ADAPTIVE SEQUENTIAL INTERPOLATOR USING ACTIVE LEARNING FOR EFFICIENT EMULATION OF COMPLEX SYSTEMS

Luca Martino$^{\top}$*, Daniel Heestermans Svendsen***, Jorge Vicent**,**, Gustau Camps-Valls**

* Image Processing Laboratory (IPL), Universitat de València, Spain.
* Magellium Company in Geoinformation and Image Processing, Toulouse, France.
$^{\top}$ Dep. of Signal Processing, Universidad Rey Juan Carlos, Madrid, Spain.

ABSTRACT

Many fields of science and engineering require the use of complex and computationally expensive models to understand the involved processes in the system of interest. Nevertheless, due to the high cost involved, the required study becomes a cumbersome process. This paper introduces an interpolation procedure which belongs to the family of active learning algorithms, in order to construct cheap surrogate models of such costly complex systems. The proposed technique is sequential and adaptive, and is based on the optimization of a suitable acquisition function. We illustrate its efficiency in a toy example and for the construction of an emulator of an atmosphere modeling system.

***Index Terms*—** Adaptive interpolation, active learning, Bayesian optimization, experimental design.

1. INTRODUCTION

Approximation theory is vast branch of mathematics embracing fields such as signal processing, statistics and machine learning. Approximate models are often used in order to summarize the behavior of more complex systems with fast and cheap routines. The so-called radiative transfer models (RTMs) are clear examples of costly complex systems widely in geoscience [1, 2]. Since RTMs are computationally expensive and very often impractical for their execution on a pixel-per-pixel basis [3], large multi-dimensional look-up tables (LUTs) are precomputed for their later interpolation [4, 5]. More generally, several emulation models have been proposed in the literature [6–10].

In this work, we address the problem of optimal selection of the points to be included in the LUT or, more generally, we propose an adaptive sequential interpolation (ASI) procedure. This field has received attention from different scientific areas. For instance, the considered topic is related to several problems such as: (a) optimal sensor placement [11], (b) optimal nonuniform sampling [12], (c) adaptive quantization [13], (d) adaptive gridding [6] and (e) adaptive mesh refinement [14]. Moreover, the problem has been also cast as optimal experimental design [15, 16] for regression of arbitrary functions f. It deserves to be mentioned that the use of random filling strategies for the sequential construction of LUT (such as Latin Hypercube and Sobol sampling) seem to be the favorites, at least in the remote sensing community.

The proposed ASI solution belongs to the class of active learning algorithms [17, 18] where we use the notion of an *acquisition function* (AF), often employed in Bayesian optimization (BO) methods [19]. More specifically, the ASI technique provides an approximation $\widehat{f}(\mathbf{x})$ of a complicated, costly function $f(\mathbf{x})$ by using an interpolation scheme based on a set of support points. This approximation is sequentially improved by adding new points. The position of a new point is adaptively determined by the algorithm, according to a suitable AF (which is considered as a pay-off function or an "oracle"). In the literature, different approaches have been recently proposed for efficiently solving this problem. In some of them, the evaluation of the AF is quite expensive since its construction is based on a Cross-Validation (CV) "Leave-one-out" procedure [20]. In other approaches, the construction of the AF is based on a statistical interpretation of a specific interpolation procedure and its evolution of the inversion of a square matrix whose dimension increases with the number of support points [6, 8]. This inversion can make the method unstable as the number of nodes grows or the distances among nodes decrease. Another scheme has been proposed in [21], but its performance strongly depends on the choice of the parameters used for building the corresponding AF.

In this paper, we propose a robust construction of a suitable AF which does not depend on a statistical derivation or a specific interpolation method (i.e., it can be applied whatever interpolation procedure is used), and does not require any matrix inversion. Its evaluation is fast since it is a polynomial-based AF and no CV procedures are required. Furthermore, its evaluation is also stable even when a very high number of support points has been added. The numerical experiments also show its benefit compared with other benchmark strategies.

The research was funded by the European Research Council (ERC) under the ERC-CoG-2014 SEDAL project (grant agreement 647423).

2. ADAPTIVE SEQUENTIAL INTERPOLATION

In this section, we describe the proposed adaptive sequential interpolation (ASI) scheme. We start by fixing the notation and presenting the general procedure, and then we discuss more specific details.

2.1. Generic ASI scheme

Let us consider a D-dimensional bounded input space $\mathcal{X}$, i.e., $\mathbf{x} = [x_1, ..., x_D]^\intercal \in \mathcal{X} \subset \mathbb{R}^D$. We consider a costly complex system represented by a function $f(\mathbf{x}) : \mathcal{X} \mapsto \mathbb{R}$. Let $t \in \mathbb{N}^+$ denote the index of the ASI algorithm, and m_t be the number of points $\{\mathbf{x}_k, y_k\}_{k=1}^{m_t}$ used by at the iteration t, where

$$y_k = f(\mathbf{x}_k), \tag{1}$$

where $y_k \in \mathbb{R}$ and $k = 1, \dots, m_t$. Thus, given an input matrix of nodes, $\mathbf{X}_t = [\mathbf{x}_1, \cdots, \mathbf{x}_{m_t}]$ of dimension $D \times m_t$, we have a vector of outputs, $\mathbf{y}_t = [y_1, \dots, y_{m_t}]$. At each iteration t, given the pairs of points $\{\mathbf{x}_k, y_k\}_{k=1}^{m_t}$, the ASI procedure constructs an approximating function $\widehat{f}_t(\mathbf{x})$ using an arbitrary interpolation technique, such as piecewise polynomial functions, Splines, and Gaussian Processes (GPs) [8, 20, 22].

An acquisition function $A_t(\mathbf{x}) : \mathbb{R}^D \to \mathbb{R}$ is constructed, in order to recommend in which part of the input space an additional node is required. Then, an optimization step is needed for obtaining an additional node, i.e.,

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

Thus, we update $\mathbf{X}_{t+1} = [\mathbf{X}_t, \mathbf{x}_{m_t+1}]$, $\mathbf{y}_{t+1} = [\mathbf{y}_t, y_{m_t+1} = f(\mathbf{x}_{m_t+1})]$ adding a new node, set $m_{t+1} = m_t + 1$ and $t \leftarrow t + 1$. The procedure is repeated until a stopping condition is met. Table 1 summarizes the generic ASI scheme. The maximization of $A_t(\mathbf{x})$ can be performed by different types of optimization algorithms (see, e.g., [23]).

Stopping rules. One possible stopping condition is to reach a pre-established maximum number of points M, which is determined by the available computational resources. Moreover, a more sophisticated rule is stop the algorithm when a precision error $\epsilon > 0$ is achieved, i.e., $\|f(\mathbf{x}) - \widehat{f}_t(\mathbf{x})\| \le \epsilon$. However, generally the system $\mathbf{y} = f(\mathbf{x})$ is a costly black-box mapping, linking the inputs $\mathbf{x}$ with the outputs $\mathbf{y}$. At each new input $\mathbf{x}'$, the system returns $\mathbf{y}' = f(\mathbf{x}')$ but, we cannot compute integrals and/or other analytical expressions involving $f(\mathbf{x})$. One alternative is to approximate the error $\|f(\mathbf{x}) - \widehat{f}_t(\mathbf{x})\|$ with the alternative formula $\|\widehat{f}_t(\mathbf{x}) - \widehat{f}_{t-1}(\mathbf{x})\|$ which involves only the approximating function in two consecutive iterations.

Suitable construction of an AF. In this section, we introduce the general properties that a suitable acquisition function $A_t(\mathbf{x})$ should satisfy. A complete acquisition function should be formed by the multiplication two terms, a *geometry factor* $G_t(\mathbf{x})$ and a *density factor* $D_t(\mathbf{x})$, i.e.,

$$A_t(\mathbf{x}) = [G_t(\mathbf{x})]^{\beta_t} D_t(\mathbf{x}), \quad \beta_t \in [0, 1], \tag{3}$$

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 density factor $D_t(\mathbf{x})$ depends on the distribution of the points in the current vector $\mathbf{X}_t$. More specifically, $D_t(\mathbf{x})$ will have a greater value around empty regions of the input space, whereas $D_t(\mathbf{x})$ will be virtually zero close to the nodes and more specifically, $D_t(\mathbf{x}_k) = 0$ (exactly zero at the nodes). Note that, for this reason, we have also that

$$A_t(\mathbf{x}_k) = 0, \quad \forall k = 1, \dots, m_t, \quad \forall t \in \mathbb{N}. \tag{4}$$

The geometry factor $G_t(\mathbf{x})$ represents some suitable geometrical information of the system f. Since f is generally a complex and analytical intractable function, the term $G_t(\mathbf{x})$ can be only obtained by considering the approximation $\widehat{f}$. Clearly, in this case, the approximation $\widehat{f}$ is usually not well-fitted in the first iterations of the algorithm, so that the information provided by $G_t(\mathbf{x})$ should be partially excluded in the first iterations. This is the reason of using a tempering factor $\beta_t \in [0, 1]$, which is a non-decreasing function of t. If $\beta_t = 0$, we omit $G_t(\mathbf{x})$ and $A_t(\mathbf{x}) = D_t(\mathbf{x})$ whereas, if $\beta_t = 1$, we have $A_t(\mathbf{x}) = G_t(\mathbf{x}) D_t(\mathbf{x})$. Strategies only using the density factor has been often used in the literature, generally related to GP regressors (see, e.g., [6]). Examples of acquisition function are given in Figure 1.

3. ACQUISITION FUNCTION OF ASI

In this work, we propose a novel acquisition function inspired by the Lebesgue functions, employed in the theoretical analysis of the classical polynomial interpolation [22].[1] The density term is defined as

$$D_t(\mathbf{x}) = \prod_{k=1}^{m_t} ||\mathbf{x} - \mathbf{x}_k||_1 = \prod_{k=1}^{m_t} \left[\sum_{d=1}^{D} |x_d - x_{d,k}| \right], \tag{6}$$

where $||\cdot||_1$ represents the L_1 distance and $\mathbf{x}_k = [x_{1,k}, ..., x_{D,k}]^\intercal$ are the current interpolation nodes. Note that $D_t(\mathbf{x}_k) = 0$ for all $k = 1, ..., m_t$. The geometry term, taking into account the variations of the outputs, is defined as

$$G_t(\mathbf{x}) = \prod_{k=1}^{m_t} |y_k - f(\mathbf{x})|. \tag{7}$$

[1] In this work, we consider piecewise interpolations, such as piecewise constant or linear solutions, Splines and GPs interpolators. The location of the Chebyshev's nodes is optimal only when the interpolation is performed with a unique polynomial function of the same order of number of nodes (then, with several oscillations and worse performance than piecewise interpolations).

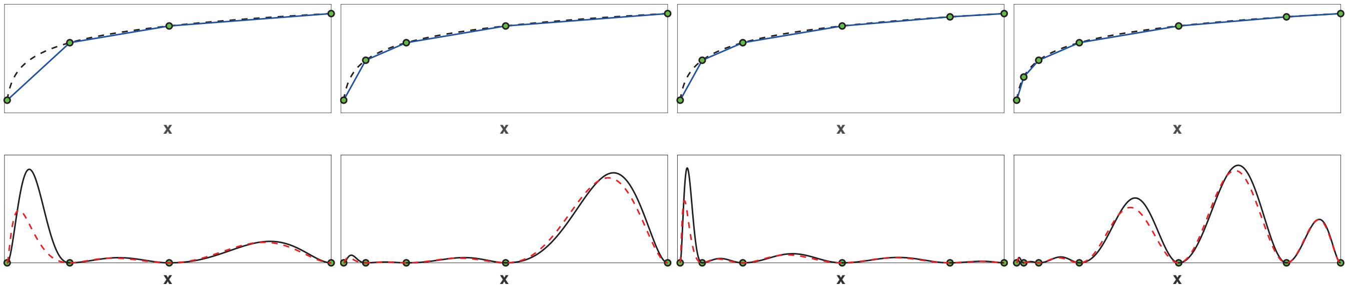

Fig. 1. **(Top-row)** An example of function $f(x)$ (dashed line) and its approximation $\widehat{f}_t(x)$ built with a piecewise linear interpolation (solid line), at different iterations. **(Bottom-row)** The acquisition function $A_t(x)$ of the ASI method with $\beta_t = 1$, considering the ideal factor $G_t(x)$ given in Eq. (7) (dashed line), or alternatively the feasible (approximate) factor $G_t(x)$ given in Eq. (8) (solid line), at different consecutive iterations of the ASI algorithm.

Table 1. Adaptive Sequential Interpolation (ASI) algorithm.

```
1. Set $t = 0$, select initial points $\mathbf{X}_0 = [\mathbf{x}_1, \cdots, \mathbf{x}_{m_0}]$,
   and $\mathbf{y}_0 = [y_1, \dots, y_{m_0}]$.
2. While the stopping condition is not satisfied:
   (a) Given $\mathbf{X}_t = [\mathbf{x}_1, \cdots, \mathbf{x}_{m_t}]$ and $\mathbf{y}_t = [y_1, \dots, y_{m_t}]$, build function $\widehat{f}_t(\mathbf{x})$.
   (b) Build the acquisition function $A_t(\mathbf{x})$ from $\widehat{f}_t$,
       and obtain the new input
           $\mathbf{x}_{m_t+1} = \arg\max_{\mathbf{x}\in\mathcal{X}} A_t(\mathbf{x}).$     (5)
   (c) Obtain outputs $y_{m_t+1} = f(\mathbf{x}_{m_t+1})$.
   (d) Update $\mathbf{X}_{t+1} = [\mathbf{X}_t, \mathbf{x}_{m_t+1}]$, $\mathbf{y}_{t+1} = [\mathbf{y}_t, y_{m_t+1}]$.
   (e) Set $m_{t+1} = m_t + 1$ and $t \longleftarrow t + 1$.
3. Build the interpolating function $\widehat{f}_t(\mathbf{x})$.
4. Return final set of optimal nodes $\{\mathbf{x}_k, y_k\}_{k=1}^{m_t}$ as a
   Look-up Table (LUT) and the approximation $\widehat{f}_t(\mathbf{x})$.
```

Since we are considering systems $f(\mathbf{x})$ whose evaluation is costly, we use an approximation by means of the built interpolator yielding

$$G_t(\mathbf{x}) = \prod_{k=1}^{m_t} |y_k - \widehat{f}_t(\mathbf{x})|, \tag{8}$$

where we have replaced $f(\mathbf{x})$ with the current approximation $\widehat{f}_t(\mathbf{x})$. Clearly, the reliability of the geometry factor depends on the quality of the approximation $\widehat{f}_t$. The confidence over $G_t(\mathbf{x})$ is represented by the tempering value $\beta_t \in [0, 1]$ (which is a a non-decreasing function of t). Some possible choices are $\beta_t = 1 - \frac{1}{t}$ or $\beta_t = 1 - e^{-bt}$ (we recall $t \in \mathbb{N}^+$), and $b \geq 0$ should be tuned according to the specific application, the number of initial nodes, and the dimension D of the problem. However, good performance can be obtained with constant β_t, for instance $\beta_t = 0$ or $\beta_t = 1$, as shown in the numerical simulations.

The complete acquisition function of the ASI method is $A_t(\mathbf{x}) = [G_t(\mathbf{x})]^{\beta_t} D_t(\mathbf{x})$ where D_t and G_t are given in Eqs. (6)-(8). It is important to remark that even the case $A_t(\mathbf{x}) = D_t(\mathbf{x})$ (i.e., $\beta_t = 0$ for all t) can provide better results compared with a Sobol sequence and/or a Latin Hypercube (LHC) approach. See Section 4 for further details. In the case that the geometry term is employed (i.e., $\beta_t \neq 0$), a practical suggestion, useful in some scenarios, is to scale outputs and inputs in order to be in the same range of values. This is not strictly needed, but when the ranges of values is quite different, is advisable. Note that the use of a proper factor β_t is advisable also to mitigate this effect.

Acquisition function of a Multioutput ASI. In the case that we have a complex system with P outputs, $\mathbf{f}(\mathbf{x}) : \mathcal{X} \mapsto \mathbb{R}^{P\times 1}$, i.e.,

$$\mathbf{y} = [y_1, \dots, y_P]^\intercal = \mathbf{f}(\mathbf{x}), \tag{9}$$

where $\mathbf{y}$ is a column vector of dimension P. Then, at each iteration $t \in \mathbb{N}^+$, we have an input matrix $\mathbf{X}_t = [\mathbf{x}_1, \cdots, \mathbf{x}_{m_t}]$ of dimension $D \times m_t$ and $P \times m_t$ matrix of outputs,

$$\mathbf{Y}_t = [\mathbf{y}_1, \dots, \mathbf{y}_{m_t}], \tag{10}$$

where $\mathbf{y}_k = [y_{1,k}, \dots, y_{P,k}]^\intercal = \mathbf{f}(\mathbf{x}_k)$ with $k = 1, \dots, m_t$. In this scenario, the density term $D_t(\mathbf{x})$ remains unaltered, whereas the geometry term becomes

$$G_t(\mathbf{x}) = \prod_{k=1}^{m_t} ||\mathbf{y}_k - \widehat{\mathbf{f}}_t(\mathbf{x})||_1 = \prod_{k=1}^{m_t} \left[\sum_{j=1}^{P} \left| y_{j,k} - \widehat{f}_{j,t}(\mathbf{x}) \right| \right],$$

where $\widehat{\mathbf{f}}_t(\mathbf{x}) = [\widehat{f}_{1,t}(\mathbf{x}), ..., \widehat{f}_{D,t}(\mathbf{x})]^\intercal$ is obtained with a multi-output interpolation procedure. The rest of the algorithm remains unchanged.

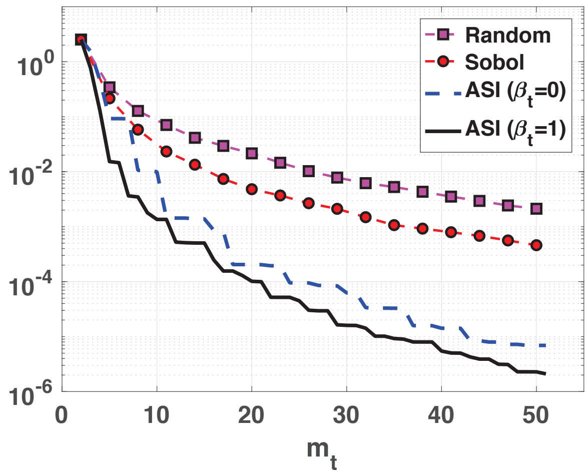

Fig. 2. The L_2 distance as function of the number of nodes m_t, obtained by different schemes, using piecewise linear interpolation in all cases.

4. SIMULATIONS

4.1. First numerical analysis

In this example, we compare the approximation $\widehat{f}_t(x)$ achieved by using different methodologies. We consider the unidimensional function

$$f(x) = \log(x), \quad x \in \mathcal{X} = [0.1, 12], \tag{11}$$

so that we can exactly check the true accuracy of the obtained approximation using several schemes. We consider $\mathbf{X}_0 = [0.1, 12]$ i.e, $m_0 = 2$ starting points. We sequentially and automatically add 50 points using different techniques: using **(a)** the ASI method, **(b)** Sobol sequences and **(c)** a random choice, uniformly within $[0.1, 12]$. We averaged the results over 200 independent runs. In all cases, we use piecewise linear interpolators. We estimate the L_2 distance between $f(x)$ and the approximation $\widehat{f}_t(x)$ (that is function of t),

$$L_2(t) = \int_{\mathcal{X}} (\widehat{f}_t(x) - f(x))^2 dx. \tag{12}$$

Note that since m_0, in this example, we have $m_t = t + 2$, hence $L_2(t)$ can be expressed as function of m_t as $L(m_t) = L_2(t + 2)$. We consider the two extreme case of constant β_t, i.e., $\beta_t = 0$ for all t and $\beta_t = 1$ for all t. The first case corresponds to $A_t(x) = D_t(x)$ given in Eq. (6). We can observe that the ASI schemes outperform the other strategies. Furthermore, the incorporation of the geometric information in ASI provides the best results.

4.2. Approximation of MODTRAN5

In this section, we focus on the optimization of selected points for a MODTRAN5-based LUT. MODTRAN5 is considered as *de facto* standard atmospheric RTM for atmospheric correction applications [1]. This RTM solves the radiative transfer equation in the atmosphere considering the effect of scattering and absorption by gasses and aerosols for a flexible configuration of viewing and illumination conditions and surface reflectance. In our test application, and for the sake of simplicity, we have considered $D = 2$ with the Aerosol Optical Thickness at 550 nm (τ) and ground elevation (h) as key input parameters. The underlying function $f(\mathbf{x})$ consists therefore on the taget-to-sensor transmittance obtained from the execution of MODTRAN5 at given values of τ and h at the single output wavelength of 760 nm (i.e. bottom of the O_2-A band). The input parameter space is bounded to 0.05-0.4 for τ and 0-3 km for h. In order to test the accuracy of the different schemes, we have evaluated $f(x)$ at all the possible 1750 combinations of 35 values of τ and 50 values of h. Namely, this thin grid represents the ground-truth in this example.

We test **(a)** a random approach choosing points uniformly within $\mathcal{X} = [0.05, 0.4] \times [0, 3]$, **(b)** the Latin Hypercube sampling (see, e.g., [6]), **(c)** AGAPE in [8] and **(d)** ASI with $\beta_t = 0$ (with the algorithm in [23] for obtaining the maximum of the AF). We start with $m_0 = 5$ points, *randomly* chosen in each run, uniformly within the square $[0.05, 0.4] \times [0, 3]$, for all the techniques. We compute the final number of nodes m_t required to obtain an L_2 distance between f and $\widehat{f}$ smaller than 0.03, defined in Eq. (12). The results, averaged over 200 runs, are given in Table 2. ASI requires the addition of ≈ 5 new points to obtain a distance smaller than 0.03.

Table 2. Averaged (over 200 runs) number of nodes m_t.

Random	**Latin Hypercube** [6]	**AGAPE** [8]	**ASI**
20.02	13.39	10.77	9.83

5. CONCLUSIONS

We introduced an adaptive sequential interpolation scheme inspired in active learning procedure. The method can be employed for approximating complex and expensive systems. For instance, it can efficiently construct emulators and optimal look-up-tables for costly radiative transfer models (RTMs). We illustrated the good capabilities of the method in two examples, one involving a complex atmospheric model widely used on Earth observation.

6. REFERENCES

[1] A. Berk, G.P. Anderson, P.K. Acharya, L.S. Bernstein, L. Muratov, J. Lee, M. Fox, S.M. Adler-Golden, J.H. Chetwynd, M.L. Hoke, R.B. Lockwood, J.A. Gardner, T.W. Cooley, C.C. Borel, P.E. Lewis, and E.P. Shettle,

"MODTRAN5: 2006 update," *The International Society for Optical Engineering*, 2006.

[2] 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.

[3] G. Camps-Valls, D. Tuia, L. Gómez-Chova, S. Jiménez, and J. Malo, Eds., *Remote Sensing Image Processing*, Morgan & Claypool Publishers, LaPorte, CO, USA, Sept 2011.

[4] Luis Guanter, Rudolf Richter, and Hermann Kaufmann, "On the application of the MODTRAN4 atmospheric radiative transfer code to optical remote sensing," *International Journal of Remote Sensing*, vol. 30, no. 6, pp. 1407–1424, 2009.

[5] 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.

[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. P. Rivera, J. Verrelst, J. Gómez-Dans, J. Muñoz Marí, J. Moreno, and G. Camps-Valls, "An emulator toolbox to approximate radiative transfer models with statistical learning," *Remote Sensing*, vol. 7, no. 7, pp. 9347, 2015.

[10] J. Verrelst, J.P. Rivera, 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.

[11] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies," *J. Mach. Learn. Res.*, vol. 9, pp. 235–284, 2008.

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

[13] Y. Yoo and A. Ortega, "Adaptive quantization without side information using scalar-vector quantization and trellis coded quantization," in *IEEE Asilomar Conference on Signals, Systems and Computers (ASILOMAR)*, 1995, vol. 2, pp. 1398–1402.

[14] R. I. Klein, "Star formation with 3-d adaptive mesh refinement: the collapse and fragmentation of molecular clouds," *Journal of Computational and Applied Mathematics*, vol. 109, no. 1, pp. 123 – 152, 1999.

[15] K. Chaloner and I. Verdinelli, "Bayesian experimental design: A review," *Statistical Science*, vol. 10, no. 3, pp. 237–304, 1995.

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

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

[18] J. Verrelst, S. Dethier, J.P. Rivera, J. Muñoz-Mar, G. Camps-Valls, and J. Moreno, "Active learning methods for efficient hybrid biophysical variable retrieval," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 7, pp. 1012–1016, 2016.

[19] 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.

[20] J. V. Servera, L. Alonso, L. Martino, N. Sabater, J. Verrelst, G. Camps-Valls, and J. Moreno, "Gradient-based automatic Lookup Table generator for Radiative Transfer Models," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 2, pp. 1040–1048, 2019.

[21] L. Martino, J. Vicent, and G. Camps-Valls, "Automatic Emulation by adaptive Relevance Vector Machines," *Scandinavian Conference on Image Analysis (SCIA)*, pp. 1–11, 2017.

[22] U. M. Ascher and C. Greif, *A First Course in Numerical Methods*, Society for Industrial and Applied Mathematics (SIAM), 2011.

[23] 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.