MULTIOUTPUT AUTOMATIC EMULATOR FOR RADIATIVE TRANSFER MODELS

Daniel Heestermans Svendsen, Luca Martino, Jorge Vicent, Gustau Camps-Valls

Image Processing Laboratory (IPL), Universitat de València, Spain

ABSTRACT

This paper introduces a methodology to construct emulators of costly radiative transfer models (RTMs). The proposed methodology is sequential and adaptive, and it is based on the notion of acquisition functions in Bayesian optimization. Here, instead of optimizing the unknown underlying RTM function, one aims to achieve accurate approximations. The Automatic Multi-Output Gaussian Process Emulator (AMOGAPE) methodology combines the interpolation capabilities of Gaussian processes (GPs) with the accurate design of an acquisition function that favors sampling in low density regions and flatness of the interpolation function. We illustrate the promising capabilities of the method for the construction of an emulator for a standard leaf-canopy RTM.

***Index Terms*—** Radiative transfer model, Gaussian process, emulation, self-learning, look-up table, interpolation, PROSAIL

1. INTRODUCTION

Physically-based radiative transfer models (RTMs) have contributed fundamentally in understanding the radiation processes occurring on the Earth's surface and their interactions with water, vegetation and atmosphere [1]. RTMs are physically-based computer models that describe scattering, absorption and emission processes [2]. They are useful in a wide range of applications including (i) developing inversion models to accurately retrieve atmospheric and vegetation properties from remotely sensed data (see [3] for a review), (ii) sensitivity analysis, and (iii) to generate artificial scenes as would be observed by a sensor [4].

Continuous improvement in the accuracy of RTMs have diversified them from simple turbid medium models towards advanced ray tracing models that allow for explicit 3D representations of complex scenes. Consequently, when it comes to selecting an RTM for applications that demand many simulations, the current pragmatic approach is to search for a good balance between acceptable accuracy and computational complexity [3].

RTMs are typically used in remote sensing applications by the generation of look-up tables (LUTs) [3]. LUTs are prestored RTM input-output data pairs. At the retrieval phase, one then seeks through the LUT by means of interpolation techniques [5]. However, such techniques are very computationally and memory demanding, especially when facing high dimensional problems. Emulation of costly codes is an alternative to this type of approach. The core idea of emulation is approximating the original deterministic model by a surrogate statistical learning model, also referred to as a meta-model, or *emulator* [6–8]. When an accurate emulator has been developed based on a limited set of simulations, it can then approximate the original RTM at a tiny fraction of the original speed and this be readily applied in tedious processing routines [9]. Essentially, an emulator functions as an interpolation method, but based on statistical learning principles.

In this work, we advance in the construction of *optimal emulators*. Optimality is here defined in terms of both aproximation error and compactness of the generated LUT. We are interested in addressing the problem of optimal selection of the points to be included in the LUT, and in turn to optimize the corresponding emulator. This problem has received attention from different sub-fields of statistical signal processing and machine learning: from experimental optimal design [10] of interpolators of arbitrary functions f, to optimal nonuniform sampling, quantization and interpolation of continuous functions [11], Bayesian Optimization (BO) [12], and recently from active learning [13]. Our proposal for automatic emulation and LUT construction is rooted in the field of BO, more specifically in the concept of the *acquisition function* [12]. Unlike in BO, however, our goal is not the optimization of the unknown underlying function f but its accurate *approximation* $\hat{f}$. We extend the preliminary work in [8] to a multioutput framework. Given a set of initial points, the emulator is built automatically with the addition of new points maximizing the acquisition function at each iteration. The design of the acquisition function is crucial. To this end we use Gaussian processes [7, 14] for the modeling and inference. GPs provide not only state-of-the-art approximation errors, but also mathematical tractability. The use of GPs allows us to design acquisition functions involving analytical expressions. Our method incorporates two terms, accounting for geometric information of the unknown function f, and diversity information of the included nodes. Areas with either high variability or uncertainty of f thus require the addition of more points.

2. AUTOMATIC EMULATION

In this section, we describe the generic automatic emulation (AE) procedure of an unknown complex system $\mathbf{f}(\mathbf{x})$, e.g., an expensive RTM model. We start by fixing the notation and then presenting the processing scheme. Let us consider a

The research was funded by the European Research Council (ERC) under the ERC-CoG-2014 SEDAL project (grant agreement 647423), and the Spanish Ministry of Economy and Competitiveness (MINECO) through the project TIN2015-64210-R.

D-dimensional bounded input space $\mathcal{X}$, i.e., $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^D$. We consider a complex system with P outputs, $\mathbf{f}(\mathbf{x}) : \mathcal{X} \mapsto \mathbb{R}^{P\times 1}$, i.e., Furthermore, let $t \in \mathbb{N}^+$ denote the index of the AE algorithm, and m_t be the number of datapoints used by the algorithm at iteration t. Then, corresponding to an input matrix $\mathbf{X}_t = [\mathbf{x}_1, \cdots, \mathbf{x}_{m_t}]$ of dimension $D \times m_t$, we have a $P \times m_t$ matrix of outputs,

$$\mathbf{Y}_t = [\mathbf{y}_1, \ldots, \mathbf{y}_{m_t}] \tag{1}$$

where $\mathbf{y}_k = [y_{1,k}, \ldots, y_{P,k}]^\intercal = \mathbf{f}(\mathbf{x}_k)$ with $k = 1, \ldots, m_t$.

At each iteration t, given $\mathbf{X}_t$ and $\mathbf{Y}_t$, the AE method constructs an interpolator $\widehat{\mathbf{f}}_t(\mathbf{x})$. Then, an acquisition function $A_t(\mathbf{x}) : \mathbb{R}^D \to \mathbb{R}$ is built according to some suitable criteria. This is followed by an optimization step for obtaining the next input $\mathbf{x}_{m_t+1}$, more specifically,

$$\mathbf{x}_{m_t+1} = \arg\max_{\mathbf{x}\in\mathcal{X}} A_t(\mathbf{x}). \tag{2}$$

Thus, we update $\mathbf{X}_{t+1} = [\mathbf{X}_t, \mathbf{x}_{m_t+1}]$, $\mathbf{Y}_{t+1} = [\mathbf{Y}_t, \mathbf{y}_{t+1} = \mathbf{f}(\mathbf{x}_{t+1})]$ adding a new node, set $m_t \leftarrow m_t + 1$ and $t \leftarrow t+1$. The procedure is repeated until a stopping condition is met such as a certain maximum number of points is included or a least precision error ϵ is achieved, $\|\widehat{\mathbf{f}}_t(\mathbf{x}) - \widehat{\mathbf{f}}_{t-1}(\mathbf{x})\| \leq \epsilon$. Figure 1 shows a graphical representation of a generic multi-output AE procedure. In the next sections, we present a specific implementation involving Gaussian Process interpolators [14].

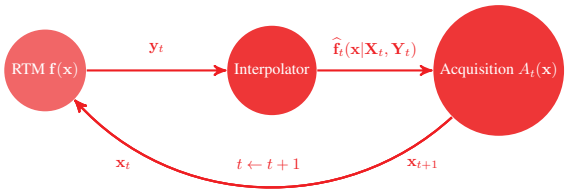

Fig. 1. Scheme of an automatic emulator.

3. THE GAUSSIAN PROCESS INTERPOLATOR

An automatic emulator is completely defined by the choice of the interpolation method for building $\widehat{f}(\mathbf{x})$ and the construction procedure for the acquisition function $A_t(\mathbf{x})$. In this work, we consider a Gaussian Process (GP) interpolator [14], which has been successfully used in remote sensing [7].

For simplicity, first let us consider the GP solution for the scalar output case, i.e., $P = 1$. Hence, in this case the vectorial function $\mathbf{y} = \mathbf{f}(\mathbf{x})$ is a simple standard function $y = f(\mathbf{x})$, and the matrix

$$\mathbf{Y}_t = [y_{1,1}, \ldots, y_{1,m_t}],$$

becomes a $1 \times m_t$ vector. GPs give a full Gaussian predictive density with predictive mean μ_{GP} and variance σ^2_{GP} for an input point $\mathbf{x}$. The predictive mean of the interpolating function for a new point $\mathbf{x}$ is given by

$$\widehat{f}_t(\mathbf{x}) = \mu_{\text{GP}}(\mathbf{x}) = \mathbf{k}_x^\intercal \mathbf{K}^{-1} \mathbf{Y}_t^\intercal, \tag{3}$$

where we defined a kernel function $k(\mathbf{x}, \mathbf{z}) : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$, the corresponding kernel matrix $\mathbf{K}_{ij} := k(\mathbf{x}_i, \mathbf{x}_j)$ of dimension $m_t \times m_t$ containing all kernel entries, and the kernel vector $\mathbf{k}_x = [k(\mathbf{x}, \mathbf{x}_1), \ldots, k(\mathbf{x}, \mathbf{x}_{m_t})]^\intercal$ of dimension $m_t \times 1$. The interpolation for $\mathbf{x}$ can be simply expressed as a linear combination of $\widehat{f}_t(\mathbf{x}) = \mathbf{k}_x^\intercal \boldsymbol{\alpha} = \sum_{i=1}^{m_t} \alpha_i k(\mathbf{x}, \mathbf{x}_i)$, where the weights $\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_{m_t}]^\intercal$ are $\boldsymbol{\alpha} = \mathbf{K}^{-1}\mathbf{Y}_t^\intercal$. The GP formulation provides also an expression for the predictive variance

$$\sigma^2_{\text{GP}}(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}) - \mathbf{k}_x^\intercal \mathbf{K}^{-1} \mathbf{k}_x. \tag{4}$$

In this work, we consider the exponentiated quadratic kernel function,

$$k(\mathbf{x}, \mathbf{z}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{z}\|^2}{2\delta^2}\right), \tag{5}$$

where $\|\cdot\|$ is the ℓ_2-norm, and $\delta > 0$ is a positive scalar hyper-parameter. In this work, the scalar hyper-parameter is tuned by maximizing the marginal likelihood [14]. Note that the norm of the gradient of the interpolating function $\widehat{f}_t$ w.r.t. the input data $\mathbf{x}$ can be easily computed,

$$\texttt{Gr}(\mathbf{x}) = \left\|\nabla_x \widehat{f}_t(\mathbf{x}|\mathbf{X}_t, \mathbf{y}_t)\right\| = \left\|\sum_{i=1}^{m_t} \alpha_i \nabla_x k(\mathbf{x}, \mathbf{x}_i)\right\|. \tag{6}$$

The gradient vector of $k(\mathbf{x}, \mathbf{x}_i)$ in Eq. (5) with $\mathbf{x} = [x_1, \ldots, x_D]^\intercal$ and $\mathbf{x}_i = [x_{1,i}, \ldots, x_{D,i}]^\intercal$, is

$$\nabla_x k(\mathbf{x}, \mathbf{x}_i) = -\frac{k(\mathbf{x}, \mathbf{x}_i)}{\delta^2}[(x_1 - x_{1,i}), \ldots, (x_D - x_{D,i})]^\intercal. \tag{7}$$

3.1. Multi-output GP interpolator

Several Multi-output GP schemes have been proposed [15]. For the sake of simplicity, in this work we consider a simple approach. Let us define the p-th row of the matrix $\mathbf{Y}_t$ as

$$\widetilde{\mathbf{y}}_{p,t} = [y_{p,1}, \ldots, y_{p,m_t}],$$

with $p = 1, \ldots, P$ as shown in Eq. (1). We apply one GP interpolator for each output, i.e.,

$$\widehat{\mathbf{f}}_t(\mathbf{x}) = \begin{cases} \widehat{f}_{1,t}(\mathbf{x}) = \mathbf{k}_{x,1}^\intercal \mathbf{K}_1^{-1} \widetilde{\mathbf{y}}_{1,t}^\intercal \\ \quad\vdots \\ \widehat{f}_{P,t}(\mathbf{x}) = \mathbf{k}_{x,P}^\intercal \mathbf{K}_P^{-1} \widetilde{\mathbf{y}}_{P,t}^\intercal \end{cases}, \tag{8}$$

where the subindex in the kernel vector $\mathbf{k}_{x,p}$ and the kernel matrix $\mathbf{K}_p$ denotes the dependence to a different hyper-parameter δ_p (we learn one for each output). Hence, for each output, we have a different variance

$$\sigma^2_p(\mathbf{x}) = k_p(\mathbf{x}, \mathbf{x}) - \mathbf{k}_{x,p}^\intercal \mathbf{K}_p^{-1} \mathbf{k}_{x,p}. \tag{9}$$

Naturally, we have a different norm of the gradient of the interpolating function, $\texttt{Gr}_p(\mathbf{x})$, one for each output as well.

4. THE ACQUISITION FUNCTION

Let us start describing the general properties that a generic acquisition function $A_t(\mathbf{x})$ should satisfy [8]. We consider an acquisition function defined as the product of two functions, a *geometry term* $G_t(\mathbf{x})$ and a *diversity term* $D_t(\mathbf{x})$, i.e.,

$$A_t(\mathbf{x}) \quad = \quad [G_t(\mathbf{x})]^{\beta_t} D_t(\mathbf{x}), \quad \beta_t \in [0,1], \qquad (10)$$

where $G_t(\mathbf{x}) : \mathcal{X} \mapsto \mathbb{R}$, $D_t(\mathbf{x}) : \mathcal{X} \mapsto \mathbb{R}$ and hence $A_t(\mathbf{x}) : \mathcal{X} \mapsto \mathbb{R}$. Moreover, β_t is an increasing function with respect to t, with $\lim_{t\to\infty} \beta_t = 1$ (or $\beta_t = 1$ for $t > t'$). The first function $G_t(\mathbf{x})$ represents some suitable geometrical information of the hidden function f. The second function $D_t(\mathbf{x})$ depends on the distribution of the points in the current vector $\mathbf{X}_t$. We desire that $D_t(\mathbf{x})$ is approximately zero close to the nodes ($D_t(\mathbf{x}_i) = 0$, for $i = 1, \ldots, m_t$ and $\forall t \in \mathbb{N}$) and takes higher values around empty areas within $\mathcal{X}$. Since f is unknown, the function $G_t(\mathbf{x})$ can be only derived from information acquired in advance or by considering the approximation $\widehat{\mathbf{f}}$. The approximation has usually not achieved a good fit in the first iterations of the algorithm, so that the information provided by $G_t(\mathbf{x})$ should be disregarded or "tempered" (as in tempering strategies for optimization [16]) in these first iterations. This is the reason of using the tempering value β_t. If $\beta_t = 0$, we disregard $G_t(\mathbf{x})$ and $A_t(\mathbf{x}) = D_t(\mathbf{x})$ whereas, if $\beta_t = 1$, we have $A_t(\mathbf{x}) = G_t(\mathbf{x})D_t(\mathbf{x})$.

Note that $\sigma_p^2(\mathbf{x}_i) = 0$ for all $i = 1, \ldots, m_t$ and for all p, and each $\sigma_p^2(\mathbf{x})$ depends on the distance among the support points $\mathbf{x}_t$, and the chosen kernel function k and associated hyper-parameter δ_p. For this reason, it is reasonable to consider as diversity term the following function

$$D_t(\mathbf{x}) := \prod_{p=1}^{P} \sigma_p^2(\mathbf{x}). \qquad (11)$$

We wish to use the geometric information term to sample where the norm of the gradient is high and thus define

$$G_t(\mathbf{x}) := \prod_{p=1}^{P} \mathtt{Gr}_p(\mathbf{x}). \qquad (12)$$

The intuition behind this choice is that wavy regions of $\mathbf{f}$ (estimated by $\widehat{\mathbf{f}}_t$) require more support points than flat regions. Then, the acquisition function is defined $A_t(\mathbf{x}) = [G_t(\mathbf{x})]^{\beta_t} D_t(\mathbf{x})$ as in Eq. (10). For the parameter β_t, we suggest $\beta_t = 1 - \exp(-\gamma t)$, where $\gamma \geq 0$ is a positive scalar, established by the user.

5. EXPERIMENTAL RESULTS

We tested our method for the emulation of the standard leaf-canopy PROSAIL model. PROSAIL is the most widely used RTM in the last twenty years in remote sensing studies [17]. PROSAIL models canopy reflectance using the turbid medium assumption (i.e., modelling the canopy as a turbid medium for which leaves are randomly distributed), which is particularly well suited for homogeneous canopies. PROSAIL simulates leaf reflectance from 400 to 2500 nm with a 1 nm spectral resolution as a function of biochemistry and structure of the canopy, its leaves, the background soil reflectance and the sun-sensor geometry. Leaf optical properties are given by the mesophyll structural parameter (N) and leaf chlorophyll (Chl), dry matter (Cm), water (Cw), carotenoid (Car) and brown pigment (Cbr) contents. At canopy level PROSAIL is characterized by leaf area index (LAI), the average leaf angle inclination (ALA) and the hotspot parameter ($Hotspot$). The system geometry is described by the solar zenith angle (θ_s), view zenith angle (θ_ν), and the relative azimuth angle between both angles ($\Delta\Theta$). In this experiment, for ease of computation, we chose as free parameters the most important variable at leaf and canopy-level respectively, namely Chl and LAI, and keep the rest fixed. Table 1 shows the values for the remaining parameters which are set for simulation of wheat. This results in an input space of dimension two, where we restrict the variables $Chl \in [0; 80]\mu$g/cm^2 and $LAI \in [0; 8]$. We scale down the output dimension from 2101 to 20 by principal component analysis (PCA). This results in a function $\mathbf{f}(\mathbf{x})$, where $\mathbf{x} = [Chl, LAI]$, mapping from an input space of dimension $D = 2$ to the output space of dimension $P = 20$.

Table 1. Values of physical parameters used for simulating with the PROSAIL model, corresponding to wheat.

N	Cm	Cw	Car	Cbr
1.5	0.01 µg/cm^2	0.01 µg/cm^2	8 g/cm^2	0
ALA	$Hotspot$	θ_s	θ_ν	$\Delta\Theta$
Spherical	0.01	30°	10°	0

We compare different approaches to sampling the datapoints that lead to a good emulator. In order to test the accuracy of the different schemes, we evaluated $\mathbf{f}(\mathbf{x})$ at all the possible 4900 combinations of 70 values of Chl and 70 values of LAI on a grid. This fine grid represents the ground-truth in the experiment. It is on this dataset that we perform PCA in order to obtain the 20 principal components used for dimensionality reduction.

We test **(a)** a random approach choosing points uniformly at each iteration, **(b)** the Latin Hypercube sampling approach (see, e.g., [6]) and **(c)** AMOGAPE, using simulated annealing for optimization of A_t. We start with 50 points generated by LHS sampling for all the methods. For each added point we compute the test root mean square error (RMSE) of $\widehat{\mathbf{f}}$ applied to the grid in the original (reflectance domain). In other words the predicted 20-dimensional vector is projected back into 2101-dimensional reflectance space and compared to ground truth. The multioutput RMSE for the $N_t = 4900$ test points with output dimension $P_{refl.} = 2101$ is computed as follows,

$$\text{RMSE} = \sqrt{\frac{1}{N_t}\sum_{i=1}^{N_t}\frac{1}{P_{refl.}}\sum_{p=1}^{P_{refl.}}(y_{p,i} - \hat{y}_{p,i})^2}. \qquad (13)$$

The results, averaged over 50 runs, are shown in Fig. 2. The LHS sampling method exhibits a lot of variance as it chooses

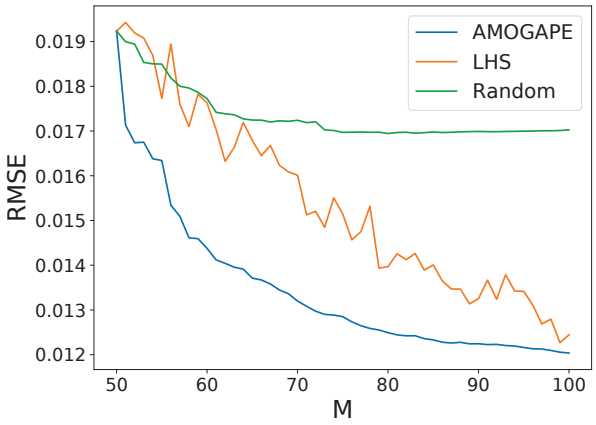

Fig. 2. RMSE on test grid computed for emulators using different sampling methods. Each method is initialized with 50 points sampled with the LHS scheme, upon which 50 more are sampled.

a completely new set of points for each iteration, as opposed to the other approaches which sequentially add points to existing datasets. We see how the AMOGAPE rather quickly identifies the points it needs to build a stronger emulator, while the LHS decreases the error and in a slower manner. The completely random sampling method, not being designed to maximize information gained, does not manage to reach the same level of error as the other methods. This gap is expected to widen as the input dimensionality grows.

6. CONCLUSIONS

We introduced an automatic methodology for constructing emulators for costly RTMs. The methodology iteratively incorporates new sample points that meet both diversity and geometry criteria, thus sampling in low-density and more 'complex' regions. This is accomplished by building an acquisition function that takes into account predictive variance and the norm of the gradient. The combination of the geometric and diversity sampling criteria was possible because the gradient of the GP predictive mean function yields a closed-form expression. We illustrated the good capabilities of the method through emulation of leaf-canopy PROSAIL RTM. Future work will tackle more challenging cases of emulation involving more parameters and the MODTRAN model.

7. REFERENCES

[1] S. Jacquemoud, W. Verhoef, F. Baret, C. Bacour, P.J. Zarco-Tejada, G.P. Asner, C. François, and S.L. Ustin, "PROSPECT + SAIL models: A review of use for vegetation characterization," *Remote Sensing of Environment*, vol. 113, no. SUPPL. 1, pp. S56–S66, 2009.

[2] M. Deiveegan, C. Balaji, and S.P. Venkateshan, "A polarized microwave radiative transfer model for passive remote sensing," *Atmospheric Research*, vol. 88, no. 3-4, pp. 277–293, 2008.

[3] J. Verrelst, G. Camps-Valls, J. Muñoz Marí, J.P. Rivera, F. Veroustraete, J.G.P.W. Clevers, and J. Moreno, "Optical remote sensing and the retrieval of terrestrial vegetation bio-geophysical properties - a review," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 108, pp. 273–290, 2015.

[4] W. Verhoef and H. Bach, "Simulation of Sentinel-3 images by four-stream surface-atmosphere radiative transfer modeling in the optical and thermal domains," *Remote Sensing of Environment*, vol. 120, pp. 197–207, 2012.

[5] M. Abramowitz and I.A. Stegun, "Handbook of mathematical functions," in *Applied Mathematics Series, Volume 55.*, chapter 25.2. National Bureau of Standards, Washington, USA, 1964.

[6] D. Busby, "Hierarchical adaptive experimental design for Gaussian process emulators," *Reliability Engineering and System Safety*, vol. 94, pp. 1183–1193, 2009.

[7] Gustau Camps-Valls, Jochem Verrelst, Jordi Muñoz-Marí, Valero Laparra, Fernando Mateo-Jiménez, and José Gómez-Dans, "A survey on Gaussian processes for earth observation data analysis," *IEEE Geoscience and Remote Sensing Magazine*, vol. 4, no. 2, 2016.

[8] L. Martino, J. Vicent, and G. Camps-Valls, "Automatic emulator and optimized look-up table generation for radiative transfer models," *Proc. of IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 1–4, 2017.

[9] J. Verrelst, J.P. Rivera Caicedo, J. Muñoz Marí, G. Camps-Valls, and J. Moreno, "SCOPE-based emulators for fast generation of synthetic canopy reflectance and sun-induced fluorescence Spectra," *Remote Sensing*, vol. 9, no. 9, 2017.

[10] G. da Silva Ferreira and D. Gamerman, "Optimal design in geostatistics under preferential sampling," *Bayesian Analysis*, vol. 10, no. 3, pp. 711–735, 2015.

[11] F. Marvasti, "Nonuniform sampling: Theory and Practice," *Kluwer Academic Publishers*, 2001.

[12] M. U. Gutmann and J. Corander, "Bayesian optimization for likelihood-free inference of simulator-based statistical models," *Journal of Machine Learning Research*, vol. 16, pp. 4256–4302, 2015.

[13] D. Cohn, Z. Ghahramani, and M.I. Jordan, "Active learning with statistical models," *Journal of Artificial Intelligence Research*, vol. 4, pp. 129–145, 1996.

[14] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, The MIT Press, New York, 2006.

[15] Mauricio A Alvarez, Lorenzo Rosasco, Neil D Lawrence, et al., "Kernels for vector-valued functions: A review," *Foundations and Trends® in Machine Learning*, vol. 4, no. 3, pp. 195–266, 2012.

[16] L. Martino, V. Elvira, D. Luengo, J. Corander, and F. Louzada, "Orthogonal parallel MCMC methods for sampling and optimization," *Digital Signal Processing*, vol. 58, pp. 64–84, 2016.

[17] Stéphane Jacquemoud, Wout Verhoef, Frédéric Baret, Cédric Bacour, Pablo J Zarco-Tejada, Gregory P Asner, Christophe François, and Susan L Ustin, "Prospect+ sail models: A review of use for vegetation characterization," *Remote sensing of environment*, vol. 113, pp. S56–S66, 2009.