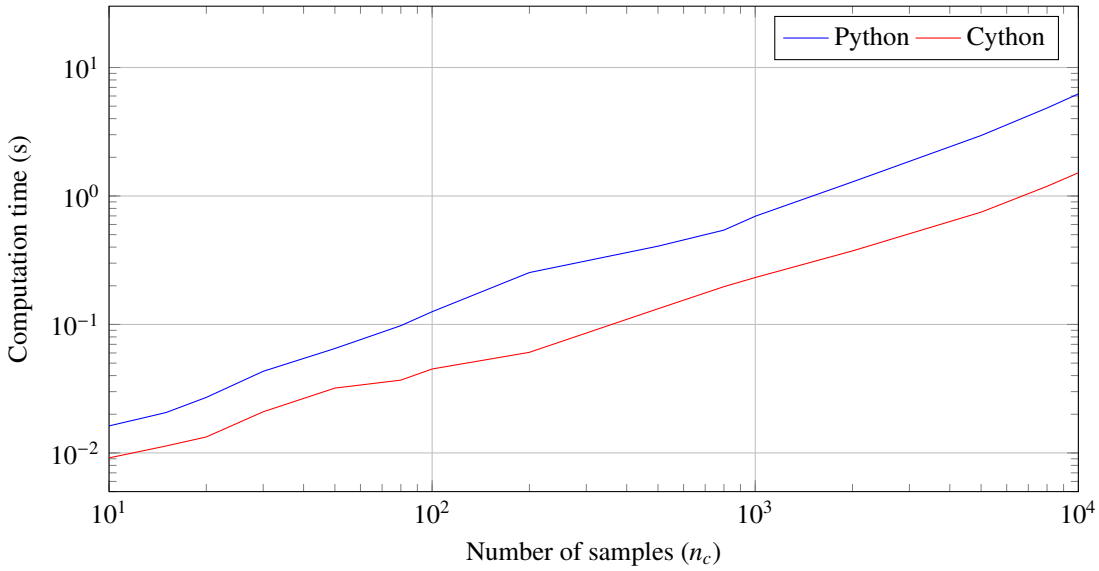


Figure C.1: Computation time of log-likelihood averaged on 10 runs.

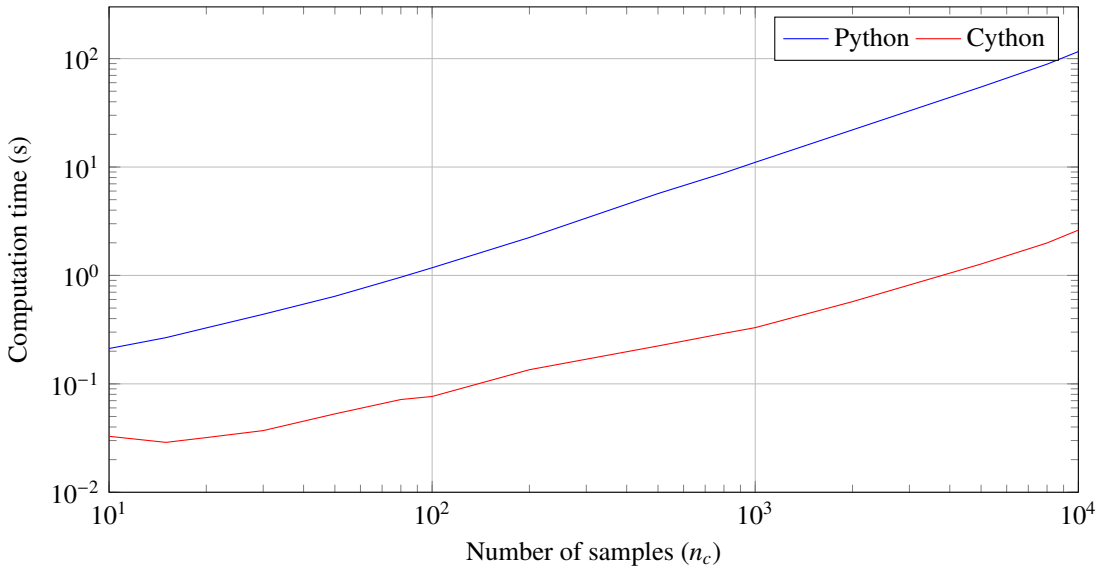


Figure C.2: Computation time of the computation of posterior probabilites on all classes averaged on 10 runs.

C.3 Cython code examples

Algorithm 4: Compute $\hat{\alpha}$ (1/2)

```
# Cython decorations to obtain time
@cython.boundscheck(False)
@cython.wraparound(False)
# Subroutine Compute alpha for a fixed band / class
cpdef double[::1] c_compute_alpha(const double[:, :] K,
                                  const double[:] T, const double[:, :] B,
                                  const double[:] X_c_flatten, const double[:] t_c_flatten,
                                  const int[:] t_indexes_flatten,
                                  const unsigned long[:] starts,
                                  int n_basis, double epsilon = 1e-12):
    cdef:
        int i, j, k, n_samples, ti
        int info, rank_, lwork, unit_ = 1
        int len_t = T.shape[0], nlvl = max(0, int(log(n_basis/26)/log(2) + 1))
        double rcond = -1., alpha_ = 1., neg_alpha_ = -1., beta_ = 1., zero_ = 0., wkopt
        double[:] t_c, X_c
        int[:] tmp_ind
        double[:, :] Ki, Li
        # Fortran order variables
        double[::1, :] Mi, Bi
        double[::1, :] G = np.zeros((n_basis, n_basis), order = 'F')
        double[::1, :] G_copy = np.zeros((n_basis, n_basis), order = 'F')
        double[::1] D = np.zeros((n_basis), order = 'F')
        double[::1] alpha = np.empty((n_basis), order = 'F')
        # To solve linear equations
        double[::1] s_ = np.empty((n_basis))
        double *work
        double *work_corr
        int *iwork

    # Initialization
    n_samples = starts.shape[0] - 1

    # Loop for each samples n_c:
    #    - Compute the matrix to be inverted (G).
    #    - Compute the right vector of the system (D).
    for i in range(n_samples):
        # Get the observed time samples for sample i.
        ti = starts[i+1] - starts[i]

        # Init views on the signal
        t_c = t_c_flatten[starts[i]:starts[i+1]].copy() # Time
        X_c = X_c_flatten[starts[i]:starts[i+1]].copy() # Reflectance
        tmp_ind = t_indexes_flatten[starts[i]:starts[i+1]].copy()
        # Init Matrices
        Mi = np.empty((ti, n_basis), order = 'F')
        Li = np.empty((ti, ti))
        # Extract information
        Bi = np.asfortranarray(B.base[tmp_ind, :]) # Designe matrix for sample i
        Ki = K.base[np.ix_(tmp_ind, tmp_ind)]      # Covariance operator for sample i

        # Solve numerical issues
        for j in range(ti):
            Ki[j, j] += epsilon

        # Solve linear problem
        chol_c(Ki, Li, ti)            # Cholesky decomposition of Ki saved within Li
        cho_solve_mat_c(Li, Bi, Mi, ti, n_basis) # Save Bi*inv(Ki) within Mi

        # G += Mi.T * Bi  (Bi.T * inv(Ki) * Bi)
        dgemm('T', 'N', &n_basis, &n_basis, &ti, &alpha_,
              &Mi[0,0], &ti, &Bi[0,0], &ti, &beta_, &G[0,0], &n_basis)
```

Algorithm 5: Compute $\hat{\alpha}$ (2/2)

```
66          # D += Mi.T * yi  (Bi.T * inv(Ki) * yi)
            dgemm('T', 'N', &n_basis, &unit_, &ti, &alpha_,
68                &Mi[0,0], &ti, &X_c[0], &ti, &beta_, &D[0], &n_basis)

70      # Compute alpha: Solve the system G alpha = D
        alpha[...] = D; G_copy[...] = G
72      # Call solve_c function written below
        solve_c(G_copy, D, alpha, n_basis, &info)
74      # If inverse of G cannot be computed numerically, info = n_basis + 1,
        #   then compute the best solution using least square solution.
76      if info > 0:
            alpha[...] = D; G_copy[...] = G
78          # Initialize parameters - query the optimal workspace
            lwork = -1  # Compute nothing but return computation space within wkopt
80          iwork = <int *> PyMem_Malloc((3*n_basis*nlvl + 11*n_basis) * sizeof(int))
            dgelsd(&n_basis, &n_basis, &unit_, &G_copy[0,0], &n_basis, &alpha[0], &n_basis,
82                 &s_[0], &rcond, &rank_, &wkopt, &lwork, iwork, &info)
            # Get parameters and compute least square solution ||G * alpha - D||^2
84          lwork = int(wkopt)
            work = <double *> PyMem_Malloc(lwork * sizeof(double))
86          dgelsd(&n_basis, &n_basis, &unit_, &G_copy[0,0], &n_basis, &alpha[0], &n_basis,
                   &s_[0], &rcond, &rank_, work, &lwork, iwork, &info)
88          # Free memory
            PyMem_Free(work); PyMem_Free(iwork)
90
        # Return alpha
92      return alpha

94


96
    # C function (written in cython) to solve, wrt X, the system: AX = B
98  cdef void solve_c(double[::1, :] A, double[::1] B, double[::1] X,
                      int n_basis_, int* info) nogil:
100     cdef :
            char equed = b'N'
102         int unit_ = 1, i
            double rcond_s
104         double ferr, berr
            int *ipiv_ = <int *> malloc(n_basis_ * sizeof(unsigned long))
106         int *iwork_ = <int *> malloc(n_basis_ * sizeof(unsigned long))
            double *work_ = <double*> malloc(4 * n_basis_ * sizeof(double))
108         double *r_ = <double*> malloc(n_basis_ * sizeof(double))
            double *c_ = <double*> malloc(n_basis_ * sizeof(double))
110         double *Af = <double *> malloc(n_basis_ * n_basis_ * sizeof(double))

112     X[...] = B

114     # Call fortran function
        dgesvx('E', 'N', &n_basis_, &unit_, &A[0,0], &n_basis_, Af, &n_basis_,
116            &ipiv_[0], &equed, r_, c_, &B[0], &n_basis_, &X[0], &n_basis_, &rcond_s,
               &ferr, &berr, work_, &iwork_[0], info)
118
        # Free memory
120     free(<void*> ipiv_); free(<void*> iwork_)
        free(<void*> r_); free(<void*> c_)
122     free(<void*> work_); free(<void*> Af)
```

REFERENCES

[1] G. I. Allen and R. Tibshirani, "Transposable regularized covariance models with an application to missing data imputation", *The Annals of Applied Statistics*, vol. 4, no. 2, pp. 764–790, Jun. 2010. DOI: 10.1214/09-AOAS314.

[2] M. A. Àlvarez and N. D. Lawrence, "Computationally Efficient Convolved Multiple Output Gaussian Processes", *Journal of Machine Learning Research*, vol. 12, no. 41, pp. 1459–1500, 2011, ISSN: 1533-7928.

[3] M. A. Àlvarez, L. Rosasco, and N. D. Lawrence, "Kernels for vector-valued functions: A review", *Foundations and Trends® in Machine Learning*, vol. 4, no. 3, pp. 195–266, Jun. 2012, ISSN: 1935-8237, 1935-8245. DOI: 10.1561/2200000036.

[4] J. L. Andrews and P. D. McNicholas, "Model-based clustering, classification, and discriminant analysis via mixtures of multivariate t-distributions", *Statistics and Computing*, vol. 22, no. 5, pp. 1021–1029, Sep. 2012.

[5] N. Audebert, B. Le Saux, and S. Lefevre, "Deep Learning for Classification of Hyperspectral Data: A Comparative Review", *IEEE Geoscience and Remote Sensing Magazine*, vol. 7, no. 2, pp. 159–173, Jun. 2019. DOI: 10.1109/MGRS.2019.2912563.

[6] A. Azzalini, "A class of distributions which includes the normal ones", *Scandinavian Journal of Statistics*, vol. 12, no. 2, pp. 171–178, 1985.

[7] A. Azzalini and A. D. Valle, "The Multivariate Skew-Normal Distribution", *Biometrika*, vol. 83, no. 4, pp. 715–726, 1996.

[8] L. Baetens, C. Desjardins, and O. Hagolle, "Validation of Copernicus Sentinel-2 Cloud Masks Obtained from MAJA, Sen2Cor, and FMask processors using reference cloud masks generated with a supervised active learning procedure", *Remote Sensing*, vol. 11, no. 4, 2019. DOI: 10.3390/rs11040433.

[9] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh, "The great time series classification bake off: A review and experimental evaluation of recent algorithmic advances", *Data Mining and Knowledge Discovery*, vol. 31, no. 3, pp. 606–660, May 2017.

[10] E. Batur and D. Maktav, "Assessment of Surface Water Quality by Using Satellite Images Fusion Based on PCA Method in the Lake Gala, Turkey", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 5, pp. 2983–2989, May 2019. DOI: 10.1109/tgrs.2018.2879024.

[11] P. Beck, C. Atzberger, K. Hogda, B. Johansen, and A. Skidmore, "Improved monitoring of vegetation dynamics at very high latitudes: A new method using MODIS NDVI", *Remote Sensing of Environment 100 (2006) 3*, vol. 100(3), Jan. 2006.

[12] P. Benedetti, D. Ienco, R. Gaetano, K. Ose, R. G. Pensa, and S. Dupuy, "M^3Fusion: A Deep Learning Architecture for Multiscale Multimodal Multitemporal Satellite Data Fusion", *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 12, pp. 4939–4949, Dec. 2018. DOI: 10.1109/jstars.2018.2876357.

[13] M. Berger, J. Moreno, J. A. Johannessen, P. F. Levelt, and R. F. Hanssen, "ESA's sentinel missions in support of Earth system science", *Remote Sensing of Environment*, The Sentinel Missions - New Opportunities for Science, vol. 120, pp. 84–90, May 2012. DOI: 10.1016/j.rse.2011.07.023.

[14] M. Bertolacci, E. Cripps, O. Rosen, J. W. Lau, and S. Cripps, "Climate inference on daily rainfall across the Australian continent, 1876-2015", *The Annals of Applied Statistics*, vol. 13, no. 2, pp. 683–712, Jun. 2019. DOI: 10.1214/18-AOAS1218.

[15] C. Biernacki and J. Jacques, "Model-based clustering of multivariate ordinal data relying on a stochastic binary search algorithm", *Statistics and Computing*, vol. 26, no. 5, pp. 929–943, Sep. 2016, ISSN: 1573-1375. DOI: 10.1007/s11222-015-9585-2.

[16] I. Bilionis and N. Zabaras, "Multi-output local Gaussian process regression: Applications to uncertainty quantification", *Journal of Computational Physics*, vol. 231, no. 17, pp. 5718–5746, Jul. 2012. DOI: 10.1016/j.jcp.2012.04.047.

[17] M. Bilodeau and D. Brenner, *Theory of multivariate statistics*. Springer Science & Business Media, 2008.

[18] E. V. Bonilla, K. Chai, and C. Williams, "Multi-task Gaussian Process Prediction", *Advances in Neural Information Processing Systems*, vol. 20, pp. 153–160, 2007.

[19] M. S. Boori, K. Choudhary, R. Paringer, A. K. Sharma, A. Kupriyanov, and S. Corgne, "Monitoring crop phenology using ndvi time series from sentinel 2 satellite data", *2019 5th International Conference on Frontiers of Signal Processing (ICFSP)*, Sep. 2019. DOI: 10.1109/icfsp48124.2019.8938078.

[20] M. Bossard, J. Feranec, J. Otahel, *et al.*, *CORINE land cover technical guide: Addendum 2000*. European Environment Agency Copenhagen, 2000, vol. 40.

[21] N. Bouguila, D. Ziou, and J. Vaillancourt, "Novel Mixtures Based on the Dirichlet Distribution: Application to Data and Image Classification", in *Machine Learning and Data Mining in Pattern Recognition*, P. Perner and A. Rosenfeld, Eds., ser. Lecture Notes in Computer Science, Berlin, Heidelberg: Springer, 2003, pp. 172–181.

[22] C. Bouveyron, G. Celeux, T. B. Murphy, and A. E. Raftery, *Model-Based Clustering and Classification for Data Science: With Applications in R*. Cambridge University Press, Jun. 2019, vol. 50. DOI: 10.1017/9781108644181.

[23] C. Bouveyron, M. Fauvel, and S. Girard, "Kernel discriminant analysis and clustering with parsimonious gaussian process models", *Statistics and Computing*, vol. 25, no. 6, pp. 1143–1162, Nov. 2015. DOI: 10.1007/s11222-014-9505-x.

[24] C. Bouveyron, S. Girard, and C. Schmid, "High-Dimensional Discriminant Analysis", *Communications in Statistics - Theory and Methods*, vol. 36, no. 14, pp. 2607–2623, Oct. 2007. DOI: 10.1080/03610920701271095.

[25] P. Boyle and M. Frean, "Dependent Gaussian Processes", in *Advances in Neural Information Processing Systems*, vol. 17, MIT Press, 2005.

[26] L. Breiman, "Random Forests", *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001. DOI: 10.1023/A:1010933404324.

[27] ——, "Statistical Modeling: The Two Cultures", *Statistical Science*, vol. 16, no. 3, pp. 199–231, Aug. 2001, Publisher: Institute of Mathematical Statistics, ISSN: 0883-4237, 2168-8745. DOI: 10.1214/ss/1009213726.

[28] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising", *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005. DOI: 10.1109/cvpr.2005.38.

[29] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, "API design for machine learning software: Experiences from the scikit-learn project", in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 2013, pp. 108–122.

[30] R. Calandra, J. Peters, C. E. Rasmussen, and M. P. Deisenroth, "Manifold Gaussian Processes for regression", in *2016 International Joint Conference on Neural Networks (IJCNN)*, Vancouver, BC, Canada: IEEE, Jul. 2016, pp. 3338–3345, ISBN: 978-1-5090-0620-5. DOI: 10.1109/IJCNN.2016.7727626.

[31] G. Camps-Valls, J. Verrelst, J. Munoz-Mari, V. Laparra, F. Mateo-Jimenez, and J. Gomez-Dans, "A Survey on Gaussian Processes for Earth-Observation Data Analysis: A Comprehensive Investigation", *IEEE Geoscience and Remote Sensing Magazine*, vol. 4, no. 2, pp. 58–78, Jun. 2016. DOI: 10.1109/mgrs.2015.2510084.

[32] G. Camps-Valls, L. Gomez-Chova, J. Munoz-Mari, J. L. Rojo-Alvarez, and M. Martinez-Ramon, "Kernel-Based Framework for Multitemporal and Multisource Remote Sensing Data Classification and Change Detection", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 6, pp. 1822–1835, Jun. 2008. DOI: 10.1109/TGRS.2008.916201.

[33] H. D. V. Cardona, M. A. Àlvarez, and À. A. Orozco, "Convolved Multi-output Gaussian Processes for Semi-Supervised Learning", in *Image Analysis and Processing — ICIAP 2015*, V. Murino and E. Puppo, Eds., ser. Lecture Notes in Computer Science, Springer International Publishing, 2015, pp. 109–118. DOI: 10.1007/978-3-319-23231-7_10.

[34] G. Celeux and G. Govaert, "Clustering criteria for discrete data and latent class models", *Journal of Classification*, vol. 8, no. 2, pp. 157–176, Dec. 1991.

[35] F. Chamroukhi, "Skew t mixture of experts", *Neurocomputing*, vol. 266, pp. 390–408, Nov. 2017.

[36] J. Chen, J. Chen, A. Liao, X. Cao, L. Chen, X. Chen, C. He, G. Han, S. Peng, M. Lu, W. Zhang, X. Tong, and J. Mills, "Global land cover mapping at 30m resolution: A POK-based operational approach", *ISPRS Journal of Photogrammetry and Remote Sensing*, Global Land Cover Mapping and Monitoring, vol. 103, pp. 7–27, May 2015. DOI: 10.1016/j.isprsjprs.2014.09.002.

[37] Z. Chen, B. Wang, and A. N. Gorban, "Multivariate Gaussian and Student-t process regression for multi-output prediction", *Neural Computing and Applications*, vol. 32, no. 8, pp. 3005–3028, Apr. 2020. DOI: 10.1007/s00521-019-04687-8.

[38] L.-F. Cheng, B. Dumitrascu, G. Darnell, C. Chivers, M. Draugelis, K. Li, and B. E. Engelhardt, "Sparse multi-output Gaussian processes for online medical time series prediction", *BMC Medical Informatics and Decision Making*, vol. 20, no. 1, p. 152, Jul. 2020. DOI: 10.1186/s12911-020-1069-4.

[39] M. Chi, A. Plaza, J. A. Benediktsson, Z. Sun, J. Shen, and Y. Zhu, "Big Data for Remote Sensing: Challenges and Opportunities", *Proceedings of the IEEE*, vol. 104, no. 11, pp. 2207–2219, Nov. 2016. DOI: 10.1109/jproc.2016.2598228.

[40] A. D. Chouakria and P. N. Nagabhushan, "Adaptive dissimilarity index for measuring time series proximity", *Advances in Data Analysis and Classification*, vol. 1, no. 1, pp. 5–21, Mar. 2007. DOI: 10.1007/s11634-006-0004-6.

[41] A. Constantin, M. Fauvel, and S. Girard, "Joint supervised classification and reconstruction of irregularly sampled satellite image times series", *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–13, 2021, To appear. DOI: 10.1109/TGRS.2021.3076667.

[42] ——, "Mixture of multivariate gaussian processes for classification of irregularly sampled satellite image time-series", working paper or preprint, 2021.

[43] C. Cortes and V. Vapnik, "Support-vector networks", *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995. DOI: 10.1007/BF00994018.

[44] N. A. C. Cressie, *Statistics for Spatial Data*. New-York: John Wiley & Sons, Ltd, 1993, ISBN: 978-1-119-11515-1.

[45] N. Cressie and C. K. Wikle, *Statistics For Spatio-Temporal Data*. John Wiley & Sons, 2015, ISBN: 978-0-471-69274-4.

[46] R. Cresson, D. Ienco, R. Gaetano, K. Ose, and D. H. Tong Minh, "Optical image gap filling using deep convolutional autoencoder from optical and radar images", IEEE, Jul. 2019. DOI: 10.1109/igarss.2019.8900353.

[47] A. Damianou and N. D. Lawrence, "Deep Gaussian Processes", in *Artificial Intelligence and Statistics*, ISSN: 1938-7228, PMLR, Apr. 2013, pp. 207–215.

[48] A. P. Dawid, "Some matrix-variate distribution theory: Notational considerations and a Bayesian application", *Biometrika*, vol. 68, no. 1, pp. 265–274, Apr. 1981, ISSN: 0006-3444. DOI: 10.1093/biomet/68.1.265.

[49] D. Derksen, J. Inglada, and J. Michel, "Geometry Aware Evaluation of Handcrafted Superpixel-Based Features and Convolutional Neural Networks for Land Cover Mapping Using Satellite Imagery", *Remote Sensing*, vol. 12, no. 3, 2020. DOI: 10.3390/rs12030513.

[50] A. Dezfouli and E. V. Bonilla, "Scalable Inference for Gaussian Process Models with Black-Box Likelihoods", in *Advances in Neural Information Processing Systems*, vol. 28, Curran Associates, Inc., 2015.

[51] C. Donlon, B. Berruti, A. Buongiorno, M. .-.-H. Ferreira, P. Féménias, J. Frerick, P. Goryl, U. Klein, H. Laur, C. Mavrocordatos, J. Nieke, H. Rebhan, B. Seitz, J. Stroede, and R. Sciarra, "The Global Monitoring for Environment and Security (GMES) Sentinel-3 mission", *Remote Sensing of Environment*, The Sentinel Missions - New Opportunities for Science, vol. 120, pp. 37–57, May 2012. DOI: 10.1016/j.rse.2011.07.024.

[52] M. Drusch, U. Del Bello, S. Carlier, O. Colin, V. Fernandez, F. Gascon, B. Hoersch, C. Isola, P. Laberinti, P. Martimort, A. Meygret, F. Spoto, O. Sy, F. Marchese, and P. Bargellini, "Sentinel-2: ESA's Optical High-Resolution Mission for GMES Operational Services", *Remote Sensing of Environment*, The Sentinel Missions - New Opportunities for Science, vol. 120, pp. 25–36, May 2012. DOI: 10.1016/j.rse.2011.11.026.

[53] P. Dutilleul, "The MLE algorithm for the matrix normal distribution", *Journal of Statistical Computation and Simulation*, vol. 64, no. 2, pp. 105–123, Mar. 1999. DOI: 10.1080/00949659908811970.

[54] P. H. C. Eilers, "A Perfect Smoother", *Analytical Chemistry*, vol. 75, no. 14, pp. 3631–3636, Jul. 2003. DOI: 10.1021/ac034173t.

[55] J. Elith and J. R. Leathwick, "Species distribution models: Ecological explanation and prediction across space and time", *Annual Review of Ecology, Evolution, and Systematics*, vol. 40, no. 1, pp. 677–697, 2009. DOI: 10.1146/annurev.ecolsys.110308.120159.

[56] M. Fauvel, C. Bouveyron, and S. Girard, "Parsimonious Gaussian Process Models for the Classification of Hyperspectral Remote Sensing Images", *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 12, pp. 2423–2427, Dec. 2015. DOI: 10.1109/LGRS.2015.2481321.

[57] M. Fauvel, M. Lopes, T. Dubo, J. Rivers-Moore, P.-L. Frison, N. Gross, and A. Ouin, "Prediction of plant diversity in grasslands using Sentinel-1 and -2 satellite image time series", *Remote Sensing of Environment*, vol. 237, p. 111 536, Feb. 2020. DOI: 10.1016/j.rse.2019.111536.

[58] S. Feng, J. Zhao, T. Liu, H. Zhang, Z. Zhang, and X. Guo, "Crop Type Identification and Mapping Using Machine Learning Algorithms and Sentinel-2 Time Series Data", *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 9, pp. 3295–3306, Sep. 2019. DOI: 10.1109/jstars.2019.2922469.

[59] R. Fernandez-Beltran, J. M. Haut, M. E. Paoletti, J. Plaza, A. Plaza, and F. Pla, "Multimodal Probabilistic Latent Semantic Analysis for Sentinel-1 and Sentinel-2 Image Fusion", *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 9, pp. 1347–1351, Sep. 2018. DOI: 10.1109/lgrs.2018.2843886.

[60] F. Ferraty and P. Vieu, *Nonparametric Functional Data Analysis: Theory and Practice*, ser. Springer Series in Statistics. New York: Springer-Verlag, 2006, ISBN: 978-0-387-30369-7.

[61] S. Flaxman, M. Chirico, P. Pereira, and C. Loeffler, "Scalable high-resolution forecasting of sparse spatiotemporal events with kernel methods: A winning solution to the NIJ "Real-Time Crime Forecasting Challenge"", *The Annals of Applied Statistics*, vol. 13, no. 4, pp. 2564–2585, Dec. 2019, ISSN: 1932-6157, 1941-7330. DOI: 10.1214/19-AOAS1284.

[62] G. Frasso and P. H. C. Eilers, "L- and v-curves for optimal smoothing", *Statistical Modelling*, vol. 15, pp. 91–111, Feb. 2015. DOI: 10.1177/1471082X14549288.

[63] J. H. Friedman, "Regularized Discriminant Analysis", *Journal of the American Statistical Association*, vol. 84, no. 405, pp. 165–175, Mar. 1989, ISSN: 0162-1459. DOI: 10.1080/01621459.1989.10478752.

[64] P. J. García-Laencina, J.-L. Sancho-Gómez, and A. R. Figueiras-Vidal, "Pattern classification with missing data: A review", *Neural Computing and Applications*, vol. 19, no. 2, pp. 263–282, Mar. 2010.

[65] V. S. F. Garnot, L. Landrieu, S. Giordano, and N. Chehata, "Satellite image time series classification with pixel-set encoders and temporal self-attention", in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020.

[66] Y. J. E. Gbodjo, D. Ienco, and L. Leroux, "Toward Spatio–Spectral Analysis of Sentinel-2 Time Series Data for Land Cover Mapping", *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 2, pp. 307–311, Feb. 2020. DOI: 10.1109/lgrs.2019.2917788.

[67] M. G. Genton, "Classes of kernels for machine learning: A statistics perspective", *Journal of machine learning research*, vol. 2, pp. 299–312, Dec. 2001.

[68] U. B. Gewali, S. T. Monteiro, and E. Saber, "Gaussian Processes for Vegetation Parameter Estimation from Hyperspectral Data with Limited Ground Truth", *Remote Sensing*, vol. 11, no. 13, p. 1614, Jul. 2019, ISSN: 2072-4292. DOI: 10.3390/rs11131614. [Online]. Available: http://dx.doi.org/10.3390/rs11131614.

[69] C. Giri and J. Long, "Land Cover Characterization and Mapping of South America for the Year 2010 Using Landsat 30 m Satellite Data", *Remote Sensing*, vol. 6, no. 10, pp. 9494–9510, Oct. 2014. DOI: 10.3390/rs6109494.

[70] A. Gittens and M. W. Mahoney, "Revisiting the nyström method for improved large-scale machine learning", *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 3977–4041, Jan. 2016, ISSN: 1532-4435.

[71] H. Glanz and L. Carvalho, "An expectation-maximization algorithm for the matrix normal distribution with an application in remote sensing", *Journal of Multivariate Analysis*, vol. 167, pp. 31–48, Sep. 2018. DOI: 10.1016/j.jmva.2018.03.010.

[72] L. Gómez-Chova, J. Munoz-Marí, V. Laparra, J. Malo-López, and G. Camps-Valls, "A Review of Kernel Methods in Remote Sensing Data Analysis", in *Optical Remote Sensing: Advances in Signal Processing and Exploitation Techniques*, ser. Augmented Vision and Reality, S. Prasad, L. M. Bruce, and J. Chanussot, Eds., Berlin, Heidelberg: Springer, 2011, pp. 171–206, ISBN: 978-3-642-14212-3. DOI: 10.1007/978-3-642-14212-3_10.

[73] P. Gong, J. Wang, L. Yu, Y. Zhao, Y. Zhao, L. Liang, Z. Niu, X. Huang, H. Fu, S. Liu, C. Li, X. Li, W. Fu, C. Liu, Y. Xu, X. Wang, Q. Cheng, L. Hu, W. Yao, H. Zhang, P. Zhu, Z. Zhao, H. Zhang, Y. Zheng, L. Ji, Y. Zhang, H. Chen, A. Yan, J. Guo, L. Yu, L. Wang, X. Liu, T. Shi, M. Zhu, Y. Chen, G. Yang, P. Tang, B. Xu, C. Giri, N. Clinton, Z. Zhu, J. Chen, and J. Chen, "Finer resolution observation and monitoring of global land cover: First mapping results with Landsat TM and ETM+ data", *International Journal of Remote Sensing*, vol. 34, no. 7, pp. 2607–2654, Apr. 2013. DOI: 10.1080/01431161.2012.748992.

[74] P. Gonzalez-Navarro, M. Moghadamfalahi, M. Akcakaya, and D. Erdogmus, "Spatio-Temporal EEG Models for Brain Interfaces", *Signal processing*, vol. 131, pp. 333–343, Feb. 2017. DOI: 10.1016/j.sigpro.2016.08.001.

[75] P. Goovaerts, *Geostatistics for Natural Resources Evaluation*. Oxford University Press, USA, 1997, Publisher: Cambridge University Press, ISBN: 0-19-511538-4.

[76] K. Greenewald and A. O. Hero, "Robust kronecker product PCA for spatio-temporal covariance estimation", *IEEE Transactions on Signal Processing*, vol. 63, no. 23, pp. 6368–6378, 2015. DOI: 10.1109/TSP.2015.2472364.

[77] A. K. Gupta and D. K. Nagar, *Matrix Variate Distributions*. Chapman and Hall/CRC, Oct. 1999. DOI: 10.1201/9780203749289.

[78] J. Haas and Y. Ban, "Urban Land Cover and Ecosystem Service Changes based on Sentinel-2A MSI and Landsat TM Data", *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 2, pp. 485–497, Feb. 2018. DOI: 10.1109/jstars.2017.2786468.

[79] O. Hagolle, M. Huc, D. V. Pascual, and G. Dedieu, "A multi-temporal method for cloud detection, applied to FORMOSAT-2, VENμS, LANDSAT and SENTINEL-2 images", *Remote Sensing of Environment*, vol. 114, no. 8, pp. 1747–1755, Aug. 2010. DOI: 10.1016/j.rse.2010.03.002.

[80] O. Hagolle. (2014). The product level names, how they work ?, [Online]. Available: https://labo.obs-mip.fr/multitemp/the-product-names-how-they-work/ (visited on 08/16/2021).

[81] ——, (2015). MACCS/MAJA, how it works, [Online]. Available: https://labo.obs-mip.fr/multitemp/maccs-how-it-works/ (visited on 08/12/2021).

[82] ——, (2015). The sentinel-2 tiles, how they work ?, [Online]. Available: https://labo.obs-mip.fr/multitemp/the-sentinel-2-tiles-how-they-work/ (visited on 08/12/2021).

[83] ——, (2016). Radiometric quantities : Irradiance, radiance, reflectance, [Online]. Available: https://labo.obs-mip.fr/multitemp/radiometric-quantities-irradiance-radiance-reflectance/ (visited on 08/16/2021).

[84] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY: Springer, 2009.

[85] K. Hayashi, M. Imaizumi, and Y. Yoshida, "On Random Subsampling of Gaussian Process Regression: A Graphon-Based Analysis", in *International Conference on Artificial Intelligence and Statistics*, ISSN: 2640-3498, PMLR, Jun. 2020, pp. 2055–2065.

[86] J. Hensman, A. Matthews, and Z. Ghahramani, "Scalable Variational Gaussian Process Classification", in *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, G. Lebanon and S. V. N. Vishwanathan, Eds., ser. Proceedings of Machine Learning Research, vol. 38, San Diego, California, USA: PMLR, 2015, pp. 351–360.

[87] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory", *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. DOI: 10.1162/neco.1997.9.8.1735.

[88] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning", *The Annals of Statistics*, vol. 36, no. 3, pp. 1171–1220, Jun. 2008.

[89] C. Homer, J. Dewitz, S. Jin, G. Xian, C. Costello, P. Danielson, L. Gass, M. Funk, J. Wickham, S. Stehman, R. Auch, and K. Riitters, "Conterminous United States land cover change patterns 2001–2016 from the 2016 National Land Cover Database", *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 162, pp. 184–199, Apr. 2020. DOI: 10.1016/j.isprsjprs.2020.02.019.

[90] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines", *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, Mar. 2002. DOI: 10.1109/72.991427.

[91] C. Huang, L. S. Davis, and J. R. G. Townshend, "An assessment of support vector machines for land cover classification", *International Journal of Remote Sensing*, vol. 23, no. 4, pp. 725–749, Jan. 2002. DOI: 10.1080/01431160110040323.

[92] G. C. Iannelli and P. Gamba, "Urban Extent Extraction Combining Sentinel Data in the Optical and Microwave Range", *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 7, pp. 2209–2216, Jul. 2019. DOI: 10.1109/jstars.2019.2920678.

[93] J. Inglada, A. Vincent, M. Arias, B. Tardy, D. Morin, and I. Rodes, "Operational High Resolution Land Cover Map Production at the Country Scale Using Satellite Image Time Series", *Remote Sensing*, vol. 9, no. 1, 2017. DOI: 10.3390/rs9010095.

[94] M. Ingram, D. Vukcevic, and N. Golding, "Multi-output Gaussian processes for species distribution modelling", *Methods in Ecology and Evolution*, vol. 11, no. 12, pp. 1587–1598, 2020. DOI: 10.1111/2041-210X.13496.

[95] P. Jonsson and L. Eklundh, "Seasonality extraction by function fitting to time-series of satellite sensor data", *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 40, pp. 1824–1832, Sep. 2002. DOI: 10.1109/TGRS.2002.802519.

[96] K. Karra, C. Kontgis, Z. Statman-Weil, J. C. Mazzariello, M. Mark, and S. P. Brumby, "Global land use / land cover with sentinel 2 and deep learning", in *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, 2021, pp. 4704–4707. DOI: 10.1109/IGARSS47720.2021.9553499.

[97] P. Kempeneers and P. Soille, "Optimizing Sentinel-2 Image Selection in a big data context", *Big Earth Data*, vol. 1, no. 1-2, pp. 145–158, 2017. DOI: 10.1080/20964471.2017.1407489.

[98] N. Keshava and J. Mustard, "Spectral unmixing", *IEEE Signal Processing Magazine*, vol. 19, no. 1, pp. 44–57, Jan. 2002. DOI: 10.1109/79.974727.

[99] J. H. Kim, H. Lee, S. J. Hong, S. Kim, J. Park, J. Y. Hwang, and J. P. Choi, "Objects Segmentation From High-Resolution Aerial Images Using u-Net With Pyramid Pooling Layers", *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 1, pp. 115–119, Jan. 2019. DOI: 10.1109/LGRS.2018.2868880.

[100] O. Koyejo, C. Lee, and J. Ghosh, "Constrained Gaussian Process Regression for Gene-Disease Association", in *2013 IEEE 13th International Conference on Data Mining Workshops*, Dec. 2013, pp. 72–79. DOI: 10.1109/ICDMW.2013.150.

[101] N. M. Kriege, F. D. Johansson, and C. Morris, "A survey on graph kernels", *Applied Network Science*, vol. 5, no. 1, pp. 1–42, Dec. 2020.

[102] A. Lagrange, M. Fauvel, and M. Grizonnet, "Large-Scale Feature Selection With Gaussian Mixture Models for the Classification of High Dimensional Remote Sensing Images", *IEEE Transactions on Computational Imaging*, vol. 3, no. 2, pp. 230–242, Jun. 2017. DOI: 10.1109/TCI.2017.2666551.

[103] D. A. Landgrebe, *Signal Theory Methods in Multispectral Remote Sensing*. Newark, NJ: Wiley, 2005.

[104] R. M. Lark and A. Papritz, "Fitting a linear model of coregionalization for soil properties using simulated annealing", *Geoderma*, vol. 115, no. 3, pp. 245–260, Aug. 2003. DOI: 10.1016/S0016-7061(03)00065-X.

[105] Q. V. Le, A. J. Smola, and S. Canu, "Heteroscedastic Gaussian process regression", in *ICML 2005: Proceedings of the 22nd international conference on Machine learning*, New York, NY, USA, 2005, pp. 489–496. DOI: 10.1145/1102351.1102413.

[106] C. Li, H. Wulf, B. Schmid, J.-S. He, and M. E. Schaepman, "Estimating Plant Traits of Alpine Grasslands on the Qinghai-Tibetan Plateau Using Remote Sensing", *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 7, pp. 2263–2275, Jul. 2018. DOI: 10.1109/jstars.2018.2824901.

[107] H. Li, P. Xiao, X. Feng, Y. Yang, L. Wang, W. Zhang, X. Wang, W. Feng, and X. Chang, "Using Land Long-Term Data Records to Map Land Cover Changes in China Over 1981–2010", *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 4, pp. 1372–1389, Apr. 2017. DOI: 10.1109/JSTARS.2016.2645203.

[108] P. Li and S. Chen, "Hierarchical Gaussian Processes model for multi-task learning", *Pattern Recognition*, vol. 74, pp. 134–144, Feb. 2018. DOI: 10.1016/j.patcog.2017.09.021.

[109] W.-C. Lin and C.-F. Tsai, "Missing value imputation: A review and analysis of the literature (2006-2017)", *Artificial Intelligence Review*, vol. 53, no. 2, pp. 1487–1509, Feb. 2020.

[110] H. Liu, J. Cai, and Y.-S. Ong, "Remarks on multi-output Gaussian process regression", *Knowledge-Based Systems*, vol. 144, pp. 102–121, Mar. 2018. DOI: 10.1016/j.knosys.2017.12.034.

[111] H. Liu, Y.-S. Ong, X. Shen, and J. Cai, "When Gaussian process meets big data: A review of scalable GPs", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 11, pp. 4405–4423, Nov. 2020, ISSN: 2162-2388.

[112] X. Liu, V. Gopal, and J. Kalagnanam, "A spatio-temporal modeling framework for weather radar image data in tropical Southeast Asia", *The Annals of Applied Statistics*, vol. 12, no. 1, pp. 378–407, Mar. 2018. DOI: 10.1214/17-AOAS1064.

[113] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins, "Text classification using string kernels", *The Journal of Machine Learning Research*, vol. 2, pp. 419–444, Mar. 2002.

[114] M. Lopes, M. Fauvel, S. Girard, and D. Sheeren, "Object-based classification of grasslands from high resolution satellite image time series using gaussian mean map kernels", *Remote Sensing*, vol. 9, no. 7, 2017. DOI: 10.3390/rs9070688.

[115] M. Lopes, M. Fauvel, A. Ouin, and S. Girard, "Spectro-temporal heterogeneity measures from dense high spatial resolution satellite image time series: Application to grassland species diversity estimation", *Remote Sensing*, vol. 9, no. 10, 2017. DOI: 10.3390/rs9100993.

[116] N. Lu and D. L. Zimmerman, "The likelihood ratio test for a separable covariance matrix", *Statistics & Probability Letters*, vol. 73, no. 4, pp. 449–457, Jul. 2005. DOI: 10.1016/j.spl.2005.04.020.

[117] J. Luo, K. Ying, and J. Bai, "Savitzky–Golay smoothing and differentiation filter for even number data", *Signal Processing*, vol. 85, no. 7, pp. 1429–1434, Jul. 2005. DOI: 10.1016/j.sigpro.2005.02.002.

[118] D. J. MacKay, "Introduction to gaussian processes", *NATO ASI series F computer and systems sciences*, vol. 168, pp. 133–166, 1998.

[119] J. R. Magnus, "On the concept of matrix derivative", *Journal of Multivariate Analysis*, vol. 101, no. 9, pp. 2200–2206, Oct. 2010. DOI: 10.1016/j.jmva.2010.05.005.

[120] M. S. Mahanta, A. S. Aghaei, and K. N. Plataniotis, "Regularized LDA based on separable scatter matrices for classification of spatio-spectral EEG patterns", in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 1237–1241. DOI: 10.1109/ICASSP.2013.6637848.

[121] A. M. Manceur and P. Dutilleul, "Maximum likelihood estimation for the tensor normal distribution: Algorithm, minimum sample size, and empirical bias and dispersion", *Journal of Computational and Applied Mathematics*, vol. 239, pp. 37–49, Feb. 2013, ISSN: 0377-0427. DOI: 10.1016/j.cam.2012.09.017.

[122] D. G. Manolakis, R. B. Lockwood, and T. W. Cooley, *Hyperspectral Imaging Remote Sensing: Physics, Sensors, and Algorithms*. Cambridge University Press, 2016. DOI: 10.1017/CBO9781316017876.

[123] K. V. Mardia and C. R. Goodall, "Spatial-temporal analysis of multivariate environmental monitoring data", in *Multivariate Environmental Statistics*, 76, vol. 6, North-Holland, New York: Elsevier, 1993, pp. 347–385.

[124] A. Mathur and G. M. Foody, "Crop classification by support vector machine with intelligently selected training data for an operational application", *International Journal of Remote Sensing*, vol. 29, no. 8, pp. 2227–2240, Apr. 2008. DOI: 10.1080/01431160701395203.

[125] E. Maugeais, F. Lecordix, X. Halbecq, and A. Braun, "Dérivation cartographique multi échelles de la BDTopo de l'IGN france: Mise en œuvre du processus de production de la nouvelle carte de base", in French, in *Proceedings of the 25th Int. Cartographic Conference*, Paris, France, 2011, pp. 3–8.

[126] J. Mercer, "Functions of positive and negative type, and their connection the theory of integral equations", *Philosophical Transactions of the Royal Society A*, vol. 209, no. 441-458, pp. 415–446, Jan. 1909. DOI: 10.1098/rsta.1909.0016.

[127] T. P. Minka, "A family of algorithms for approximate Bayesian inference", PhD Thesis, Massachusetts Institute of Technology, 2001.

[128] G. Misra, F. Cawkwell, and A. Wingler, "Status of Phenological Research Using Sentinel-2 Data: A Review", *Remote Sensing*, vol. 12, no. 17, p. 2760, Jan. 2020. DOI: 10.3390/rs12172760.

[129] A. Moeini Rad, D. Ashourloo, H. Salehi Shahrabi, and H. Nematollahi, "Developing an Automatic Phenology-Based Algorithm for Rice Detection Using Sentinel-2 Time-Series Data", *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 5, pp. 1471–1481, May 2019, ISSN: 2151-1535. DOI: 10.1109/jstars.2019.2906684.

[130] P. Morales-Alvarez, A. Perez-Suay, R. Molina, and G. Camps-Valls, "Remote sensing image classification With Large-scale Gaussian Processes", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 2, pp. 1103–1114, Feb. 2018. DOI: 10.1109/tgrs.2017.2758922.

[131] P. Moreno-Muñoz, A. Artés, and M. Àlvarez, "Heterogeneous Multi-output Gaussian Process Prediction", in *Advances in Neural Information Processing Systems*, vol. 31, Curran Associates, Inc., 2018.

[132] P. M. Murray, R. P. Browne, and P. D. McNicholas, "A mixture of SDB skew-t factor analyzers", *Econometrics and Statistics*, vol. 3, no. C, pp. 160–168, 2017, Publisher: Elsevier.

[133] H. Nickisch and C. E. Rasmussen, "Approximations for binary gaussian process classification", *Journal of Machine Learning Research*, vol. 9, no. 67, pp. 2035–2078, 2008.

[134] A. O'Hagan, "Curve fitting and optimal design for prediction", *Journal of the Royal Statistical Society. Series B*, vol. 40, no. 1, pp. 1–42, 1978, ISSN: 0035-9246.

[135] C. Paris, J. Bioucas-Dias, and L. Bruzzone, "A Novel Sharpening Approach for Superresolving Multiresolution optical images", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 3, pp. 1545–1560, Mar. 2019. DOI: 10.1109/tgrs.2018.2867284.

[136] M. Pastorino, A. Montaldo, L. Fronda, I. Hedhli, G. Moser, S. B. Serpico, and J. Zerubia, "Multisensor and Multiresolution Remote Sensing Image Classification through a Causal Hierarchical Markov Framework and Decision Tree Ensembles", *Remote Sensing*, vol. 13, no. 5, p. 849, Jan. 2021. DOI: 10.3390/rs13050849.

[137] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in python", *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[138] C. Pelletier, G. I. Webb, and F. Petitjean, "Temporal Convolutional Neural Network for the Classification of Satellite Image Time Series", *Remote Sensing*, vol. 11, no. 5, p. 523, Jan. 2019. DOI: 10.3390/rs11050523.

[139] A. Pensoneault, X. Yang, and X. Zhu, "Nonnegativity-enforced Gaussian process regression", *Theoretical and Applied Mechanics Letters*, vol. 10, no. 3, pp. 182–187, Mar. 2020. DOI: 10.1016/j.taml.2020.01.036.

[140] M. Pereira-Sandoval, A. Ruiz-Verdu, C. Tenjo, J. Delegido, P. Urrego, R. Pena, E. Vicente, J. Soria, J. Soria, and J. Moreno, "Calibration and validation of algorithms for the estimation of chlorophyll-a in inland waters with sentinel-2", in *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, Jul. 2018. DOI: 10.1109/igarss.2018.8517371.

[141] L. Pipia, J. Muñoz-Marí, E. Amin, S. Belda, G. Camps-Valls, and J. Verrelst, "Fusing optical and SAR time series for LAI gap filling with multioutput Gaussian processes", *Remote Sensing of Environment*, vol. 235, p. 111 452, Dec. 2019. DOI: 10.1016/j.rse.2019.111452.

[142] R. J. Povinelli, M. T. Johnson, A. C. Lindgren, and Jinjin Ye, "Time series classification using gaussian mixture models of reconstructed phase spaces", *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 6, pp. 779–783, Jun. 2004. DOI: 10.1109/TKDE.2004.17.

[143] R. Pu and S. Landry, "Mapping urban tree species by integrating multi-seasonal high resolution pléiades satellite imagery with airborne LiDAR data", *Urban Forestry & Urban Greening*, vol. 53, p. 126 675, Aug. 2020. DOI: 10.1016/j.ufug.2020.126675.

[144] C. Qiu, L. Mou, M. Schmitt, and X. X. Zhu, "Local climate zone-based urban land cover classification from multi-seasonal Sentinel-2 images with a recurrent residual network", *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 154, pp. 151–162, Aug. 2019. DOI: 10.1016/j.isprsjprs.2019.05.004.

[145] J. Quiñonero-Candela and C. E. Rasmussen, "A unifying view of sparse approximate gaussian process regression", *Journal of Machine Learning Research*, vol. 6, no. Dec, pp. 1939–1959, 2005.

[146] J. Ramsay and B. Silverman, *Functional Data Analysis*, ser. Springer Series in Statistics. Springer, 2005, ISBN: 9780387400808.

[147] S. Remes, M. Heinonen, and S. Kaski, "Non-Stationary Spectral Kernels", *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[148] C. Revel, V. Lonjou, S. Marcq, C. Desjardins, B. Fougnie, C. Coppolani-Delle Luche, N. Guilleminot, A.-S. Lacamp, E. Lourme, C. Miquel, and et al., "Sentinel-2A and 2B absolute calibration monitoring", *European Journal of Remote Sensing*, vol. 52, no. 1, pp. 122–137, Jan. 2019. DOI: 10.1080/22797254.2018.1562311.

[149] S. Roberts, M. Osborne, M. Ebden, S. Reece, N. Gibson, and S. Aigrain, "Gaussian processes for time-series modelling", *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 371, no. 1984, Feb. 2013. DOI: 10.1098/rsta.2011.0550.

[150] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation", in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds., ser. Lecture Notes in Computer Science, Cham: Springer International Publishing, 2015, pp. 234–241. DOI: 10.1007/978-3-319-24574-4_28.

[151] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain", *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958. DOI: 10.1037/h0042519.

[152] M. Rußwurm and M. Körner, "Multi-temporal land cover classification with long short-term memory neural networks", *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLII-1/W1, pp. 551–558, May 2017. DOI: 10.5194/isprs-archives-xlii-1-w1-551-2017.

[153] J. L. Schafer, *Analysis of Incomplete Multivariate Data*, en. Chapman and Hall/CRC, 1997.

[154] A. Schmutz, J. Jacques, C. Bouveyron, L. Chèze, and P. Martin, "Clustering multivariate functional data in group-specific functional subspaces", *Computational Statistics*, vol. 35, no. 3, pp. 1101–1131, Sep. 2020.

[155] J. R. Schott, *Matrix Analysis for Statistics.* Ser. Wiley Series in Probability and Statistics. Wiley, 2016, ISBN: 9781119092469.

[156] A. Shah, A. Wilson, and Z. Ghahramani, "Student-t Processes as Alternatives to Gaussian Processes", in *Artificial Intelligence and Statistics*, vol. 33, PMLR, Apr. 2014, pp. 877–885.

[157] Z. Shao, J. Cai, P. Fu, L. Hu, and T. Liu, "Deep learning-based fusion of Landsat-8 and Sentinel-2 images for a harmonized surface reflectance product", *Remote Sensing of Environment*, vol. 235, p. 111 425, Dec. 2019. DOI: 10.1016/j.rse.2019.111425.

[158] H. Shen, X. Li, Q. Cheng, C. Zeng, G. Yang, H. Li, and L. Zhang, "Missing Information Reconstruction of Remote Sensing Data: A Technical Review", *IEEE Geoscience and Remote Sensing Magazine*, vol. 3, no. 3, pp. 61–85, Sep. 2015. DOI: 10.1109/MGRS.2015.2441912.

[159] Y. Shendryk, Y. Rist, C. Ticehurst, and P. Thorburn, "Deep learning for multi-modal classification of cloud, shadow and land cover scenes in planetscope and Sentinel-2 imagery", *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 157, pp. 124–136, Nov. 2019. DOI: 10.1016/j.isprsjprs.2019.08.018.

[160] M. Sheykhmousa, M. Mahdianpari, H. Ghanbari, F. Mohammadimanesh, P. Ghamisi, and S. Homayouni, "Support Vector Machine Versus Random Forest for Remote Sensing Image Classification: A Meta-Analysis and Systematic Review", *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 6308–6325, 2020. DOI: 10.1109/JSTARS.2020.3026724.

[161] S. Skakun, E. Vermote, J.-C. Roger, and C. Justice, "Multispectral Misregistration of sentinel-2A Images: Analysis and Implications for Potential Applications", *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 12, pp. 2408–2412, 2017. DOI: 10.1109/lgrs.2017.2766448.

[162] G. Skolidis and G. Sanguinetti, "Bayesian Multitask Classification With Gaussian Process Priors", *IEEE Transactions on Neural Networks*, vol. 22, no. 12, pp. 2011–2021, Dec. 2011. DOI: 10.1109/TNN.2011.2168568.

[163] P. Soille, A. Burger, D. D. Marchi, P. Kempeneers, D. Rodriguez, V. Syrris, and V. Vasilev, "A versatile data-intensive computing platform for information retrieval from big geospatial data", *Future Generation Computer Systems*, vol. 81, pp. 30–40, 2018. DOI: 10.1016/j.future.2017.11.007.

[164] J. Spinnato, "Modèles de covariance pour l'analyse et la classification de signaux électroencéphalogrammes", Thèse de doctorat, Aix-Marseille, Jul. 2015.

[165] J. Spinnato, M.-C. Roubaud, B. Burle, and B. Torrésani, "Finding EEG space-time-scale localized features using matrix-based penalized discriminant analysis", in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 6004–6008. DOI: 10.1109/ICASSP.2014.6854756.

[166] M. S. Srivastava, T. von Rosen, and D. von Rosen, "Models with a kronecker product covariance structure: Estimation and testing", *Mathematical Methods of Statistics*, vol. 17, no. 4, pp. 357–370, 2008.

[167] M. Sudmanns, D. Tiede, S. Lang, H. Bergstedt, G. Trost, H. Augustin, A. Baraldi, and T. Blaschke, "Big Earth data: Disruptive changes in earth observation data management and analysis?", *International Journal of Digital Earth*, pp. 1–19, Mar. 2019. DOI: 10.1080/17538947.2019.1585976.

[168] V. Syrris, P. Hasenohr, B. Delipetrev, A. Kotsev, P. Kempeneers, and P. Soille, "Evaluation of the Potential of Convolutional Neural Networks and Random Forests for Multi-Class Segmentation of Sentinel-2 Imagery", *Remote Sensing*, vol. 11, no. 8, p. 907, Apr. 2019. DOI: 10.3390/rs11080907.

[169] J. M. Tabeart, S. L. Dance, S. A. Haben, A. S. Lawless, N. K. Nichols, and J. A. Waller, "The conditioning of least-squares problems in variational data assimilation: The conditioning of least squares problems in variational data assimilation", *Numerical Linear Algebra with Applications*, vol. 25, no. 5, pp. 2165–2187, Oct. 2018. DOI: 10.1002/nla.2165.

[170] S. Talukdar, P. Singha, S. Mahato, Shahfahad, S. Pal, Y.-A. Liou, and A. Rahman, "Land-Use Land-Cover Classification by Machine Learning Classifiers for Satellite Observations—a Review", *Remote Sensing*, vol. 12, no. 7, p. 1135, Jan. 2020. DOI: 10.3390/rs12071135.

[171] O. D. Team, *Orfeo ToolBox 7.1*, version 7.1.0, Mar. 2020. DOI: 10.5281/zenodo.3715021. [Online]. Available: https://doi.org/10.5281/zenodo.3715021.

[172] Y. W. Teh, M. Seeger, and M. I. Jordan, "Semiparametric latent factor models", in *International Workshop on Artificial Intelligence and Statistics*, PMLR, Jan. 2005, pp. 333–340.

[173] P. Thanh Noi and M. Kappas, "Comparison of Random Forest, k-Nearest Neighbor, and Support Vector Machine Classifiers for Land Cover Classification Using Sentinel-2 Imagery", *Sensors*, vol. 18, no. 1, p. 18, Jan. 2018. DOI: 10.3390/s18010018.

[174] A. Tharwat, "Classification assessment methods", *Applied Computing and Informatics*, vol. 17, no. 1, pp. 168–192, Jan. 2021. DOI: 10.1016/j.aci.2018.08.003.

[175] P. Theodossiou, "Financial Data and the Skewed Generalized t Distribution", *Management Science*, vol. 44, no. 12-Part-1, pp. 1650–1661, 1998.

[176] G. Z. Thompson, R. Maitra, W. Q. Meeker, and A. F. Bastawros, "Classification With the Matrix-Variate-t Distribution", *Journal of Computational and Graphical Statistics*, vol. 29, no. 3, pp. 668–674, Jul. 2020. DOI: 10.1080/10618600.2019.1696208.

[177] J. Tronicke and U. Böniger, "Steering kernel regression: An adaptive denoising tool to process GPR data", in *2013 7th International Workshop on Advanced Ground Penetrating Radar*, Jul. 2013, pp. 1–4. DOI: 10.1109/IWAGPR.2013.6601539.

[178] C. J. Tucker, "Red and photographic infrared linear combinations for monitoring vegetation", *Remote Sensing of Environment*, vol. 8, no. 2, pp. 127–150, May 1979. DOI: 10.1016/0034-4257(79)90013-0.

[179] M. O. Ulfarsson, F. Palsson, M. Dalla Mura, and J. R. Sveinsson, "Sentinel-2 Sharpening Using a Reduced-Rank Method", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 9, pp. 6408–6420, Sep. 2019. DOI: 10.1109/tgrs.2019.2906048.

[180] J. Useya and S. Chen, "Comparative Performance Evaluation of Pixel-Level and Decision-Level Data Fusion of Landsat 8 OLI, Landsat 7 ETM+ and Sentinel-2 MSI for Crop Ensemble Classification", *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 11, pp. 4441–4451, Nov. 2018. DOI: 10.1109/jstars.2018.2870650.

[181] A. Verhegghen, R. d'Andrimont, F. Waldner, and M. Van der Velde, "Accuracy assessment of the first EU-wide crop type map with LUCAS data", in *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, 2021, pp. 1990–1993. DOI: 10.1109/IGARSS47720.2021.9553758.

[182] J. Verrelst, L. Alonso, G. Camps-Valls, J. Delegido, and J. Moreno, "Retrieval of Vegetation Biophysical Parameters Using Gaussian Process Techniques", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 5, pp. 1832–1843, May 2012. DOI: 10.1109/tgrs.2011.2168962.

[183] N. Viovy, O. Arino, and A. S. Belward, "The Best Index Slope Extraction (BISE): A method for reducing noise in NDVI time-series", *International Journal of Remote Sensing*, vol. 13, no. 8, pp. 1585–1590, May 1992. DOI: 10.1080/01431169208904212.

[184] F. Vuolo, W.-T. Ng, and C. Atzberger, "Smoothing and gap-filling of high resolution multi-spectral time series: Example of Landsat data", *International Journal of Applied Earth Observation and Geoinformation*, vol. 57, pp. 202–213, May 2017. DOI: 10.1016/j.jag.2016.12.012.

[185] B. Wang, K. Jia, S. Liang, X. Xie, X. Wei, X. Zhao, Y. Yao, and X. Zhang, "Assessment of Sentinel-2 MSI Spectral Band Reflectances for Estimating Fractional Vegetation Cover", *Remote Sensing*, vol. 10, no. 12, p. 1927, Dec. 2018. DOI: 10.3390/rs10121927.

[186] J. Wang, B. Huang, H. K. Zhang, and P. Ma, "Sentinel-2A Image Fusion Using a Machine Learning Approach", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 12, pp. 9589–9601, Dec. 2019. DOI: 10.1109/tgrs.2019.2927766.

[187] A. K. Whitcraft, I. Becker-Reshef, and C. O. Justice, "A Framework for Defining Spatially Explicit Earth Observation Requirements for a Global Agricultural Monitoring Initiative (GEOGLAM)", *Remote Sensing*, vol. 7, no. 2, pp. 1461–1481, Feb. 2015. DOI: 10.3390/rs70201461.

[188] A. K. Whitcraft, I. Becker-Reshef, B. D. Killough, and C. O. Justice, "Meeting Earth Observation Requirements for Global Agricultural Monitoring: An Evaluation of the Revisit Capabilities of Current and Planned Moderate Resolution Optical Earth Observing Missions", *Remote Sensing*, vol. 7, no. 2, pp. 1482–1503, Feb. 2015. DOI: 10.3390/rs70201482.

[189] C. K. I. Williams, "Computation with infinite neural networks", *Neural Computation*, vol. 10, no. 5, pp. 1203–1216, Jul. 1998, ISSN: 0899-7667. DOI: 10.1162/089976698300017412.

[190] C. K. I. Williams and C. E. Rasmussen, "Gaussian processes for machine learning", *the MIT Press*, vol. 2, no. 3, p. 4, 2006.

[191] C. Williams and M. Seeger, "Using the nyström method to speed up kernel machines", in *Advances in neural information processing systems*, vol. 13, MIT Press, 2001.

[192] L. Yu, J. Wang, and P. Gong, "Improving 30 m global land-cover map FROM-GLC with time series MODIS and auxiliary data sets: A segmentation-based approach", *International Journal of Remote Sensing*, vol. 34, no. 16, pp. 5851–5867, Aug. 2013. DOI: 10.1080/01431161.2013.798055.

[193] N. Zang, Y. Cao, Y. Wang, B. Huang, L. Zhang, and P. T. Mathiopoulos, "Land-Use Mapping for High-Spatial Resolution Remote Sensing Image Via Deep Learning: A Review", *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 5372–5391, 2021. DOI: 10.1109/JSTARS.2021.3078631.

[194] Q. Zhang, Q. Yuan, J. Li, Z. Li, H. Shen, and L. Zhang, "Thick cloud and cloud shadow removal in multitemporal imagery using progressively spatio-temporal patch group deep learning", *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 162, pp. 148–160, 2020. DOI: https://doi.org/10.1016/j.isprsjprs.2020.02.008.

[195] Q. Zhang, Q. Yuan, C. Zeng, X. Li, and Y. Wei, "Missing Data Reconstruction in Remote Sensing Image With a Unified Spatial–Temporal–Spectral Deep Convolutional Neural Network", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 8, pp. 4274–4288, Aug. 2018. DOI: 10.1109/TGRS.2018.2810208.

[196] T. Zhang, "Solving Large Scale Linear Prediction Problems Using Stochastic Gradient Descent Algorithms", in *ICML 2004: Proceedings of the Twenty-First International Conference on Machine Learning. Omnipress*, 2004, pp. 919–926.

[197] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, "Algorithm 778: L-BFGS-b: Fortran subroutines for large-scale bound-constrained optimization", *ACM Transactions on Mathematical Software*, vol. 23, no. 4, pp. 550–560, Dec. 1997. DOI: 10.1145/279232.279236.

[198] L. Zhu, Q. Hu, J. Yang, J. Zhang, P. Xu, and N. Ying, "EEG Signal Classification Using Manifold Learning and Matrix-Variate Gaussian Model", *Computational Intelligence and Neuroscience*, vol. 2021, p. 12, Mar. 2021. DOI: 10.1155/2021/6668859.