Classification de séries temporelles massives d'images satellitaires par des processus gaussiens variationnels parcimonieux et des descripteurs spatio-spectro-temporels

Valentine BELLET [*], Mathieu FAUVEL, Jordi INGLADA

CESBIO, Université de Toulouse, CNES/CNRS/INRAe/IRD/UPS
18 Avenue Édouard Belin, 31400 Toulouse, FRANCE
valentine.bellet@univ-toulouse.fr, mathieu.fauvel@inrae.fr
jordi.inglada@cesbio.eu

Résumé – Cet article présente une approche permettant la classification pixellaire de séries temporelles d'images satellites massives à partir de processus gaussiens. Notre modèle s'appuie sur des processus gaussiens variationnels parcimonieux associés à des fonctions de covariance spatio-spectro-temporelles. Les résultats expérimentaux réalisés sur un jeu de données à l'échelle nationale montrent que notre méthode est efficace et comparable à des approches de l'état de l'art (forêt d'arbres décisionnels et méthodes d'apprentissage profond).

Abstract – This paper introduces an approach for land cover pixel-based classification from massive satellite image time series scale using Gaussian Processes. Sparse methods combined with variational inference allow the learning of large scale Gaussian Process. Using a spatio-spectro-temporal covariance function can help to take into account all the structure of the data. Experimental results conducted on a national scale data-set show that our method is effective and close to several state-of-the-art approaches (random forest and deep learning methods).

1 Introduction

Lors de cette dernière décennie, l'avènement des missions satellitaires d'observation de la Terre à haute fréquence de revisite et à haute résolution spatiale a conduit à la mise à disposition d'une quantité sans précédente d'images de nature physique hétérogène (optique, radar, etc.) et à différentes échelles (submétrique, décamétrique, etc.). Ces données massives permettent de comprendre, d'expliquer et de prédire l'état et le fonctionnement de nos écosystèmes. Ces données constituent ainsi une source d'information majeure pour répondre aux enjeux du changement climatique [1].

Cependant, en raison du volume de données à analyser, l'utilisation d'algorithmes automatiques est nécessaire pour les exploiter efficacement. Malheureusement, la complexité croissante des données à traiter limite la capacité des méthodes statistiques classiques à extraire et à traiter toute l'information pertinente. Les méthodes d'apprentissage statistique, et en particulier l'apprentissage profond, ont permis des avancées opérationnelles conséquentes sur différentes tâches comme la classification de l'occupation des sols par exemple [2].

Après une période d'intense développement en télédétection, les méthodes à noyaux reproduisants [3] ont été délaissées au profit d'algorithmes d'apprentissage profond tels que les réseaux de neurones et les méthodes associées (auto-encoder, transformer, etc.) [4]. Ces derniers algorithmes ont bénéficié des développements de la puissance calculatoire disponible (GPU, HPC, etc.), de la disponibilité de jeux de données massifs et de mises en œuvre efficaces, polyvalentes et ouvertes (e.g. Pytorch ou Tensorflow). Cela permet aux réseaux d'apprentissage profond (*Deep Neural Networks*-DNN) de traiter efficacement de gros volumes de données, là où les méthodes à noyaux conventionnelles sont limitées – algorithmiquement et numériquement – par le nombre d'échantillons à traiter.

Actuellement, les DNN sont utilisés avec succès dans de nombreux domaines de recherche en télédétection [1, 4]. Toutefois, l'entraînement des DNN reste une tâche délicate pour éviter des problèmes de sur-apprentissage en raison d'un grand nombre de paramètres à optimiser et d'un coût calculatoire qui reste important. Une autre limite des DNN est le coté « boîte noire » des modèles, dont il est difficile d'obtenir des informations sur le processus étudié [4].

Les processus gaussiens (GP) intègrent dans un cadre Bayésien les méthodes à noyaux reproduisants [5]. Ils ont été appliqués avec succès en télédétection pour de la classification ou de l'estimation de paramètres [6]. Contrairement aux DNN, le nombre de paramètres à estimer est limité et leurs valeurs après apprentissage renseignent sur le processus étudié (e.g. longueur de corrélation temporelle). Les formulations conventionnelles des GP restent cependant très limitées par rapport au nombre n d'échantillons d'entraînement (quelques milliers de points). En effet, l'apprentissage du modèle ainsi que l'inférence sont de complexité cubique par rapport à n ($\mathcal{O}(n^3)$).Plusieurs approches ont été proposées dans la littérature ces dernières années, [7] en propose une revue détaillée. L'approche par inférence variationnelle proposée dans [8] et étendue dans [9] permet notamment de réduire la complexité en $\mathcal{O}(nm^2)$ avec $m \ll n$, m correspondant au nombre de pseudo-points. Elle s'appuie sur une fonction de coût minimisable par descente de gradient stochastique, l'utilisation de GPU et une formulation

[*]Ce travail est co-financé par le CNES et CS-Group dans le cadre du projet ANR-3IA ANITI-19-PI3A-0004 de l'Université Fédérale Toulouse Midi-Pyrénées.

parcimonieuse de l'opérateur de covariance.

Dans ce travail, nous avons évalué ces GP variationnels parcimonieux sur des séries temporelles massives d'images satellitaires pour de la classification pixellaire. Deux fonctions de covariances prenant en compte la structure de la donnée sont analysées. Les résultats en terme de classification sont comparés à des méthodes issues de la littérature en télédétection.

2 GP pour la classification

Par la suite $\mathcal{S} = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^n$ représente l'ensemble des n pixels référencés avec $\mathbf{x}_i \in \mathbb{R}^d$, $\mathbf{y}_i \in \Omega$, $\Omega \subseteq \mathbb{R}^p$, $p \geq 1$. $\mathbf{x}_i$ sont les prédicteurs et $\mathbf{y}_i$ sont les variables à prédire. On notera f un processus stochastique gaussien univarié à valeurs dans $\mathbb{R}$ caractérisé par sa fonction moyenne μ et sa fonction de covariance, ou noyau, k [5] : $f \sim \mathcal{GP}(\mu, k)$. Par définition, tout ensemble fini $f(\mathbf{X}) = [f(\mathbf{x}_1), ..., f(\mathbf{x}_N)]^\top$ de taille N suit une distribution multivariée gaussienne : $f(\mathbf{X}) \sim \mathcal{N}_N(\mathbf{m}, \mathbf{\Sigma})$ avec $\mathbf{m} = [\mu(\mathbf{x}_1), ..., \mu(\mathbf{x}_N)]^\top$ et $\Sigma_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, $\forall i, j \in \{1, \dots, N\}^2$. De manière analogue, on notera $\mathbf{f}$ un processus gaussien multivarié à valeurs dans $\mathbb{R}^p$, de fonction moyenne $\boldsymbol{\mu} \in \mathbb{R}^p$ et de fonction de covariance $\mathbf{K} \in \mathbb{R}^{p \times p}$: $\mathbf{f} \sim \mathcal{GP}(\boldsymbol{\mu}, \mathbf{K})$. Plusieurs approches existent pour construire des processus multivariés [10]. Dans la suite de notre article, nous construirons ces processus comme des combinaisons linéaires de L processus univariés latents indépendants (modèles additifs) [11] : $\mathbf{f} = \mathbf{M}\mathbf{g}$ avec $\mathbf{M} \in \mathbb{R}^{p \times L}$, $\mathbf{g} = [g_1, \dots, g_L]^\top$, $g_l \sim \mathcal{GP}(\mu_l, k_l)$ et $g_l \perp\!\!\!\perp g_{l'}$ $\forall (l, l') \in \{1, \dots, L\}^2$ et $l \neq l'$. Afin d'alléger les notations, nous omettrons par la suite $\mathbf{g}$ quand cela ne nuira pas à la compréhension. Nous supposons que le lien entre $\mathbf{x}$ et $\mathbf{y}$ peut se formaliser par un processus gaussien latent et un modèle d'observation liant le processus latent à la variable à prédire.

2.1 GP variationnels parcimonieux

Pour un problème de classification à C classes, $\mathbf{y}_i \in \{0, 1\}^C$ et ses composantes sont nulles sauf $y_{ic} = 1$ pour $\mathbf{x}_i$ appartenant à la classe c. Dans ce cadre, le modèle d'observation entre la variable latente $\mathbf{f}_i = [f_1(\mathbf{x}_i), ..., f_C(\mathbf{x}_i)]^\top$ et la variable observée $\mathbf{y}_i$ est de type softmax [9] :

$$p(\mathbf{y}_i|\mathbf{f}_i) = \prod_{c=1}^{C} \left[\frac{\exp(f_{ic})}{\sum_{c'=1}^{C} \exp(f_{ic'})} \right]^{y_{ic}} = \frac{\exp(\mathbf{f}_i^\top \mathbf{y}_i)}{\sum_{c'=1}^{C} \exp(f_{ic'})}. \tag{1}$$

En supposant les échantillons d'entraînement *i.i.d.*, et en notant $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_n]^\top$ et $\mathbf{F} = [\mathbf{f}_1, \dots, \mathbf{f}_n]^\top$, la vraisemblance totale s'écrit alors :

$$p(\mathbf{Y}, \mathbf{F}, \mathbf{X}) = p(\mathbf{Y}|\mathbf{F}) p(\mathbf{F}|\mathbf{X}) = \prod_{i=1}^{n} p(\mathbf{y}_i|\mathbf{f}_i) \times p(\mathbf{F}|\mathbf{X}). \tag{2}$$

Par définition, nous avons $p(\mathbf{F}|\mathbf{X}) = \mathcal{N}_{nC}(0, \mathbf{\Sigma}_{\mathbf{X},\mathbf{X}})$. À partir de cette équation, deux quantités doivent être estimées en pratique [5] : la vraisemblance marginale (ou évidence) $p(\mathbf{Y})$, pour optimiser les hyperparamètres du modèle (e.g. $\boldsymbol{\theta}$ les paramètres de la fonction de covariance et de la fonction moyenne), et la distribution *a posteriori* $p(\mathbf{f}_*|\mathbf{x}_*, \mathbf{X}, \mathbf{Y})$ associée à un nouvel échantillon $\mathbf{x}_*$ à classer. Malheureusement, il n'est pas possible d'intégrer analytiquement (2) et les approximations conventionnelles sont trop coûteuses numériquement pour n supérieur à quelques milliers [5]. De plus, dans le cadre de modèles additifs, il est nécessaire d'inverser $\mathbf{\Sigma}_{\mathbf{X},\mathbf{X}}$, qui est de complexité $\mathcal{O}(n^3 C)$.

Les processus gaussiens parcimonieux ont été ainsi proposés comme une approximation de rang faible de $\mathbf{\Sigma}_{\mathbf{X},\mathbf{X}}$ à partir de m pseudo-points $\mathbf{Z} = \{\mathbf{z}_i\}_{i=1}^m$, de sorte que $\mathbf{\Sigma}_{\mathbf{X},\mathbf{X}} \approx \mathbf{\Sigma}_{\mathbf{X},\mathbf{Z}} \mathbf{\Sigma}_{\mathbf{Z},\mathbf{Z}}^{-1} \mathbf{\Sigma}_{\mathbf{Z},\mathbf{X}}$ [5, Chap 8.3]. La complexité est réduite à $\mathcal{O}(nm^3 C)$, mais le choix des pseudo-points est crucial : l'approximation n'est correcte que pour les points proches des $\mathbf{z}_i$ dans l'espace d'entrée, et il est donc important de choisir des $\mathbf{z}_i$ qui représentent correctement les données d'entrée. En pratique, ce choix est complexe.

[12] puis [8] ont proposé une formulation variationnelle qui permet d'optimiser à la fois les paramètres du modèle et les pseudo-points à partir de la vraisemblance marginale. Wilson *et al.* ont étendu cette formulation pour le cas multivarié dans [9]. En notant $\mathbf{u}_l = \{g_l(\mathbf{z}_{li})\}_{i=1}^m$ les pseudo-variables latentes associées aux pseudo-points du processus gaussien g_l et en choisissant $q(\mathbf{u}_l) = \mathcal{N}_m(\mathbf{u}_l|\mathbf{m}_l, \mathbf{S}_l)$ pour la loi variationnelle, la borne variationnelle suivante est obtenue (ELBO) :

$$\log p(\mathbf{Y}) \geq \sum_{i=1}^{n} \mathbb{E}_{q(\mathbf{g}_i)} \big[\log p(\mathbf{y}_i|\mathbf{M}\mathbf{g}_i) \big] - \sum_{l=1}^{L} \mathrm{KL}[q(\mathbf{u}_l)||p(\mathbf{u}_l)]. \tag{3}$$

Par construction, nous avons $p(\mathbf{u}_l) = \mathcal{N}_l(\boldsymbol{\mu}_l, \mathbf{\Sigma}_{\mathbf{Z},\mathbf{Z}}^l)$ et $q(\mathbf{g}_i) = \prod_{l=1}^{L} q\big(g_l(\mathbf{x}_i)\big)$ avec $q\big(g_l(\mathbf{x}_i)\big)$ la loi marginale de $q\big(g_l(\mathbf{X})\big) = \int p\big(g_l(\mathbf{X})|\mathbf{u}_l\big) q(\mathbf{u}_l) d\mathbf{u}_l$:

$$q\big(g_l(\mathbf{X})\big) \sim \mathcal{N}_n\Big(\mathbf{A}_l \mathbf{m}_l, \mathbf{\Sigma}_{\mathbf{X},\mathbf{X}}^l + \mathbf{A}_l(\mathbf{S}_l - \mathbf{\Sigma}_{\mathbf{Z},\mathbf{Z}}^l)\mathbf{A}_l^\top\Big), \tag{4}$$

avec $\mathbf{A}_l = \mathbf{\Sigma}_{\mathbf{X},\mathbf{Z}}^l \mathbf{\Sigma}_{\mathbf{Z},\mathbf{Z}}^{l^{-1}}$ [9].

L'ELBO est composée de deux termes : un terme d'attache aux données (premier terme de la partie droite de (3)) et un terme de régularisation (second terme). Le terme de régularisation est la divergence de Kullback-Leibler entre l'*a priori* et la loi *a posteriori* variationnelle : cela contraint ces deux lois à rester proches au sens de cette divergence. Il se calcule explicitement pour le cas de lois normales multivariées (ainsi que leurs dérivées). L'attache aux données obtenue pour cette borne est simplement l'entropie croisée entre $\mathbf{f}_i$ et $\mathbf{y}_i$ que l'on retrouve classiquement comme fonction de coût d'un réseau de neurones pour la classification. Cependant, son calcul n'est pas explicite ici et une estimation est calculée par échantillonnage de Monte Carlo à l'aide des lois variationnelles des $q(\mathbf{u}_l)$ et d'une reparamétrisation pour pouvoir dériver ensuite l'ELBO par rapport aux paramètres du modèle (à la manière de ce qui est fait dans un auto-encodeur variationnel).

Les paramètres du modèle sont l'ensemble des paramètres variationnels ($\mathbf{Z}_l$, $\mathbf{m}_l$, $\mathbf{S}_l$ $\forall l \in \{1, \dots, L\}$), les hyperparamètres des GP ($\boldsymbol{\theta}_l$, $\forall l \in \{1, \dots, L\}$) et $\mathbf{M}$ la matrice de relation linéaire entre les GP. Il est ainsi possible d'optimiser le

modèle par descente de gradient stochastique classique. En pratique, un seul tirage par échantillon est effectué pour l'échantillonnage.

La classification d'un nouvel échantillon $\mathbf{x}_*$ s'effectue à l'aide du *maximum a posteriori* (MAP) : $\hat{c} = \arg\max_c \mathbf{y}_*$. On obtient $\mathbf{y}_*$ par

$$p(\mathbf{y}_*|\mathbf{x}_*, \mathbf{Y}, \mathbf{X}) = \int p(\mathbf{y}_*|\mathbf{f}_*)p(\mathbf{f}_*|\mathbf{x}_*, \mathbf{y}, \mathbf{X})d\mathbf{f}_* \quad (5)$$

$$p(\mathbf{f}_*|\mathbf{x}_*, \mathbf{Y}, \mathbf{X}) \approx \int \prod_{l=1}^{L} p(\mathbf{f}_*|\mathbf{u}_l)q(\mathbf{u}_l)d\mathbf{u}_l \quad (6)$$

L'équation (5) n'est pas intégrable analytiquement : comme précédent, une approximation est obtenue par échantillonnage de Monte Carlo (avec 10 tirages).

2.2 Covariance spatio-spectro-temporelle

La fonction de covariance est l'élément central dans les GP : c'est cette fonction qui calcule la similarité entre deux échantillons et il est souvent bénéfique de prendre en compte la structure des données dans sa définition. En télédétection, il a été montré que combiner la structure spatiale et spectro-temporelle des données permet d'améliorer les résultats [13]. On propose ici de définir la fonction de covariance $k_l(\mathbf{x}, \mathbf{x}')$ comme la combinaison d'une fonction de covariance spatiale $k_\phi(\mathbf{x}_\phi, \mathbf{x}'_\phi)$ avec $\mathbf{x}_\phi$ les données spatiales et d'une fonction de covariance spectro-temporelle $k_{\lambda t}(\mathbf{x}_{\lambda t}, \mathbf{x}'_{\lambda t})$ avec $\mathbf{x}_{\lambda t}$ les données spectro-temporelles : $\mathbf{x} = [\mathbf{x}_\phi, \mathbf{x}_{\lambda t}]^\top$. Nous avons investigué deux combinaisons, la somme : $k^S(\mathbf{x}, \mathbf{x}') = \alpha^2_\phi \times k_\phi(\mathbf{x}_\phi, \mathbf{x}'_\phi) + \alpha^2_{\lambda t} \times k_{\lambda t}(\mathbf{x}_{\lambda t}, \mathbf{x}'_{\lambda t})$, et le produit : $k^P(\mathbf{x}, \mathbf{x}') = k_\phi(\mathbf{x}_\phi, \mathbf{x}'_\phi) \times k_{\lambda t}(\mathbf{x}_{\lambda t}, \mathbf{x}'_{\lambda t})$. α_ϕ et $\alpha_{\lambda t}$ sont des paramètres de pondération qui permettent de donner plus ou moins de poids à chacune des deux composantes (spatiale ou spectro-temporelle). Ils sont optimisés durant l'apprentissage.

3 Expériences et résultats

3.1 Séries temporelles d'images satellites

La zone d'étude est située au sud de la France métropolitaine et s'étend sur une zone de $200\,000\,km^2$. Les données utilisées sont des séries temporelles d'images satellites (SITS) de Sentinel-2 de niveau 2A acquises entre janvier et décembre 2018 (téléchargées via le Theia Data Center [1]). Les mêmes traitements que dans [2] ont été appliqués aux données (interpolation linéaire sur une grille temporelle régulière avec un pas de 10 jours). En plus des composantes spectro-temporelles, les coordonnées spatiales d'un pixel sont extraites : longitude et latitude (projection Lambert 93 en mètres). Chaque pixel $\mathbf{x}_i$ est décrit par 483 composantes : 13 composantes spectrales pour chacune des 37 dates ($\mathbf{x}_{i\lambda t} \in \mathbb{R}^{481}$) ainsi que les 2 composantes spatiales ($\mathbf{x}_{i\phi} \in \mathbb{R}^2$). La donnée de référence utilisée est définie dans [2]. Elle est composée de 23 classes (i.e., $\mathbf{y}_i \in \mathbb{R}^{23}$) dont la liste est résumée dans le Tableau 1. L'emprise totale correspond à un total d'environ 2 milliards de pixels.

1. https://www.theia-land.fr/en/products/

3.2 Protocole expérimental

Préparation données La donnée de référence est découpée à l'échelle du polygone en trois jeux distincts spatialement : *train*, *val* et *test*. Les jeux de données *train* et *val* sont utilisés pour entraîner/valider les modèles d'apprentissage. Le jeu de données *test* permet d'évaluer les performances de classification des modèles. Les jeux de données sont construits de façon la plus équilibrée possible en fonction du nombre d'échantillons disponibles pour chaque classe. Ils ont été générés 11 fois avec une graine aléatoire différente. Le nombre moyen d'échantillons pour chaque jeu de données est résumé dans le Tableau 1. Les données ont été standardisées à partir du jeu *train*.

Apprentissage La mise en œuvre du modèle *GP* défini dans la Section 2 a été faite à l'aide de la librairie *Gpytorch* [14]. Nous avons choisi de travailler avec $L = C$ processus latents. Le noyau *RBF* ($k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x}-\mathbf{x}'\|^2}{2\ell^2}\right)$) a été choisi pour les fonctions de covariance k_ϕ et $k_{\lambda t}$ de chaque processus latent. Leurs paramètres respectifs ont été initialisés avec les valeurs suivantes : $\ell_\phi = \sqrt{2}$ et $\ell_{\lambda t} = \sqrt{481}$. Pour k^S, on initialise le paramètre d'échelle avec : $\alpha_\phi = \alpha_{\lambda t} = \exp(1)$. La fonction moyenne choisie est une constante initialisée à zéro. Le nombre de points induits sélectionné pour chaque processus latent est égal à $m = 50$ après plusieurs essais ($m = \{30, 50, 100, 250, 500\}$). Ces derniers sont initialisés avec des échantillons aléatoirement choisis parmi le jeu de données *train*. Finalement, la matrice $\mathbf{M}$, qui est la relation linéaire entre les GPs est initialisée avec des valeurs aléatoires suivant une loi normale centrée réduite.

Ce modèle est comparé avec les modèles suivants : 1) *RF* : *Random Forest* dont la mise en œuvre a été faite avec le modèle *RandomForestClassifier* issu de la librairie *Scikit-learn* [15]. Les paramètres par défaut ont été gardés (100 arbres). 2) *MLP* : *MultiLayer Perceptron*, avec 4 couches cachées et la fonction d'activation *Relu*. 3) *LTAE* : *Light Temporal Attention Encoder*, encodeur temporel utilisant la self-attention permettant de prendre en compte la structure spectro-temporelle de la donnée. Sa description ainsi que les paramètres utilisés sont décrits dans [16]. Les mises en œuvre du *MLP* et du *LTAE* ont été faites à l'aide de la librairie *Pytorch* [17]. Les paramètres d'optimisation (Adam) des modèles *GP*, *MLP* et *LTAE* ont été fixés par essai-erreur (cf. Tableau 2).

3.3 Résultats

Les résultats issus de l'analyse statistique de la performance de classification du modèle sont résumés dans le Tableau 1. Les *GP* ont des résultats comparables aux autres modèles utilisés. En terme d'overall accuracy, les *GP* sont un point au dessus des *RF* mais un point en dessous des *MLP* ou deux points en dessous des *LTAE*. Pour tous les modèles étudiés, l'ajout de l'information spatiale permet une amélioration des performances (augmentation d'au minimum un point pour l'overall accuracy entre chaque modèle). Pour la fonction de covariance, le pro-

TABLE 1 – Liste des classes et leur nombre moyen d'échantillons pour les 3 jeux de données. Moyenne $\pm$ écart type du f-score de chaque classe, de précision globale et du f-score moyen (en %) pour les différents modèles étudiés. On note *NS* le modèle prenant en entrée uniquement $\mathbf{x}_{\lambda t}$ et *S* le modèle prenant en entrée $\mathbf{x}_{\phi}$ et $\mathbf{x}_{\lambda t}$. On note *S-GPSC*, *S-GPPC* les modèles dont la fonction de covariance respective est la somme ou le produit (cf. Section 2). ■,■,■ indiquent les 3 meilleurs résultats au sens de la moyenne.

Nom	**Nombre moyen d'échantillons**			**NS-GP**	**S-GPSC**	**S-GPPC**	**NS-RF**	**S-RF**	**NS-MLP**	**S-MLP**	**NS-LTAE**	**S-LTAE**
	train	*val*	*test*	f-score par classe								
Bâti dense	32 000	8 000	77 286	62.7 ± 2.7	65.5 ± 1.2	65.8 ± 0.5	60.8 ± 0.2	62.5 ± 0.2	65.2 ± 0.4	66.4 ± 0.3	64.7 ± 0.3	67.9 ± 0.4
Bâti diffu	32 000	8 000	80 000	61.2 ± 1.6	64.1 ± 1.8	63.6 ± 1.5	57.3 ± 0.2	58.6 ± 0.2	63.8 ± 0.6	65.1 ± 0.2	62.9 ± 0.4	66.9 ± 0.5
Zones indus. et commer.	32 000	8 000	80 000	51.1 ± 2.8	54.3 ± 1.5	54.2 ± 1.3	49.8 ± 0.2	51.7 ± 0.3	54.1 ± 0.4	56.0 ± 0.3	53.3 ± 0.4	58.4 ± 0.4
Routes	30 502	7 625	71 982	77.7 ± 0.9	79.4 ± 0.7	80.0 ± 0.7	76.2 ± 0.2	77.1 ± 0.3	80.2 ± 0.3	81.5 ± 0.3	80.6 ± 0.4	83.4 ± 0.3
Colza	32 000	7 929	74 551	95.0 ± 0.5	92.9 ± 6.3	93.9 ± 2.2	94.5 ± 0.3	94.6 ± 0.3	94.5 ± 0.4	94.4 ± 0.6	94.8 ± 0.5	95.2 ± 0.4
Céréales à paille	32 000	8 000	80 000	87.4 ± 0.6	87.3 ± 1.1	87.0 ± 0.4	85.5 ± 0.2	85.7 ± 0.2	87.6 ± 0.3	87.7 ± 0.4	89.2 ± 0.3	89.5 ± 0.3
Protéagineux	26 262	6 704	64 342	74.6 ± 1.5	73.7 ± 4.1	74.3 ± 1.6	70.4 ± 0.4	71.0 ± 0.3	75.1 ± 0.8	75.8 ± 0.6	76.7 ± 1.0	78.9 ± 0.4
Soja	31 999	7 903	77 098	89.2 ± 0.8	89.5 ± 0.5	88.8 ± 0.4	87.3 ± 0.3	87.5 ± 0.3	89.5 ± 0.4	89.8 ± 0.4	91.0 ± 0.6	92.1 ± 0.4
Tournesol	29 317	7 437	73 492	87.7 ± 0.9	87.7 ± 0.9	87.1 ± 0.6	84.4 ± 0.2	84.6 ± 0.1	88.1 ± 0.3	88.4 ± 0.5	89.7 ± 0.5	90.3 ± 0.6
Maïs	32 000	8 000	80 000	90.2 ± 3.7	91.5 ± 0.3	90.8 ± 0.3	89.4 ± 0.1	89.5 ± 0.2	91.3 ± 0.2	91.0 ± 0.3	93.3 ± 0.1	93.2 ± 0.3
Riz	8 000	2 000	20 000	98.3 ± 0.1	98.3 ± 0.3	98.2 ± 0.5	98.4 ± 0.1	98.5 ± 0.1	97.8 ± 0.1	97.8 ± 0.3	98.1 ± 0.2	98.3 ± 0.2
Tubercules / Racines	28 199	6 988	62 865	83.2 ± 0.6	82.7 ± 0.8	82.8 ± 0.5	79.1 ± 0.4	79.6 ± 0.4	83.1 ± 1.0	83.8 ± 0.7	86.8 ± 0.8	87.9 ± 0.7
Prairies	32 000	8 000	80 000	71.8 ± 0.6	72.8 ± 0.8	73.0 ± 0.4	70.2 ± 0.2	70.5 ± 0.2	72.2 ± 0.4	72.8 ± 0.4	73.5 ± 0.4	75.0 ± 0.4
Vergers	25 256	6 317	57 273	73.5 ± 1.6	76.0 ± 1.2	76.9 ± 0.6	72.9 ± 0.2	74.2 ± 0.2	74.6 ± 0.5	76.1 ± 0.6	77.9 ± 0.6	80.2 ± 0.6
Vignes	28 673	7 195	64 889	86.2 ± 1.1	87.8 ± 0.7	87.7 ± 0.5	81.7 ± 0.2	82.9 ± 0.2	87.8 ± 0.3	88.4 ± 0.4	87.7 ± 0.5	88.6 ± 0.3
Forêts de feuillus	32 000	8 000	80 000	81.8 ± 0.8	82.2 ± 1.8	82.6 ± 1.9	81.0 ± 0.2	81.5 ± 0.2	83.3 ± 0.4	84.0 ± 0.3	83.4 ± 0.3	84.6 ± 0.5
Forêts de conifères	30 599	7 649	75 317	81.3 ± 0.5	81.8 ± 1.2	81.9 ± 1.5	81.4 ± 0.2	82.4 ± 0.2	82.2 ± 0.3	82.9 ± 0.3	83.0 ± 0.3	84.8 ± 0.3
Pelouses et pâturages nat.	28 000	7 000	70 000	45.5 ± 2.6	52.0 ± 1.8	53.5 ± 2.1	47.5 ± 1.4	51.1 ± 1.1	47.7 ± 1.4	52.3 ± 1.2	45.6 ± 1.4	56.6 ± 0.9
Landes ligneuses	31 983	7 926	76 189	48.3 ± 4.6	49.2 ± 2.8	51.5 ± 3.0	52.9 ± 0.4	55.0 ± 0.5	51.7 ± 0.8	53.3 ± 0.6	52.9 ± 1.0	59.6 ± 0.6
Surfaces minérales nat.	23 438	5 769	53 140	73.3 ± 1.0	74.2 ± 1.4	74.5 ± 1.5	73.6 ± 0.7	75.0 ± 0.9	73.7 ± 0.7	75.0 ± 0.9	74.7 ± 1.0	79.7 ± 0.7
Plages et dunes	27 991	6 680	59 097	77.7 ± 0.9	81.8 ± 1.0	82.4 ± 1.3	77.1 ± 0.6	80.0 ± 0.7	80.1 ± 0.6	82.3 ± 0.4	79.8 ± 0.5	85.2 ± 0.6
Glaciers et neiges éter.	7 716	1 819	14 383	88.6 ± 1.3	89.9 ± 1.0	89.2 ± 1.3	90.0 ± 1.6	90.4 ± 1.6	88.6 ± 0.9	88.3 ± 0.8	82.1 ± 3.3	83.7 ± 2.6
Eau	32 000	8 000	80 000	95.2 ± 0.3	95.9 ± 0.1	96.0 ± 0.2	95.2 ± 0.1	95.4 ± 0.1	94.7 ± 0.3	95.3 ± 0.2	95.3 ± 0.2	96.4 ± 0.1
Total	645 935	160 940	1 551 904									
			précision globale	76.6 ± 0.6	77.8 ± 0.8	78.0 ± 0.4	75.3 ± 0.1	76.3 ± 0.1	77.7 ± 0.1	78.6 ± 0.1	78.5 ± 0.2	81.0 ± 0.2
			f-score moyen	77.5 ± 0.6	78.7 ± 0.7	78.9 ± 0.4	76.4 ± 0.2	77.4 ± 0.2	78.6 ± 0.1	79.5 ± 0.1	79.0 ± 0.3	81.6 ± 0.3

TABLE 2 – Paramètres d'optimisation de *GP*, *MLP* et *LTAE*

	GP	**MLP**	**LTAE**
Nombre d'epochs E	100	300	100
Taille de batch B	1024	1000	1000
Learning rate η	1×10^{-3}	1×10^{-5}	1×10^{-5}

duit des noyaux semble être plus performant, les résultats obtenus sont en moyenne légèrement supérieurs et avec une variabilité plus faible que ceux obtenus avec la somme.

4 Conclusion

Ce papier présente des résultats de classification par GP sur des SITS massives, une des problématiques qui était identifiée dans [6]. Les résultats sont inférieurs en terme de précision par rapport à des modèles d'apprentissage profond structurés. Nous avons pour l'instant pris en compte que la structure spatiale de la donnée dans l'opérateur de covariance. La suite des travaux vise à mettre en œuvre l'extraction de caractéristiques afin d'améliorer d'avantage la prise en compte de la structure spectro-temporelle.

Références

[1] C. Persello, J. D. Wegner, R. Hansch, D. Tuia, P. Ghamisi, M. Koeva, and G. Camps-Valls, "Deep Learning and Earth Observation to Support the Sustainable Development Goals : Current Approaches, Open Challenges, and Future Opportunities," *IEEE Geoscience and Remote Sensing Magazine*, pp. 2–30, 2022.

[2] 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.

[3] T. Hofmann, B. Schölkopf, and A. J. Smola, "Kernel methods in machine learning," *The Annals of Statistics*, vol. 36, no. 3, pp. 1171 – 1220, 2008.

[4] X. X. Zhu, D. Tuia, L. Mou, G.-S. Xia, L. Zhang, F. Xu, and F. Fraundorfer, "Deep Learning in Remote Sensing : A Comprehensive Review and List of Resources," *IEEE Geoscience and Remote Sensing Magazine*, vol. 5, no. 4, pp. 8–36, 2017.

[5] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[6] 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, 2016.

[7] 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, 2020.

[8] J. Hensman, A. G. de G. Matthews, and Z. Ghahramani, "Scalable variational gaussian process classification.," in *AISTATS* (G. Lebanon and S. V. N. Vishwanathan, eds.), vol. 38 of *JMLR Workshop and Conference Proceedings*, JMLR.org, 2015.

[9] A. G. Wilson, Z. Hu, R. R. Salakhutdinov, and E. P. Xing, "Stochastic variational deep kernel learning," in *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[10] M. A. Álvarez, L. Rosasco, and N. D. Lawrence, "Kernels for vector-valued functions : A review," *Found. Trends Mach. Learn.*, vol. 4, no. 3, 2012.

[11] N. Durrande, D. Ginsbourger, and O. Roustant, "Additive Kernels for Gaussian Process Modeling," Jan. 2010. preprint submitted to CSDA.

[12] M. Titsias, "Variational Learning of Inducing Variables in Sparse Gaussian Processes," in *Proceedings of Machine Learning Research*, vol. 5, PMLR, 2009.

[13] G. Camps-Valls, L. Gomez-Chova, J. Munoz-Mari, J. 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, pp. 1822–1835, June 2008.

[14] J. R. Gardner, G. Pleiss, D. Bindel, K. Q. Weinberger, and A. G. Wilson, "Gpytorch : Blackbox matrix-matrix gaussian process inference with gpu acceleration," in *Advances in Neural Information Processing Systems*, 2018.

[15] 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.

[16] V. Sainte Fare Garnot and L. Landrieu, "Lightweight temporal self-Attention for classifying satellite images time series," in *Workshop on Advanced Analytics and Learning on Temporal Data*, AALTD, Sept. 2020.

[17] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch : An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems 32*, pp. 8024–8035, Curran Associates, Inc., 2019.