Laplace Approximation for Divisive Gaussian Processes for Nonstationary Regression

Luis Muñoz-González, Miguel Lázaro-Gredilla, *Member, IEEE,*
and Aníbal R. Figueiras-Vidal, *Fellow, IEEE*

Abstract—The standard Gaussian Process regression (GP) is usually formulated under stationary hypotheses: The noise power is considered constant throughout the input space and the covariance of the prior distribution is typically modeled as depending only on the difference between input samples. These assumptions can be too restrictive and unrealistic for many real-world problems. Although nonstationarity can be achieved using specific covariance functions, they require a prior knowledge of the kind of nonstationarity, not available for most applications. In this paper we propose to use the Laplace approximation to make inference in a divisive GP model to perform nonstationary regression, including heteroscedastic noise cases. The log-concavity of the likelihood ensures a unimodal posterior and makes that the Laplace approximation converges to a unique maximum. The characteristics of the likelihood also allow to obtain accurate posterior approximations when compared to the Expectation Propagation (EP) approximations and the asymptotically exact posterior provided by a Markov Chain Monte Carlo implementation with Elliptical Slice Sampling (ESS), but at a reduced computational load with respect to both, EP and ESS.

Index Terms—Gaussian Processes, Nonstationary Regression, Laplace approximation, Heteroscedastic Regression

1 Introduction

Gaussian Processes (GPs) [1] are a powerful nonparametric Bayesian tool for nonlinear regression. GPs model the observations as the sum of an unknown latent function plus a Gaussian noise. Unlike other regression methods, GPs proceed in a Bayesian fashion to infer the posterior distribution of the unknown function through the likelihood and a prior distribution placed over the unknown function. So, they produce probabilistic predictions in a natural way. GPs usually employ a reduced number of hyperparameters which can be tuned with a simple continuous optimization of the evidence: This make them resilient to overfitting.

All these advantages come to a price: The $\mathcal{O}(N^3)$ time scalability with the number of training samples. However, it is possible to achieve a better scalability using approximate inference with sparse GPs [2]–[4], making inference affordable for large-scale data sets. For a more exhaustive review on GPs see [5].

Stationarity is a frequent assumption in the standard GP regression. The noise is assumed homoscedastic, i.e., with constant power throughout the input space. Stationary covariance functions are typically employed, since nonstationary covariance functions require previous knowledge of the type of nonstationarity.

Heteroscedastic GP models have been also proposed in the literature. Most of them are based on [6], where two GPs are used to model the mean and the log-noise power, respectively. As the posterior, the evidence, and the predictive distribution of this heteroscedastic model are not analytically tractable, Markov Chain Monte Carlo (MCMC) methods or approximate inference algorithms are needed to make inference on the model. Gibbs sampling is proposed in [6], point estimation of the log-noise is proposed in [7] and improved in [8], an iterative EP posterior approximation is described in [9], and a variational approximation is proposed in [10]. The main drawback of this heteroscedastic model is that the likelihood is not log-concave, which limits the application of approximate inference techniques. In contrast, using natural parameters to reformulate the model in [6], as proposed in [11], leads to a log-concave likelihood and, then, to a unimodal posterior.

A GP Product Model (GPPM) is introduced in [12], where amplitude nonstationarities are modeled with the pointwise product of two latent functions. GPPM cannot be solved analytically, so that an EP posterior approximation is proposed. Since the likelihood is not log-concave, damping and skipping techniques are applied to alleviate EP convergence problems. Due to the Gauss-Hermite approximations that are needed to approximate the calculations of the moments in the EP algorithm, it is not possible to use a Maximum Likelihood of level II (ML-II) implementation to tune the hyperparameters. On the other hand, the GPPM model does not consider input-dependent noise.

The Divisive GP (DGP) model introduced in [13] achieves amplitude nonstationarity, including het-

• *The authors are with the Department of Signal Theory and Communications, Universidad Carlos III de Madrid, Spain.
E-mail: {lmunoz, miguel, arfv}@tsc.uc3m.es.*

This work was supported by the Spanish Government under Project TIN 2011/24533.

eroscedastic noise cases, combining two GPs. The experimental results of the proposed EP method showed the high quality of the posterior approximations compared to an MCMC implementation using Elliptical Slice Sampling (ESS) [14]. The regression performance on different data sets is better than the standard GP and other heteroscedastic GP methods such as the Variational Heteroscedastic GP Regresion (VHGPR) described in [10]. Although the likelihood of the DGP model in [13] is log-concave, which favors the convergence of EP [15], it has not be proven to be a sufficient condition for convergence. Another drawback of the proposed EP method is the high computational cost.

In this paper we propose to use the Laplace approximation to make inference on the DGP model and overcome the limitations of the EP method. The likelihood log-concavity leads to a unimodal posterior that guarantees the convergence of the Laplace method. Besides, the characteristics of the likelihood allows high-quality Gaussian posterior approximations, similar to those produced by EP, but at a reduced computational burden, as shown in the experiments, allowing to apply the DGP model in larger data sets.

The rest of the paper is organized as follows: In Section 2 we review the DGP. In Section 3 inference with the Laplace method on the DGP model is described. Experimental results on synthetic and real data sets are shown in Section 4. Finally, in Section 5 we present the main conclusions and further research lines.

2 Divisive GPR Model

Given a set of input-output pairs, $\{\mathbf{x}_n, y_n\}_{n=1}^N$, where $\mathbf{x}_n \in \mathcal{R}^D$ and $y_n \in \mathcal{R}$, the DGP model describes the observations as a possibly noisy stationary latent function divided by another stationary latent function that models possible amplitude nonstationarities affecting to both the latent function and the noise associated to it. So, the observations can be expressed as

$$y(\mathbf{x}_n) = \frac{f(\mathbf{x}_n)}{g^+(\mathbf{x}_n)} + \varepsilon_n \tag{1}$$

where $f(\mathbf{x})$ is the noisy stationary latent function, $g^+(\mathbf{x})$ is the modulating function, defined as the positive part of some noise-free latent function $g(\mathbf{x})$, i.e., $g^+(\mathbf{x}) = \max(g(\mathbf{x}), 0)$, and ε_n is an input-dependent Gaussian noise term that can be modeled as $\varepsilon \sim \mathcal{N}(0, c/(g^+(\mathbf{x}))^2)$, where c is a noise power constant scale factor.

The likelihood of the DGP model given an observation $y_n = y(\mathbf{x}_n)$ is:

$$p(y_n|f_n, g_n) = \mathcal{N}(y_n|f_n/g_n^+, c/(g_n^+)^2) \tag{2}$$

where $f_n = f(\mathbf{x}_n)$ and $g_n^+ = g^+(\mathbf{x}_n)$. It can be appreciated that the DGP model also includes the standard GPR if the latent function g is considered constant.

We place GP priors on f and g, so that

$$\begin{aligned} f(\mathbf{x}) &\sim \mathcal{GP}(0, k_f(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}_f) + \sigma_f^2\ \delta_{\mathbf{x}\mathbf{x}'}) \\ g(\mathbf{x}) &\sim \mathcal{GP}(\mu_0, k_g(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}_g)) \end{aligned} \tag{3}$$

where $k_f(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}_f)$, $k_g(\mathbf{x}, \mathbf{x}'; \boldsymbol{\theta}_g)$, are any valid covariance functions, σ_f^2 is a homoscedastic noise hyperparameter multiplied by the Kronecker delta, and μ_0 is the mean of the latent function g that, together with the constant c, modulates the mean power of the heteroscedastic noise. The model is fully specified by the covariance functions, their hyperparameters ($\boldsymbol{\theta}_f$, σ_f^2, $\boldsymbol{\theta}_g$), μ_0, and the noise constant c. A more detailed description of the DGP model can be found in [13].

With the likelihood in (2) and the GP priors in (3), the posterior on the latent functions f and g at the training points is given by[1]

$$p(\mathbf{f}, \mathbf{g}|\mathbf{y}) = \frac{p(\mathbf{y}|\mathbf{f}, \mathbf{g})p(\mathbf{f})p(\mathbf{g})}{p(\mathbf{y})} \tag{4}$$

where $p(\mathbf{y})$ is the evidence or marginal likelihood.

As the posterior and the evidence cannot be computed analytically, we need to apply approximate inference techniques or MCMC methods to make inference on the DGP model. Although EP reduces the computational burden w.r.t. MCMC methods, as shown in [13], the time required to train EP-DGP is higher compared to the standard GP or other heteroscedastic GP methods such as VHGPR.

Here, we propose to use the Laplace approximation to make inference on the DGP model. As the posterior is unimodal [13], the Laplace method converges to a unique maximum and the simplicity of the algorithm makes it computationally appealing. The characteristics of the likelihood in the DGP model lead to a posterior with quite a Gaussian shape, which makes the Laplace approximation a convenient alternative to produce as good posterior approximations as EP-DGP, but at a reduced computational cost.

3 Laplace approximation for DGP

The Laplace method aims to approximate the posterior $p(\mathbf{f}, \mathbf{g}|\mathbf{y})$ with a Gaussian distribution $q(\mathbf{f}, \mathbf{g}|\mathbf{y})$ doing a second order Taylor expansion of $\log p(\mathbf{f}, \mathbf{g}|\mathbf{y})$ around the maximum of the posterior

$$q(\boldsymbol{\phi}|\mathbf{y}) = \mathcal{N}(\boldsymbol{\phi}|\hat{\boldsymbol{\phi}}, A^{-1}) \tag{5}$$

where $\boldsymbol{\phi} = [f, g]$, $\hat{\boldsymbol{\phi}} = \arg\max_{\boldsymbol{\phi}} p(\boldsymbol{\phi}|\mathbf{y})$, and $A = -\nabla\nabla \log p(\boldsymbol{\phi}|\mathbf{y})|_{\boldsymbol{\phi}=\hat{\boldsymbol{\phi}}}$ is the Hessian of the log-posterior at $\hat{\boldsymbol{\phi}} = [\hat{\mathbf{f}}, \hat{\mathbf{g}}]$.

For the sake of simplicity we have considered an equivalent model to that described in previous section

1. We use vectorized forms to refer to the observations and the latent functions evaluated at the training points, so that $\mathbf{y} \equiv \{y_n\}_{n=1}^N$, $\mathbf{f} \equiv \{f_n\}_{n=1}^N$, and $\mathbf{g} \equiv \{g_n\}_{n=1}^N$. For the sake of simplicity we omit the conditioning on the training inputs X and the set of hyperparameters $\boldsymbol{\theta} = [\boldsymbol{\theta}_f, \sigma_f^2, \boldsymbol{\theta}_g, \mu_0]^T$.

placing the offset μ_0 of the latent function g in the likelihood, so that the prior on g has zero mean.

The expression needed to maximize the log-posterior w.r.t. the latent functions at the training points removing the elements that do not depend on $\mathbf{f}$ and $\mathbf{g}$ reduces to

$$\mathbf{\Psi}(\boldsymbol{\phi}) = \log p(\mathbf{y}|\boldsymbol{\phi}) - \frac{1}{2}\boldsymbol{\phi}^T K^{-1}\boldsymbol{\phi} \tag{6}$$

where K is the $2N$-square block diagonal matrix

$$K = \begin{bmatrix} K_{\mathbf{f}} & 0_N \\ 0_N & K_{\mathbf{g}} \end{bmatrix} \tag{7}$$

$K_{\mathbf{f}}$ and $K_{\mathbf{g}}$ being the covariance matrices of the GP priors on f and g, respectively, and 0_N being a N-square matrix with all its elements equal to zero.

Differentiating the functional $\mathbf{\Psi}(\boldsymbol{\phi})$ w.r.t. $\boldsymbol{\phi}$

$$\nabla\mathbf{\Psi}(\boldsymbol{\phi}) = \nabla \log p(\mathbf{y}|\boldsymbol{\phi}) - K^{-1}\boldsymbol{\phi} \tag{8}$$

where $\nabla \log p(\mathbf{y}|\boldsymbol{\phi})$ is a vector of length $2N$ with

$$\begin{aligned}[\nabla \log p(\mathbf{y}|\boldsymbol{\phi})]_n &= \frac{\partial \log p(y_n|f_n, g_n)}{\partial f_n} \\ &= \frac{g_n'^{+} y_n - f_n}{c}\end{aligned} \tag{9}$$

for $1 \leq n \leq N$, with $g_n' = g_n + \mu_0$, and

$$\begin{aligned}[\nabla \log p(\mathbf{y}|\boldsymbol{\phi})]_n &= \frac{\partial \log p(y_{n'}|f_{n'}, g_{n'})}{\partial g_{n'}} \\ &= \begin{cases} \dfrac{1}{g_{n'}'} - \dfrac{y_{n'}^2 g_{n'}' - y_{n'} f_{n'}}{c}, & g_{n'}' > 0 \\ 0, & g_{n'}' \leq 0 \end{cases}\end{aligned} \tag{10}$$

for $N < n \leq 2N$, with $n' = n - N$.

Then, making $\nabla\mathbf{\Psi}(\boldsymbol{\phi}) = 0$ we obtain the following self-consistent equation at the maximum of $\nabla\mathbf{\Psi}(\boldsymbol{\phi})$

$$\hat{\boldsymbol{\phi}} = K(\nabla \log p(\mathbf{y}|\hat{\boldsymbol{\phi}})) \tag{11}$$

Applying Newton search, we have

$$\boldsymbol{\phi}_{\text{new}} = (K^{-1} + W)^{-1}(W\boldsymbol{\phi} + \nabla \log p(\mathbf{y}|\boldsymbol{\phi})) \tag{12}$$

where $W = -\nabla\nabla \log p(\mathbf{y}|\boldsymbol{\phi})$ is the negative Hessian of the likelihood which yields the block diagonal matrix

$$W = \begin{bmatrix} W_f & W_{fg} \\ W_{fg} & W_g \end{bmatrix} \tag{13}$$

The n-th diagonal elements of each of the three diagonal matrices W_f, W_g, and W_{fg} are given by

$$[W_f]_{nn} = -\frac{\partial^2 \log p(y_n|f_n, g_n)}{\partial f_n^2} = \frac{1}{c} \tag{14}$$

$$[W_g]_{nn} = -\frac{\partial^2 \log p(y_n|f_n, g_n)}{\partial g_n^2} = \begin{cases} \dfrac{1}{g_n'^2} + \dfrac{y_n^2}{c}, & g_n' > 0 \\ 0, & g_n' \leq 0 \end{cases} \tag{15}$$

$$[W_{fg}]_{nn} = -\frac{\partial^2 \log p(y_n|f_n, g_n)}{\partial f_n \partial g_n} = \begin{cases} -\dfrac{y_n}{c}, & g_n' > 0 \\ 0, & g_n' \leq 0 \end{cases} \tag{16}$$

As the likelihood is (jointly) log-concave on f and g [13], the Hessian results in a negative definite matrix. This leads $\mathbf{\Psi}(\boldsymbol{\phi})$ to be concave. Therefore, the Laplace method converges to a unique maximum.

3.1 Approximate Marginal Likelihood

The approximate marginal likelihood needed to find the set of hyperparameters using an ML-II implementation can be written as

$$\log q(\mathbf{y}) = -\frac{1}{2}\hat{\boldsymbol{\phi}}^T K^{-1}\hat{\boldsymbol{\phi}} + \log p(\mathbf{y}|\hat{\boldsymbol{\phi}}) - \frac{1}{2}\log|B| \tag{17}$$

where $|B| = |K| \cdot |K^{-1} + W|$.

To calculate the derivatives of (17) w.r.t. the hyperparameters we follow a treatment similar to [5] for the case of GP Classification with the Laplace method. The details of these calculations are described in the Appendix.

3.2 Predictive distribution

To calculate the approximate predictive distribution for a test sample $\mathbf{x}_*$, $q(y_*|\mathbf{x}_*)$, we need to calculate the approximate predictive distributions for the latent functions f and g. The approximate predictive distribution $q(f_*)$ is

$$\begin{aligned} q(f_*) &= \int p(f_*|\mathbf{x}_*, \mathbf{f})q(\mathbf{f}, \mathbf{g}|\mathbf{y})d\mathbf{f}d\mathbf{g} = \\ &= \mathcal{N}(f_*|\mu_{f_*}, \sigma_{f_*}^2)\end{aligned} \tag{18}$$

where

$$\begin{aligned}\mu_{f_*} &= \mathbf{k}_{*f}^T(\nabla \log p(\mathbf{y}|\hat{\boldsymbol{\phi}})|_f) \\ \sigma_{f_*}^2 &= k_{f_{**}} - \mathbf{k}_{*f}^T(K_f + W_f^{-1})^{-1}\mathbf{k}_{*f}\end{aligned} \tag{19}$$

with $[\mathbf{k}_{*f}]_j = k_f(\mathbf{x}_*, \mathbf{x}_j)$, $k_{f_{**}} = k_f(\mathbf{x}_*, \mathbf{x}_*)$, and $\nabla \log p(\mathbf{y}|\hat{\boldsymbol{\phi}})|_f$ are the first N elements of vector $\nabla \log p(\mathbf{y}|\hat{\boldsymbol{\phi}})$ (those corresponding to the partial derivatives w.r.t. f evaluated at $\hat{\boldsymbol{f}}$).

Following a similar treatment, the approximate predictive distribution for g is also Gaussian, $q(g_*) = \mathcal{N}(g_*|\mu_{g_*}, \sigma_{g_*}^2)$, with

$$\begin{aligned}\mu_{g_*} &= \mathbf{k}_{*g}^T(\nabla \log p(\mathbf{y}|\hat{\boldsymbol{\phi}})|_g) \\ \sigma_{g_*}^2 &= k_{g_{**}} - \mathbf{k}_{*g}^T(K_g + W_g^{-1})^{-1}\mathbf{k}_{*g}\end{aligned} \tag{20}$$

and $\nabla \log p(\mathbf{y}|\hat{\boldsymbol{\phi}})|_g$ is a vector containing the last N elements of $\nabla \log p(\mathbf{y}|\hat{\boldsymbol{\phi}})$.

We have neglected the covariance term $\sigma_{f_* g_*}$, as the correlation between latent functions is expected to be an artifact. Then, the approximate predictive distribution for y_* can be calculated as

$$\begin{aligned} q(y_*) &= \int p(y_*|f_*, g_*)q(f_*)q(g_*)df_*dg_* = \\ &= Z_*(y_*)\ \tilde{\mu}_{g_{*t}}\ \Phi\left(-\frac{\tilde{\mu}_{g_*}}{\tilde{\sigma}_{g_*}}\right)\end{aligned} \tag{21}$$

where

$$Z_*(y_*) = \mathcal{N}(\mu_{f_*}|\mu_{g_*}y_*, c + \sigma^2_{f_*} + \sigma^2_{g_*}y^2_*) \tag{22}$$

and $\tilde{\mu}_{g_{*t}}$ is the mean of the Gaussian $\mathcal{N}(g_*|\tilde{\mu}_{g_*}, \tilde{\sigma}^2_{g_*})$ truncated to the positive values of g_* with

$$\begin{aligned}\tilde{\sigma}^2_{g_*} &= \left(\sigma^{-2}_{g_*} + \frac{y^2_*}{c+\sigma^2_{f_*}}\right)^{-1} \\ \tilde{\mu}_{g_*} &= \tilde{\sigma}^2_{g_*}\left(\frac{\mu_{g_*}}{\sigma^2_{g_*}} + \frac{y_*\mu_{f_*}}{c+\sigma^2_{f_*}}\right)\end{aligned} \tag{23}$$

Although $q(y_*)$ can be calculated analytically, there is no analytical solution for the mean of $q(y_*)$. In general, the mean may not exist, as in the case of Cauchy distributions. However, we can sidestep this problem using the median as an estimator for the targets. To do that, we can calculate the expression of the cumulative distribution $F_{y_*}(\alpha)$ following a similar treatment than in the case of the ratio of two correlated normal random variables [16], so that:

$$F_{y_*}(\alpha) = \mathrm{L}\left(\frac{\mu_{g_*}\alpha - \mu_{f_*}}{a(\alpha)}, \frac{\mu_{g_*}}{\sigma_{g_*}}; \frac{\sigma_{g_*}\alpha}{a(\alpha)}\right) + \Phi\left(\frac{\mu_{g_*}}{\sigma_{g_*}}\right) \tag{24}$$

with

$$a(\alpha) = \sqrt{\sigma^2_{g_*}\alpha^2 + c + \sigma^2_{f_*}} \tag{25}$$

where $\mathrm{L}(h, k, \gamma)$ is the standard bivariate normal integral

$$\begin{aligned}\mathrm{L}(h,k,\gamma) &= \frac{1}{2\pi\sqrt{1-\gamma^2}} \times \\ &\times \int_h^\infty \int_k^\infty \exp\left(-\frac{x^2 - 2\gamma xy + y^2}{2(1-\gamma^2)}\right) dx\, dy\end{aligned} \tag{26}$$

If $\mu_{g_*}/\sigma_{g_*} \to \infty$, the cumulative distribution $F_{y_*}(\alpha)$ can be approximated by

$$F_{y_*}(\alpha) \to \Phi\left(\frac{\mu_{g_*}\alpha - \mu_{f_*}}{a(\alpha)}\right) \tag{27}$$

To obtain the predictive median m_{y_*} we need the inverse cumulative distribution, i.e., $m_{y_*} = F^{-1}_{y_*}(1/2)$. Although calculation of $F^{-1}_{y_*}(\alpha)$ is not analytically tractable, we can approximate the solution using numerical root-finding methods as the bisection or the secant algorithms, for example. The same serves for quantile estimation.

4 Experiments

We present experimental results to evaluate the performance of L-DGP compared with EP-DGP and MCMC-DGP. We also include standard GPR, VHGPR, and SVR as comparison benchmarks[2]. First, we assess the quality of the Laplace approximation compared with the other DGP methods on a synthetic data set. Secondly, we consider several small and medium size data sets, including a computational cost experiment for one of the proposed problems.

2. The Matlab implementation of the DGP algorithms used for the experiments is avaliable at http://github.com/lmunoz-gonzalez/Divisive-Gaussian-Processes

4.1 Synthetic experiment

We have worked with the synthetic heteroscedastic problem described in [17]. According to the proposed mean and noise power distributions, we have generated 200 samples to train the three DGP methods and the standard GP. For the standard GP, we have used an Squared Exponential (SE) covariance function, given by $k_{\mathrm{SE}}(\mathbf{x}, \mathbf{x}') = \sigma_0^2 \exp(-\|\mathbf{x} - \mathbf{x}'\|^2/2\ell^2)$. For DGP methods, we have used an SE covariance function for $g(\mathbf{x})$ and an SE plus noise covariance function for $f(\mathbf{x})$. Assuming that $f(\mathbf{x})$ is noisy, it is possible to establish a tradeoff between the heteroscedastic noise component modeled by $c/(g^+(\mathbf{x}))^2$ and the homoscedastic noise in $f(\mathbf{x})$.

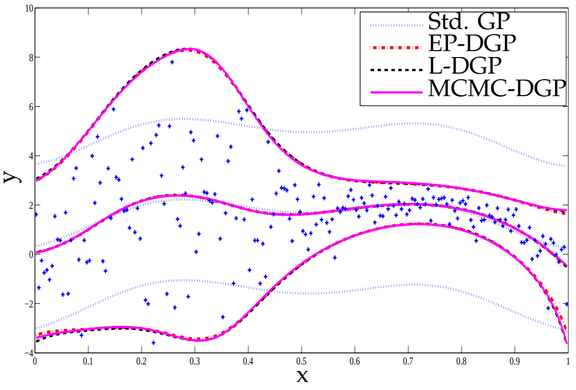

Fig. 1. Experiment using the synthetic data set proposed in [17] with 200 training samples. The estimated mean and twice the standard deviation are given for the standard GP prediction (dotted line). Median estimation is provided for EP-DGP (dashed-dotted line), L-DGP (dashed line), and MCMC-DGP (continuous line), along with the quantiles 0.023 and 0.977.

To initialize the EP-DGP and L-DGP hyperparameters we have applied the same procedure described in [13]. For MCMC-DPG, we have not implemented any search procedure to tune the hyperparameters. To evaluate the quality of the L-DGP approximation, we set MCMC-DGP hyperparameters equal to those obtained from the trained L-DGP.

The results are shown in Figure 1. For the DGP methods, we show the estimation of quantiles 0.023 and 0.977, that are equivalent to twice the standard deviation in the case of the standard GPR. We can observe that the solutions provided by all DGP methods are very similar, both in the median and the quantiles estimation. Note that the results of L-DGP and the exact solution provided by MCMC-DGP (using the same set of hyperparameters than L-DGP) almost match, showing the good quality of the proposed Laplace approximation.

4.2 Regression performance

We present performance results of L-DGP on several real data sets, comparing them with EP-DGP, MCMC-DGP, the standard GP, VHGPR, and the standard Support Vector Regression (SVR) [18].

As performance measures we have used the Normalized Mean Squared Error (NMSE)

$$\text{NMSE} = \frac{\sum_{j=1}^{n_*} (y_{*j} - \hat{y}_{*j})^2}{\sum_{j=1}^{n_*} (y_{*j} - \bar{y})^2} \tag{28}$$

the Normalized Mean Absolute Error (NMAE)

$$\text{NMAE} = \frac{\sum_{j=1}^{n_*} |y_{*j} - \hat{y}_{*j}|}{\sum_{j=1}^{n_*} |y_{*j} - \bar{y}|} \tag{29}$$

and the Negative Log-Predictive Density (NLPD), which is given by

$$\text{NLPD} = -\frac{1}{n_*} \sum_{j=1}^{n_*} \log\, p(y_{*j}|x_{*j}) \tag{30}$$

where n_* is the number of test samples, y_{*j} is the j-*th* test observation, $\hat{y}_{*j}$ is the predictive mean of the posterior of that observation, and $\bar{y}$ is the mean of the training observations.

To train EP-DGP and L-DGP methods we have initialized the hyperparameters using the same procedure applied for the synthetic experiment. For MCMC-DGP, we use the set of hyperparameters found by L-DGP. For VHGPR, we have used an SE covariance function for the mean latent function and an SE covariance function plus noise for the log-noise latent function, initializing the hyperparameters as in [10].

TABLE 1
Characteristics of the data sets used for the experiments.

Data set	Dim	# Tr samples	# Test samples
Ozo	3	89	22
Bod	13	202	50
Ser	4	134	33
Aut	7	314	78
Hou	13	253	253
Con	8	515	515
Win	11	1700	3198
Par	21	1000	4875

For the SVR, we have used a Radial Basis Function (RBF) kernel. The kernel width σ and the cost parameter C have been set to the values that minimize the averaged test NMSE over all the splits created for each data set. Notice that this induces a clear advantage for the SVR designs w.r.t. the other methods. For a fair comparison, other methodology should be applied, as for example cross validation procedures. However, due to the high number of splits used for most of the experiments, the computational burden of this approach would be very high. The values of C have been explored in the range $[2 \cdot 10^{-5}, 2 \cdot 10^{10}]$ with values of the form $2 \cdot 10^p$, with $-5 \leq p \leq 10$. For σ, we have also explored values of that form in the range $[2 \cdot 10^{-5}, 2 \cdot 10^5]$. To calculate the NLPD for the SVR we have assumed a Gaussian distribution for the predictions with a constant noise power, which has been estimated as the mean squared error of the train predictions.

The characteristics of the data sets used for the experiments are shown in Table 1. For the first 6 problems appearing in Table 1 (*Ozo* [19], *Bod*, *Ser*, *Aut*, *Hou*, and *Con* [20]) we have made 300 random splits to obtain a more complete evaluation of the compared algorithms. For the bigger data sets (*Win*, *Par* [20]), we have made a single split due to the high computational costs to train the GP methods.

For *Ozo*, *Bod*, *Ser*, and *Aut* we have made 300 random splits with 80% training samples and 20% test samples. For *Hou* and *Con* we have made the splits with 50% training and test samples.

The results of the experiments for the small and medium-size data sets are presented in Table 2. We do not present results for MCMC-DGP on *Con* because the computational burden is very high. We have performed a Wilcoxon Rank-Sum test to assess statistical significance at the 5% level.

From Table 2, it can be said that the SVR performs worse than the GP based methods with statistical significant differences in 5 of the 6 problems in terms of NMSE, NMAE, and NLPD. It can also be noticed that the three DGP methods never perform worse than the standard GP in any of the three performance measures. Even when the problem is homoscedastic, as it seems to be the case for *Bod*, the performances of the DGP methods are the same than the performance of the standard GP.

In terms of NMSE, we can say the following:

- L-DGP, MCMC-DGP, and EP-DGP performances are similar. However there are statistical significant improvements of L-DGP w.r.t. EP-DGP in *Con* and w.r.t. MCMC-DGP and EP-DGP in *Hou*.
- DGP methods also outperform VHGPR with significant differences in 4 of the 6 data sets.
- L-DGP improves the standard GP results in *Ozo*, *Ser*, and *Con*, whereas MCMC-DGP and EP-DGP outperform the standard GP in *Ozo* and *Con*.

The results in terms of NMAE show that:

- L-DGP, MCMC-DGP, and EP-DGP perform similarly, except for *Hou*, where L-DGP outperforms MCMC-DGP and EP-DGP, and *Con*, where L-DGP improves EP-DGP performance.
- DGP methods improve with statistical significance GP and VHGPR in *Ozo*, *Ser*, *Hou*, and *Con*.
- In none of the data sets DGP algorithms perform worse than the other GP methods.

Finally, in terms of NLPD it can be observed that:

- VHGPR and the DGP algorithms outperform the

TABLE 2
Experimental test results on small and medium size multidimensional data sets, providing the average NMSE, NMAE, and NLPD plus/minus one standard deviation. Statistically significant improvements are marked as ● w.r.t. standard GP, ○ w.r.t. VHGPR, ⋆ w.r.t. EP-DGP, ◇ w.r.t. L-DGP, ▷ w.r.t. MCMC-DGP, and † w.r.t. the SVR. Statistical significance is measured according to a Wilcoxon Rank-Sum test at the 5% level.

		Average NMSE	Average NMAE	Average NLPD
Ozo:	Std. GP	0.281 ± 0.091 †	0.484 ± 0.085	4.323 ± 0.271 †
	VHGPR	0.282 ± 0.083 †	0.477 ± 0.079	4.156 ± 0.209 ● †
	EP-DGP	0.258 ± 0.079 ● ○†	0.462 ± 0.079 ● ○†	4.071 ± 0.151 ● ○†
	L-DGP	0.256 ± 0.079 ● ○†	0.460 ± 0.079 ● ○†	4.072 ± 0.137 ● ○†
	MCMC-DGP	0.257 ± 0.080 ● ○†	0.461 ± 0.079 ● ○†	4.073 ± 0.143 ● ○†
	SVR ($C = 200; \sigma = 0.2$)	0.300 ± 0.094	0.486 ± 0.085	4.443 ± 0.530
Bod:	Std. GP	0.290 ± 0.061	0.525 ± 0.058	-0.989 ± 0.096 †
	VHGPR	0.290 ± 0.061	0.525 ± 0.058	-0.989 ± 0.097 †
	EP-DGP	0.291 ± 0.061	0.526 ± 0.058	-0.988 ± 0.098
	L-DGP	0.290 ± 0.061	0.525 ± 0.058	-0.989 ± 0.096 †
	MCMC-DGP	0.291 ± 0.061	0.526 ± 0.058	-0.988 ± 0.098
	SVR ($C = 20; \sigma = 0.002$)	0.294 ± 0.055	0.533 ± 0.057	-0.970 ± 0.116
Ser:	Std. GP	0.166 ± 0.090 †	0.302 ± 0.062 †	-0.871 ± 0.676 †
	VHGPR	0.182 ± 0.124 †	0.267 ± 0.076 ● †	-1.972 ± 0.412 ● ⋆ ◇ ▷†
	EP-DGP	0.151 ± 0.109 ○ †	0.245 ± 0.065 ● ○†	-1.892 ± 0.173 ● †
	L-DGP	0.143 ± 0.104 ● ○†	0.227 ± 0.062 ● ○ ⋆ ▷†	-1.940 ± 0.138 ● ⋆ ▷ †
	MCMC-DGP	0.151 ± 0.109 ○ †	0.245 ± 0.065 ● ○†	-1.892 ± 0.173 ● †
	SVR ($C = 200; \sigma = 0.02$)	0.224 ± 0.094	0.443 ± 0.070	-0.688 ± 0.650
Aut:	Std. GP	0.116 ± 0.030 †	0.289 ± 0.031 †	-1.234 ± 0.137 †
	VHGPR	0.117 ± 0.029 †	0.288 ± 0.031 †	-1.361 ± 0.117 ● †
	EP-DGP	0.119 ± 0.031 †	0.287 ± 0.033 †	-1.354 ± 0.103 ● †
	L-DGP	0.116 ± 0.030 †	0.284 ± 0.032 †	-1.359 ± 0.093 ● †
	MCMC-DGP	0.119 ± 0.031 †	0.287 ± 0.033 †	-1.354 ± 0.103 ● †
	SVR ($C = 2; \sigma = 0.02$)	0.134 ± 0.028	0.321 ± 0.035	-1.143 ± 0.137
Hou:	Std. GP	0.152 ± 0.035 ○ ◇ ▷ †	0.352 ± 0.022 †	2.624 ± 0.122 †
	VHGPR	0.166 ± 0.034 †	0.352 ± 0.022 †	2.564 ± 0.150 ● †
	EP-DGP	0.159 ± 0.037 ○ †	0.345 ± 0.023 ● ○†	2.445 ± 0.075 ● ○†
	L-DGP	0.152 ± 0.036 ○ ⋆ ▷ †	0.336 ± 0.023 ● ○ ⋆ ▷†	2.413 ± 0.057 ● ○ ⋆ ▷†
	MCMC-DGP	0.159 ± 0.037 ○ †	0.345 ± 0.023 ● ○†	2.445 ± 0.075 ● ○†
	SVR ($C = 200; \sigma = 0.02$)	0.177 ± 0.042	0.356 ± 0.024	3.126 ± 0.500
Con:	Std. GP	0.132 ± 0.014 †	0.326 ± 0.014 †	3.162 ± 0.041 †
	VHGPR	0.134 ± 0.013 †	0.327 ± 0.014 †	3.088 ± 0.042 ● †
	EP-DGP	0.123 ± 0.015 ● ○†	0.310 ± 0.014 ● ○†	3.049 ± 0.041 ● ○ ◇ †
	L-DGP	0.120 ± 0.014 ● ○ ⋆ †	0.304 ± 0.014 ● ○ ⋆ †	3.060 ± 0.036 ● ○ †
	SVR ($C = 2000; \sigma = 0.02$)	0.158 ± 0.017	0.353 ± 0.016	3.391 ± 0.106

standard GP in 5 of the 6 proposed data sets with statistical significance, and are not worse for *Bod*.
- Comparing the differences between the DGP methods and VHGPR, it can be appreciated that DGP methods outperform VHGPR in 3 data sets, whereas VHGPR only outperforms DGP methods in *Ser*.
- The performance of L-DGP, MCMC-DGP, and EP-DGP is similar, although L-DGP improves NLPD performance in *Ser* and *Hou*, whereas EP-DGP outperforms EP-DGP in *Con*.

The experimental results for the biggest data sets (*Win* and *Par*) are shown in Table 3. For data set *Win* we observe that:

- The performance in terms of NMSE and NMAE is similar for all the GP methods.
- EP-DGP and L-DGP have better results in terms of NLPD w.r.t. the standard GP and VHGPR.
- SVR outperforms GP methods in terms of NMAE. However, NMSE and NLPD are worse.

For data set *Par* we note that:

- EP-DGP outperforms the other GP methods and the SVR in terms of NMSE and NMAE.
- NLPD performance is similar for EP-DGP and VHGPR, with a slight advantage w.r.t. L-DGP.
- SVR also performs clearly worse than the other GP methods in terms of NLPD.

The results of these experiments support the high quality of the Laplace approximation compared to MCMC-DGP using the same set of hyperparameters. Moreover, the performance of L-DGP is similar to the performance of EP-DGP, which suggests that both approximations are similar.

TABLE 3
Experimental test results on data sets *Win* and *Par*. NMSE, NMAE, and NLPD are provided for the standard GP, VHGPR, EP-DGP, L-DGP, and SVR.

		NMSE	NMAE	NLPD
Win:	Std. GP	0.653	0.838	1.059
	VHGPR	0.657	0.842	1.057
	EP-DGP	0.652	0.839	1.014
	L-DGP	0.649	0.837	1.018
	SVR ($C = 2; \sigma = 0.2$)	0.667	0.818	1.267
Par:	Std. GP	0.216	0.396	-1.983
	VHGPR	0.222	0.395	-2.060
	EP-DGP	0.170	0.370	-2.063
	L-DGP	0.219	0.394	-2.049
	SVR ($C = 200; \sigma = 2 \cdot 10^{-4}$)	0.231	0.487	-1.700

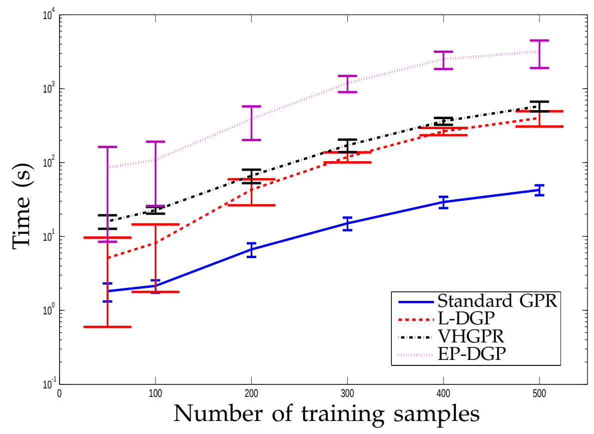

Fig. 2. Results of the time experiment using *Hou* data set with different training sizes. Error bars are shown to represent the training time for the standard GP (continuous line and narrower error bars), L-DGP (dashed line and wider error bars), VHGPR (dashed-dotted line), and EP-DGP (dotted line).

4.3 Computational cost experiments

The complexity of all the GP methods used in the experiments scales in the form $\mathcal{O}(N^3)$. However, the training times of these methods can be quite different. To illustrate this point we have performed an experiment with data set *Hou* to measure the training time for the standard GP, VHGPR, EP-DGP, and L-DGP for different number of training samples. We have measured the training time avoiding the hyperparameters learning, varying the number of training samples from 50 to 500. For a given number of training samples, we have generated 20 random training sets. To set the hyperparameters, we have first trained all the GP methods using all the available samples as training samples.

The average training time[3] along with the corresponding error bars are shown in Figure 2. First of all, it can be appreciated that all the GP methods scale similarly with the number of samples, with is consistent with the theoretic $\mathcal{O}(N^3)$ time scalability. Despite the standard GP is the fastest method, the time required to train L-DGP is lower than VHGPR and EP-DGP training times. The training time reduction of L-DGP w.r.t. EP-DGP is remarkable. For example, for 500 training points, EP-DGP takes around 2000 seconds to train the model, whereas L-DGP takes only 200 seconds (10 times faster).

5 Conclusions

The high computational burden and the lack of a formal proof of convergence (even when the likelihood is log-concave) of the EP approximation proposed to perform inference on the DGP model introduced in [13] motivates the use of the Laplace approximation for the same model. The likelihood log-concavity ensures a unimodal posterior which allows the Laplace approximation to converge to a unique maximum.

The experimental comparisons of the Laplace approximation with the asymptotically unbiased posterior estimates using MCMC-DGP show the high quality of the Laplace posterior approximation. The experimental results obtained for different real data sets show a similar performance of L-DGP compared to EP-DGP, but at a reduced computational cost.

Further research avenues include the extension of the DGP model for classification and multi-output regression tasks, the development of sparse GP methods using the DGP model, and the analysis of new GP models to achieve lengthscale nonstationarity.

Appendix
Derivatives of the approximate log-evidence with respect to the hyperparameters

We detail here the calculations of the partial derivatives of the approximate log-evidence (17) w.r.t. the hyperparameters of the covariance functions, $\boldsymbol{\theta}_f$ and $\boldsymbol{\theta}_g$, and the noise offset μ_0.

To calculate the derivatives we have to take into account not only the explicit terms, i.e. those referred to K in the case of $\boldsymbol{\theta}_f$ and $\boldsymbol{\theta}_g$ or to the likelihood in the case of μ_0, but also the implicit derivatives, since when the hyperparameters change, the optimum of the posterior $\hat{\boldsymbol{\phi}}$ also changes.

With this consideration, the derivative of the approximate log-evidence w.r.t. each hyperparameter in $\boldsymbol{\theta}_f$ can be expressed as

$$\begin{aligned}\frac{\partial \log q(\mathbf{y})}{\partial \theta_{f_j}} = \left.\frac{\partial \log q(\mathbf{y})}{\partial \theta_{f_j}}\right|_{\text{expl}} + \sum_{n=0}^{N} \frac{\partial \log q(\mathbf{y})}{\partial \hat{f}_n}\frac{\partial \hat{f}_n}{\theta_{f_j}} \\ + \sum_{n=0}^{N} \frac{\partial \log q(\mathbf{y})}{\partial \hat{g}_n}\frac{\partial \hat{g}_n}{\theta_{f_j}}\end{aligned} \quad (31)$$

Then, the explicit derivative is given by

$$\begin{aligned}\left.\frac{\partial \log q(\mathbf{y})}{\partial \theta_{f_j}}\right|_{\text{expl}} =& \frac{1}{2}\hat{\boldsymbol{\phi}}^T K^{-1}\frac{\partial K}{\partial \theta_{f_j}}K^{-1}\hat{\boldsymbol{\phi}} \\ &- \frac{1}{2}\text{tr}\left[(W^{-1}+K)^{-1}\frac{\partial K}{\partial \theta_{f_j}}\right]\end{aligned} \quad (32)$$

where tr$[M]$ is the trace of the square matrix M.

To calculate the implicit derivatives we have

$$\frac{\partial \log q(\mathbf{y})}{\partial \hat{f}_n} = -\frac{1}{2}\left[(K^{-1}+W)^{-1}\right]_{nn}\frac{\partial^3 \log p(y_n|\hat{\boldsymbol{\phi}}_n)}{\partial f_n^3} \quad (33)$$

which is equal to zero as the third derivative of the local likelihood w.r.t. to f_n is zero. For g_n the

3. The experiments have been conducted in a 4 GB computer with an Intel core i5 processor at 3.33 GHz using Matlab implementations for all the algorithms.

expression is the same than (33), but deriving on g_n. In this case, the third derivative of the local likelihood w.r.t. g_n is given by

$$\frac{\partial^3 \log p(y_n|\hat{\phi}_n)}{\partial g_n^3} = \begin{cases} \frac{2}{g_n'^3}, & \text{if } g_n' > 0 \\ 0, & \text{if } g_n' \leq 0 \end{cases} \tag{34}$$

with $g_n' = g_n + \mu_0$. Therefore, in contrast to the case of f_n, the implicit derivatives w.r.t. g_n do not vanish for positive g_n'. To compute $\partial \hat{f}_n / \theta_{f_j}$, we first calculate $\partial \hat{\phi} / \partial \theta_{f_j}$ and then select the corresponding terms to each g_n

$$\frac{\partial \hat{\phi}}{\partial \theta_{f_j}} = (I_{2N} + KW)^{-1} \frac{\partial K}{\partial \theta_{f_j}} \nabla \log p(\mathbf{y}|\hat{\phi}) \tag{35}$$

The calculations of the derivatives w.r.t. each hyperparameter in $\boldsymbol{\theta}_g$ are analogous to those for $\boldsymbol{\theta}_f$ hyperparameters.

The derivative w.r.t. the mean offset μ_0 for latent function g can be expressed

$$\frac{\partial \log q(\mathbf{y})}{\partial \mu_0} = \left. \frac{\partial \log q(\mathbf{y})}{\partial \mu_0} \right|_{\text{expl}} + \sum_{n=0}^{N} \frac{\partial \log q(\mathbf{y})}{\partial \hat{g}_n} \frac{\partial \hat{g}_n}{\partial \mu_0} \tag{36}$$

where the explicit derivative is calculated as

$$\left. \frac{\partial \log q(\mathbf{y})}{\partial \mu_0} \right|_{\text{expl}} = \sum_{n=0}^{N} \frac{\partial \log p(y_n|\hat{\phi}_n)}{\partial \mu_0} - \frac{1}{2} \frac{\partial \log |B|}{\partial \mu_0} \tag{37}$$

with

$$\frac{\partial \log p(y_n|\hat{\phi}_n)}{\partial \mu_0} = \begin{cases} \frac{1}{g_n'} - \frac{y_n^2 g_n' - y_n f_n}{c}, & g_n' > 0 \\ 0, & g_n' \leq 0 \end{cases} \tag{38}$$

and

$$-\frac{1}{2} \frac{\partial \log |B|}{\partial \mu_0} = -\frac{1}{2} \text{tr} \left(B^{-1} K \frac{\partial W}{\partial \mu_0} \right) \tag{39}$$

where $\partial W / \partial \mu_0$ is a $2N$ diagonal matrix with 0 in the first N elements of the diagonal and

$$\left[\frac{\partial W}{\partial \mu_0} \right]_{nn} = \begin{cases} \frac{2}{g_{n'}'^3}, & \text{if } g_{n'}' > 0 \\ 0, & \text{if } g_{n'}' \leq 0 \end{cases} \tag{40}$$

for $N < n \leq 2N$ with $n' = n - N$.

Finally, to calculate the implicit derivative, along with the expression for $\partial \log q(\mathbf{y}) / \partial \hat{g}_n$ calculated in (33) and (34), we have

$$\frac{\partial \hat{g}_n}{\partial \mu_0} = (I_{2N} + KW)^{-1} K \frac{\partial \nabla \log p(\mathbf{y}|\hat{\phi})}{\partial \mu_0} \tag{41}$$

where $\partial \nabla \log p(\mathbf{y}|\hat{\phi}) / \partial \mu_0 = -\text{diag}(W)$.

Acknowledgments

We would like to thank the reviewers for their valuable comments to improve the quality of the manuscript and for proposing some interesting extensions of this work that have been included in the conclusions.

References

[1] C. K. I. Williams and C. E. Rasmussen, "Gaussian processes for regression," in *Advances in Neural Information Processing Systems*, D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, Eds. 1996, vol. 8, pp. 514–520, MIT Press, Cambridge, MA.

[2] J. Quiñonero-Candela and C.E. Rasmussen, "A unifying view of sparse approximate Gaussian process regression," *J. Machine Learning Res.*, vol. 6, pp. 1939–1959, 2005.

[3] E. Snelson and Z. Ghahramani, "Sparse Gaussian processes using pseudo-inputs," in *Advances in Neural Information Processing Systems*, Y. Weiss, B. Schölkopf, and J. C. Platt, Eds., vol. 18, pp. 1257–1264. MIT Press, Cambridge, MA, 2006.

[4] M. Lázaro-Gredilla, J. Quiñonero-Candela, C.E. Rasmussen, and A.R. Figueiras-Vidal, "Sparse spectrum Gaussian process regression," *J. Machine Learning Res.*, vol. 11, pp. 1865–1881, 2010.

[5] C.E. Rasmussen and C.K.I. Williams, *Gaussian Processes for Machine Learning*, The MIT Press, Cambridge, MA, 2006.

[6] P.W. Goldberg, C.K.I. Williams, and C.M. Bishop, "Regression with input-dependent noise: A Gaussian process treatment," in *Advances in Neural Information Processing Systems*, M.I. Jordan, M.J. Kearns, and S.A. Solla, Eds. 1998, vol. 10, pp. 493–499, The MIT Press, Cambridge, MA.

[7] K. Kersting, C. Plagemann, P. Pfaff, and W. Burgard, "Most likely heteroscedastic Gaussian process regression," in *Proc. 24th Int. Conf. on Machine Learning*, Ghahramani Z., Ed. 2007, pp. 393–400, Omnipress.

[8] N. Quadrianto, K. Kersting, M.D. Reid, T.S. Caetano, and W.L. Buntine, "Kernel conditional quantile estimation via reduction revisited," in *Proc. 9th Int. Conf. on Data Mining*. 2009, pp. 938–943, IEEE.

[9] L. Muñoz-González, M. Lázaro-Gredilla, and A.R. Figueiras-Vidal, "Heteroscedastic Gaussian process regression using expectation propagation," in *Proc. Int. Workshop on Machine Learning for Signal Processing*, T. Tan, S. Katagiri, J. Tao, A. Nakamura, and J. Larsen, Eds. 2011, pp. 1–6, IEEE.

[10] M. Lázaro-Gredilla and M.K. Titsias, "Variational heteroscedastic Gaussian process regression," in *Proc. 28th Int. Conf. on Machine Learning*, L. Getoor and T. Scheffer, Eds. 2011, pp. 841–848, ACM, New York, NY.

[11] Q.V. Le, A.J. Smola, and S. Canu, "Heteroscedastic Gaussian process regression," in *Proc. 22nd Int. Conf. on Machine Learning*. 2005, pp. 489–496, ACM, New York, NY.

[12] R.P. Adams and O. Stegle, "Gaussian process product models for nonparametric nonstationarity," in *Proc. 25th Int. Conf. on Machine learning*, A. McCallum and S. Roweis, Eds. 2008, pp. 1–8, Omnipress.

[13] L. Muñoz-González, M. Lázaro-Gredilla, and A. R. Figueiras-Vidal, "Divisive Gaussian processes for nonstationary regression," *IEEE Trans. on Neural Networks and Learning Systems*, vol. 25, no. 11, pp. 1991–2003, 2014.

[14] I. Murray, R.P. Adams, and D.J.C. MacKay, "Elliptical slice sampling," in *Int. Conf. on Artificial Intelligence and Statistics 13*, Sardinia, Italy, 2010, vol. 9 of J. Machine Learning Res.: Workshop and Conf. Proc., pp. 541–548.

[15] M.W. Seeger, "Bayesian inference and optimal design for the sparse linear model," *J. Machine Learning Res.*, vol. 9, pp. 759–813, 2008.

[16] D.V. Hinkley, "On the ratio of two correlated normal random variables," *Biometrika*, vol. 56, no. 3, pp. 635–639, 1969.

[17] M. Yuan and G. Wahba, "Doubly penalized likelihood estimator in heteroscedastic regression," *Statistics and Probability Letters*, vol. 69, pp. 11–20, 2004.

[18] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik, "Support vector regression machines," in *Advances in Neural Information Processing Systems*, M. C. Mozer, M. I. Jordan, and T. Petsche, Eds. 1997, vol. 9, pp. 155–161, Morgan Kaufmann, San Mateo, CA.

[19] T. Hastie, R. Tibshirani, and J.H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, New York, NY, 2009.

[20] K. Bache and M. Lichman, "UCI Machine Learning Repository, http://archive.ics.uci.edu/ml. School of Information and Computer Sciences, University of California at Irvine, CA," 2015.