Nonstationary Gaussian Process Regression using Point Estimates of Local Smoothness

Christian Plagemann[1], Kristian Kersting[2], and Wolfram Burgard[1]

[1] University of Freiburg, Georges-Koehler-Allee 79, 79110 Freiburg, Germany
{plagem,burgard}@informatik.uni-freiburg.de
[2] Fraunhofer Institute IAIS, Sankt Augustin, Germany
kristian.kersting@iais.fraunhofer.de

Abstract. Gaussian processes using nonstationary covariance functions are a powerful tool for Bayesian regression with input-dependent smoothness. A common approach is to model the local smoothness by a latent process that is integrated over using Markov chain Monte Carlo approaches. In this paper, we demonstrate that an approximation that uses the estimated mean of the local smoothness yields good results and allows one to employ efficient gradient-based optimization techniques for jointly learning the parameters of the latent and the observed processes. Extensive experiments on both synthetic and real-world data, including challenging problems in robotics, show the relevance and feasibility of our approach.

1 Introduction

Gaussian processes (GPs) have emerged as a powerful yet practical tool for solving various machine learning problems such as nonlinear regression or multi-class classification [16]. As opposed to making parametric assumptions about the underlying functions, GP models are directly expressed in terms of the training data and thus belong to the so called nonparametric methods. Their increasing popularity is due to the fact that nonlinear problems can be solved in a principled Bayesian framework for learning, model selection, and density estimation while the basic model just requires relatively simple linear algebra. A common assumption when specifying a GP prior is stationarity, i.e., that the covariance between function values $f(\mathbf{x})$ and $f(\mathbf{x}')$ only depends on the distance $\|\mathbf{x} - \mathbf{x}'\|$ between the indexes and not on their locations directly. Consequently, standard GPs lack the ability to adapt to variable smoothness in the function of interest.

Modeling input-dependent smoothness, however, is essential in many fundamental problems in the geo-sciences, mobility mining, activity recognition, and robotics, among other areas. Consider, for example, the problem of modeling terrain surfaces given sets of noisy elevation measurements. Accurately modeling such data requires the ability to deal with a varying data density and to balance smoothing against the preservation of discontinuities. Discontinuities arise for instance at steps, stairs, curbs, or at walls. These features are important for planning paths of mobile robots, for estimating traversability, and in terrain segmentation tasks. Accordingly, the ability to flexibly adapt a regression model to the local properties of the data may greatly enhance the applicability of such techniques.

In the past, several approaches for specifying nonstationary GP models haven been proposed in the literature [12, 13]. A particularly promising approach is due to Paciorek and Schervish [8] who proposed to explicitly model input-depending smoothness using additional, latent GPs. This approach (a) provides the user with a continuous latent space of local kernels, (b) allows the user to analyze the estimated covariance function yielding important insights into the problem domain, and (c) fully stays in the GP framework so that computational methods for speeding up GP inference and fitting can be adapted.

Paciorek and Schervish provide a flexible and general framework based on MCMC integration, which unfortunately – as also noted by the authors – is computationally demanding for large data sets and which is thus not feasible in the real world situations that are typically encountered in robotics and engineering tasks. In this paper, we present a simple approximation that does not integrate over all latent values but uses the predicted mean values only. Specifically, we parameterize the nonstationary covariances using a second GP with m latent length-scales. Assuming $m \ll n$, where n is the number of training points, this results in a nonstationary GP regression method with practically no overhead compared to standard GPs. More importantly, using point estimates naturally leads to gradient-based techniques for efficiently learning the parameters of both processes jointly, which is the main contribution of this paper.

We present experiments carried out on on synthetic and real-world data sets from challenging application domains such as robotics and embedded systems showing the relevance and feasibility of our approach. More specifically, our nonstationary GP approach significantly outperforms standard GPs in terms of prediction accuracy, while it is significantly faster then [8]. We regard these empirical results as an additional substantial contribution of this paper as they tighten the link between advanced regression techniques based on GPs and application domains such as robotics and embedded systems. To the best of our knowledge, it is the first time that nonstationary GPs have been learned in a principled way in these challenging domains.

This paper is organized as follows. After reviewing related work, we introduce nonstationary Gaussian processes regression and how to make predictions in Section 3. In Section 4, we then show how to learn the hyperparameters using gradient-based methods. Before concluding, we demonstrate the feasibility and relevance of our approach in an extensive set of experiments.

2 Related Work

Gaussian process models [11] have the advantage of providing predictive uncertainties for their estimates while not requiring, for instance, a fixed discretization of the space. This has led to their successful application in a wide range of application areas including robotics and ubiquitous computing. For example, Schwaighofer *et al.* [14] applied the model for realizing positioning systems using cellular networks. GPs have been proposed as measurement models [1] and for model-based failure detection [10] in robotics because they naturally deal with noisy measurements, unevenly distributed observations, and fill small gaps in the data with high confidence while assigning higher predictive uncertainty in sparsely sampled areas. Many robotics applications, however, call for non-standard GP models. Kersting *et al.* [5], for example, have shown that heteroscedastic GP regression, i.e., regression with input-dependent noise outperforms state-of-the-art approaches in mobile robot localization. Whereas they also use a GP prior to model local noise rates, they do not estimate the hyperparameters jointly using gradient-based optimization but alternate each process in a sampling-based EM fashion. Lang *et al.* [6] modeled 3D terrain data using nonstationary GPs by following the approach of Paciorek and Schervish [8]. They derived a specific adaptation procedure that also does not require MCMC integration as originally proposed by Paciorek and Schervish, but that is not derived from first principles. Another approach to modeling nonstationarity is to use ensembles of GPs, where every GP is assigned to a specific region, an idea akin to GP mixture models such as presented by Williams' [16]. A related idea has also been proposed by Pfingsten *et al.* [9]. Cornford *et al.* [3] model straight discontinuities in wind fields by placing auxiliary GPs along the edge on both sides of the discontinuity. They are then used to learn GPs representing the process on either side of the discontinuity.

Apart from Paciorek and Schervish's [8] (see also the references in there) approach of directly modeling the covariance function using additional latent GPs, several other approaches for specifying nonstationary GP models can be found in the literature. For instance, Sampson and Guttorp [12] map a nonstationary spatial process (not based on GPs) into a latent space, in which the problem becomes approximately stationary. Schmidt and O'Hagan [13] followed this idea and used GPs to implement the mapping. Similar in spirit, Pfingsten *et al.* [9] proposed to augment the input space by a latent extra input to tear apart regions of the input space that are separated by abrupt changes of the function values. All GP approaches proposed so far, however, followed a Markov chain Monte Carlo approach to inference and learning. Instead, we present a novel maximum-a-posterior treatment of Paciorek and Schervish's approach that fully stays in the GP framework, explicitly models the covariance function, provides continuous estimates of the local kernels, and that naturally allows for gradient-based joint optimization of its parameters.

3 Nonstationary Gaussian Process Regression

The nonlinear regression problem is to recover a functional dependency $y_i = f(\mathbf{x}_i) + \epsilon_i$ from n observed data points $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$, where $y_i \in \mathbb{R}$ are the noisy observed outputs at input locations $\mathbf{x}_i \in \mathbb{R}^d$. Throughout this paper we will also use $\mathbf{X} \in \mathbb{R}^{n \times d}$ to refer to all input locations. For the sake of simplicity, we will concentrate on one-dimensional outputs, but all formulas naturally gen-

Table 1. Notation used to derive the gradient of the model selection criterion w.r.t. the joint hyperparameters $\boldsymbol{\theta}$ of the nonstationary GP.

Observed GP	$\mathcal{GP}_y$
Hyperparameters of $\mathcal{GP}_y$	$\boldsymbol{\theta}_y = \langle \sigma_f, \sigma_n \rangle$
Training set	$\mathcal{D} = \langle \mathbf{X}, \mathbf{y} \rangle, \mathbf{X} \in \mathbb{R}^{n \times d}, \mathbf{y} \in \mathbb{R}^n$
Prediction	$y^* \in \mathbb{R}$ at location $\mathbf{X}^* \in \mathbb{R}^{1 \times d}$
Latent length-scale process	$\mathcal{GP}_\ell$
Latent length-scale support values	$\overline{\boldsymbol{\ell}} \in \mathbb{R}^m$ at locations $\overline{\mathbf{X}} \in \mathbb{R}^{m \times d}$
Latent length-scales at training points of $\mathcal{GP}_y$	$\boldsymbol{\ell} \in \mathbb{R}^n$ at locations $\mathbf{X}$
Latent length-scale at test point	$\ell^* \in \mathbb{R}$ at location $\mathbf{X}^*$
Hyperparameters of $\mathcal{GP}_\ell$	$\boldsymbol{\theta}_\ell = \langle \overline{\sigma}_f, \overline{\sigma}_\ell, \overline{\sigma}_n \rangle$
Joint hyperparameters	$\boldsymbol{\theta} = \langle \boldsymbol{\theta}_y, \boldsymbol{\theta}_\ell, \overline{\boldsymbol{\ell}} \rangle = \langle \sigma_f, \sigma_n, \overline{\sigma}_f, \overline{\sigma}_\ell, \overline{\sigma}_n, \overline{\boldsymbol{\ell}} \rangle$

eralize to the multidimensional case. The regression task is to learn a model for $p(y^*|\mathbf{x}^*, \mathcal{D})$, i.e., the predictive distribution of new target values y^* at $\mathbf{x}^*$ given $\mathcal{D}$. The notation we will use is listed in Table 1.

Stationary Gaussian Process Regression: In the standard Gaussian process model for regression (STD-GP), we assume independent, normally distributed noise terms $\epsilon_i \sim \mathcal{N}(0, \sigma_n^2)$ with a constant noise variance parameter σ_n^2. The central idea is to model every finite set of samples y_i from the process as *jointly* Gaussian distributed, such that the predictive distribution $p(y^*|\mathbf{x}^*, \mathcal{D})$ at arbitrary query points $\mathbf{x}^*$ is a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ with mean

$$\mu = \mathbf{k}_{\mathbf{x}^*,\mathbf{x}}^T (\mathbf{K}_{\mathbf{x},\mathbf{x}} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{y} \tag{1}$$

and variance

$$\sigma^2 = k(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{k}_{\mathbf{x}^*,\mathbf{x}}^T (\mathbf{K}_{\mathbf{x},\mathbf{x}} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{k}_{\mathbf{x}^*,\mathbf{x}} + \sigma_n^2 \ . \tag{2}$$

Here, we have $\mathbf{K}_{\mathbf{x},\mathbf{x}} \in \mathbb{R}^{n \times n}$ with $\mathbf{K}_{\mathbf{x},\mathbf{x}}(i,j) = k(\mathbf{x}_i, \mathbf{x}_j)$, $\mathbf{k}_{\mathbf{x}^*,\mathbf{x}} \in \mathbb{R}^n$ with $\mathbf{k}_{\mathbf{x}^*,\mathbf{x}}(i) = k(\mathbf{x}^*, \mathbf{x}_i)$, $\mathbf{y} = (y_1, \ldots, y_n)^T$, and $\mathbf{I}$ the identity matrix.

An integral part of GP regression is the covariance function $k(\cdot, \cdot)$, which specifies the covariance of the corresponding targets (see [11] for more details). A common choice is the squared exponential covariance function $k_{se}(\mathbf{x}, \mathbf{x}') = \sigma_f^2 \exp\left(-1/2 \cdot (s(\mathbf{x}, \mathbf{x}')/\sigma_\ell)^2\right)$ with $s(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|$. The term σ_f denotes the amplitude (or signal variance) and σ_ℓ is the characteristic length-scale. The parameters $\boldsymbol{\theta}_y = (\sigma_f, \sigma_\ell, \sigma_n)$ are called the *hyperparameters* of the process. Note that we – as opposed to some other authors – also treat the global noise rate σ_n^2 as a hyperparameter for ease of notation.

Modeling Input-Dependent Smoothness: A limitation of the standard GP framework as described above is the assumption of constant length-scales σ_ℓ over the whole input space. Intuitively, length-scales define the extent of the area

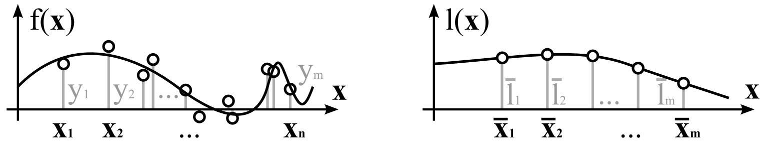

Fig. 1. Placing a GP prior over the latent length-scales for nonstationary GP regression. An observed Gaussian process $\mathcal{GP}_y$ is sketched on left-hand side and the latent $\mathcal{GP}_\ell$ governing the local length-scales is shown on the right-hand side.

in which observations strongly influence each other. For 3D terrain modeling, for instance, within the context of mobile robot localization, one would like to use locally varying length-scales to account for the different situations. For example in flat plains, the terrain elevations are strongly correlated over long distances. In high-variance, "wiggly" terrain, on the other hand and at strong discontinuities, the terrain elevations are correlated over very short distances only, if at all. To address this problem of varying correlation scale, an extension of the squared exponential (SE) covariance function was proposed by Paciorek and Schervish [8], which takes the form

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \, |\Sigma_i|^{\frac{1}{4}} \, |\Sigma_j|^{\frac{1}{4}} \left| \frac{\Sigma_i + \Sigma_j}{2} \right|^{-\frac{1}{2}} \cdot \exp\left[-\mathbf{d}_{ij}^T \left(\frac{\Sigma_i + \Sigma_j}{2} \right)^{-1} \mathbf{d}_{ij} \right] , \tag{3}$$

where $\mathbf{d}_{ij} = (\mathbf{x}_i - \mathbf{x}_j)$. The intuition is that each input location $\mathbf{x}'$ is assigned a local Gaussian kernel matrix Σ' and the covariance between two targets y_i and y_j is calculated by averaging between the two local kernels at the input locations $\mathbf{x}_i$ and $\mathbf{x}_j$. In this way, the local characteristics at both locations influence the modeled covariance of the corresponding target values. For the sake of simplicity, we consider the isotropic case only in this paper. The general case can be treated in the same way. In the isotropic case, where the eigenvectors of the local kernels are aligned to the coordinate axes and their eigenvalues are equal, the matrices $\mathbf{\Sigma}_i$ simplify to $\ell_i^2 \cdot \mathbf{I}_n$ with a real-valued length-scale parameter ℓ_i. In the one-dimensional case, for instance, Eq. (3) then simplifies to

$$k(x_i, x_j) = \sigma_f^2 \cdot (\ell_i^2)^{\frac{1}{4}} \cdot (\ell_j^2)^{\frac{1}{4}} \cdot \left(\frac{1}{2}\ell_i^2 + \frac{1}{2}\ell_j^2 \right)^{-\frac{1}{2}} \cdot \exp\left[-\frac{(x_i - x_j)^2}{\frac{1}{2}\ell_i^2 + \frac{1}{2}\ell_j^2} \right] . \tag{4}$$

We do not specify a functional form for the length-scale $\ell(x)$ at location x but place a GP prior over them. More precisely, an independent GP is used to model the logarithms $\log(\ell(x))$ of the length-scales, to avoid negative values. This process, denoted as $\mathcal{GP}_\ell$ is governed by a different covariance function specified by the hyperparameters $\boldsymbol{\theta}_\ell = \langle \overline{\sigma}_f, \overline{\sigma}_\ell, \overline{\sigma}_n \rangle$. Additionally, we have to maintain the set of m support values $\overline{\boldsymbol{\ell}}$ as part of $\boldsymbol{\theta}$ as depicted in Figure 1.

Making Predictions: In the extended model, we now have to integrate over all possible latent length-scales to get the predictive distribution

$$p(y^*|\mathbf{X}^*,\mathcal{D},\boldsymbol{\theta}) = \iint p(y^*|\mathbf{X}^*,\mathcal{D},\exp(\ell^*),\exp(\boldsymbol{\ell}),\boldsymbol{\theta}_y)\cdot p(\ell^*,\boldsymbol{\ell}|\mathbf{X}^*,\mathbf{X},\overline{\boldsymbol{\ell}},\overline{\mathbf{X}},\boldsymbol{\theta}_\ell)\, d\boldsymbol{\ell}\, d\ell^*$$

of a regressand y^* at location $\mathbf{X}^*$ given a dataset $\mathcal{D}$ and hyperparameters $\boldsymbol{\theta}$ (Note that we explicitly highlight here that $\mathcal{GP}_\ell$ is defined over the log length-scales). Because this marginalization is intractable, [8] apply MCMC to approximate it. Instead, we seek for the solution using the most probable length-scale estimates, i.e., $p(y^*|\mathbf{X}^*,\mathcal{D},\boldsymbol{\theta}) \approx p(y^*|\mathbf{X}^*,\exp(\ell^*),\exp(\boldsymbol{\ell}),\mathcal{D},\boldsymbol{\theta}_y)$ where $(\ell^*,\boldsymbol{\ell})$ are the mean predictions of the length-scale process at $\mathbf{X}^*$ and the locations in $\mathcal{D}$. Since the length-scales are independent latent variables in the combined regression model, making predictions amounts to making two standard GP predictions using Eqs. (1) and (2), one using $\mathcal{GP}_\ell$ to get $(\ell^*,\boldsymbol{\ell})$ and one using $\mathcal{GP}_y$ with $(\ell^*,\boldsymbol{\ell})$ treated as fixed parameters.

4 Learning Hyperparameters

So far, we have described our model assuming that we have the joint hyperparameters $\boldsymbol{\theta}$ of the overall process. In practice, we are unlikely to have these parameters a-priori and, instead, we wish to estimate them from observations $\mathbf{y}$.

Assume a given set of n observations $\mathbf{y}$ at locations $\mathbf{X}$. We seek to find those hyperparameters that maximize the probability of observing $\mathbf{y}$ at $\mathbf{X}$, i.e., we seek to maximize $p(\mathbf{y}|\mathbf{X},\boldsymbol{\theta}) = \int p(\mathbf{y}|\mathbf{X},\boldsymbol{\ell},\boldsymbol{\theta}_y)\cdot p(\boldsymbol{\ell}|\mathbf{X},\overline{\boldsymbol{\ell}},\overline{\mathbf{X}},\boldsymbol{\theta}_\ell)\; d\boldsymbol{\ell}$. As for making predictions, such a marginalization is intractable. Instead, we seek to make progress by seeking a solution that maximizes the a-posteriori probability of the latent length-scales

$$\log p(\boldsymbol{\ell}|\mathbf{y},\mathbf{X},\boldsymbol{\theta}) = \log p(\mathbf{y}|\mathbf{X},\exp(\boldsymbol{\ell}),\boldsymbol{\theta}_y) + \log p(\boldsymbol{\ell}|\mathbf{X},\overline{\boldsymbol{\ell}},\overline{\mathbf{X}},\boldsymbol{\theta}_\ell) + \text{const.}\,, \qquad (5)$$

where, again, the $\boldsymbol{\ell}$ are the mean predictions of $\mathcal{GP}_\ell$. The gradient of this objective function w.r.t. to hyperparameters $\boldsymbol{\theta}$ or a subset of them can be employed within gradient-based optimization to find the corresponding solution. In our experiments, we optimized $\overline{\sigma}_f$, $\overline{\sigma}_n$, and $\overline{\sigma}_\ell$ of the latent kernel width process in an outer cross-validation loop on an independent validation set and assumed $\partial L(\boldsymbol{\theta})/\partial\bullet = 0$, where $\bullet$ denotes one of them, within the inner gradient optimization. The locations $\overline{\mathbf{X}}$ of the latent kernel width variables were sampled uniformly on the bounding rectangle given by $\mathbf{X}$.

In the following, we will detail the objective function and the gradient of it with respect to the hyperparameter.

4.1 The Objective Function

We maximize the marginal likelihood (5) of the data with respect to the joint hyperparameters as well as the support values $\overline{\boldsymbol{\ell}}$ of the length-scale process.

The first term in this equation is the standard objective function for Gaussian processes

$$\log p(\mathbf{y}|\mathbf{X}, \exp(\boldsymbol{\ell}), \boldsymbol{\theta}_y) = -\frac{1}{2}\mathbf{y}^T(\mathbf{K}_{\mathbf{x},\mathbf{x}}+\sigma_n^2\mathbf{I})^{-1}\mathbf{y} - \frac{1}{2}\log|\mathbf{K}_{\mathbf{x},\mathbf{x}}+\sigma_n^2\mathbf{I}| - \frac{n}{2}\log(2\pi) \ ,$$

where $|\mathbf{M}|$ denotes the determinant of a matrix $\mathbf{M}$ and $\mathbf{K}_{\mathbf{x},\mathbf{x}}$ stands for the noise-free nonstationary covariance matrix for the training locations $\mathbf{X}$ that will be detailed below. Our point estimate approach considers the most likely latent length-scales $\boldsymbol{\ell}$, i.e. the mean predictions of $\mathcal{GP}_\ell$ at locations $\mathbf{X}$. Thus, the second term of Eq. (5) has the form

$$\log p(\boldsymbol{\ell}|\mathbf{X}, \overline{\boldsymbol{\ell}}, \overline{\mathbf{X}}, \boldsymbol{\theta}_\ell) = -\frac{1}{2}\log|\mathbf{K}_{\overline{\mathbf{x}},\overline{\mathbf{x}}} + \overline{\sigma}_n^2\mathbf{I}| - \frac{n}{2}\log(2\pi) \ .$$

Putting both together, we get the objective function

$$L(\boldsymbol{\theta}) = \log p(\boldsymbol{\ell}|\mathbf{y}, X, \boldsymbol{\theta}) = c_1 + c_2 \cdot \left[\mathbf{y}^T\mathbf{A}^{-1}\mathbf{y} + \log|\mathbf{A}| + \log|\mathbf{B}|\right] \ , \qquad (6)$$

where c_1 and c_2 are real-valued constants, and $\mathbf{A} := \mathbf{K}_{\mathbf{x},\mathbf{x}} + \sigma_n^2\mathbf{I}$ and $\mathbf{B} := \mathbf{K}_{\overline{\mathbf{x}},\overline{\mathbf{x}}} + \overline{\sigma}_n^2\mathbf{I}$ are covariance matrices. The noise-free part of the nonstationary covariance matrix $\mathbf{K}_{\mathbf{x},\mathbf{x}}$ is calculated according to Eq. (3). As mentioned above, we consider the isotropic case only for the sake of simplicity. We express Eq. (4) for the case of multivariate inputs $\mathbf{x}_i$ using the compact matrix-vector notation suggested in [2]. Recalling that $\boldsymbol{\ell}$ represents the local length-scales at the training locations $\mathbf{X}$, we get

$$\mathbf{K}_{\mathbf{x},\mathbf{x}} = \sigma_f^2 \cdot {\mathbf{P_r}}^{\frac{1}{4}} \circ {\mathbf{P_c}}^{\frac{1}{4}} \circ (1/2)^{-\frac{1}{2}}\, {\mathbf{P_s}}^{-\frac{1}{2}} \circ \mathbf{E} \qquad (7)$$

with

$$\begin{aligned} &\mathbf{P_r} = \mathbf{p}\cdot\mathbf{1}_n^T \ , && \mathbf{P_c} = \mathbf{1}_n^T\cdot\mathbf{p}^T \ , && \mathbf{p} = \boldsymbol{\ell}^T\boldsymbol{\ell} \ , \\ &\mathbf{P_s} = \mathbf{P_r} + \mathbf{P_c} \ , && \mathbf{E} = \exp[-s(\mathbf{X}) \div \mathbf{P_s}] \ , && \boldsymbol{\ell} = \exp\left[\overline{\mathbf{K}}_{\mathbf{x},\overline{\mathbf{x}}}^T\left[\overline{\mathbf{K}}_{\overline{\mathbf{x}},\overline{\mathbf{x}}} + \overline{\sigma}_n^2\mathbf{I}\right]^{-1}\overline{\boldsymbol{\ell}}\right] . \end{aligned}$$

Note that $\mathbf{p} \in \mathbb{R}^n$ and, thus, $\mathbf{P_r}$ and $\mathbf{P_c}$ are matrices built using the outer vector product. Here, $s(\mathbf{X})$ calculates the $n \times n$ matrix of squared distances between the input vectors $\mathbf{x}$ contained in $\mathbf{X}$. $\circ$ and $\div$ denote element-wise multiplication and division respectively and matrix exponentiation $\mathbf{M}^\alpha$ is also defined element-wise for $\alpha \neq -1$. In the same notation, the covariance function for the latent length-scale process $\mathcal{GP}_\ell$ becomes (in the stationary squared exponential form)

$$\mathbf{K}_{\overline{\mathbf{x}},\overline{\mathbf{x}}} = {\overline{\sigma}_f}^2 \cdot \exp\left[-\frac{1}{2}s({\overline{\sigma_\ell}}^{-2}\overline{\mathbf{X}})\right]$$

and, analogously, for making predictions within $\mathcal{GP}_\ell$

$$\mathbf{K}_{\mathbf{x},\overline{\mathbf{x}}} = {\overline{\sigma}_f}^2 \cdot \exp\left[-\frac{1}{2}s({\overline{\sigma_\ell}}^{-2}\mathbf{X}, {\overline{\sigma_\ell}}^{-2}\overline{\mathbf{X}})\right] \ .$$

4.2 The Gradient

Using standard results from matrix calculus, the partial derivative of the objective (6) w.r.t. an element $\bullet$ of $\boldsymbol{\theta}$ turns out to be

$$\frac{\partial L(\boldsymbol{\theta})}{\partial \bullet} = -\mathbf{y}^T \mathbf{A}^{-1} \frac{\partial \mathbf{A}}{\partial \bullet} \mathbf{A}^{-1} \mathbf{y} + \mathrm{tr}(\mathbf{A}^{-1} \frac{\partial \mathbf{A}}{\partial \bullet}) + \mathrm{tr}(\mathbf{B}^{-1} \frac{\partial \mathbf{B}}{\partial \bullet}) \ , \tag{8}$$

where $\mathrm{tr}(\mathbf{M})$ is the trace of a matrix $\mathbf{M}$. For the two hyperparameters of $\mathcal{GP}_y$ we get the straight-forward results

$$\tfrac{\partial \mathbf{A}}{\partial \sigma_n} = 2\sigma_n \mathbf{I} \, , \qquad \tfrac{\partial \mathbf{B}}{\partial \sigma_n} = 0 \, , \qquad \tfrac{\partial \mathbf{A}}{\partial \sigma_f} = 2\sigma_f \mathbf{K}_{\mathbf{x},\mathbf{x}} \ , \qquad \tfrac{\partial \mathbf{B}}{\partial \sigma_f} = 0 \ .$$

The case $\bullet = \bar{\boldsymbol{\ell}}$ yields $(\partial \mathbf{B} / \partial \bar{\boldsymbol{\ell}}) = 0$ and $(\partial \mathbf{A})/(\partial \bar{\boldsymbol{\ell}}) = (\partial \mathbf{K}_{\mathbf{x},\mathbf{x}})/(\partial \bar{\boldsymbol{\ell}}) =$

$$\sigma_f^2 \ (1/2)^{-\frac{1}{2}} \cdot \left[\left(\frac{\partial (\mathbf{P_r}^{\frac{1}{4}})}{\partial \bar{\boldsymbol{\ell}}} \circ \mathbf{P_c}^{\frac{1}{4}} \circ \mathbf{P_s}^{-\frac{1}{2}} \circ \mathbf{E} \right) + \left(\mathbf{P_r}^{\frac{1}{4}} \circ \frac{\partial (\mathbf{P_c}^{\frac{1}{4}})}{\partial \bar{\boldsymbol{\ell}}} \circ \mathbf{P_s}^{-\frac{1}{2}} \circ \mathbf{E} \right) + \right.$$
$$\left. \left(\mathbf{P_r}^{\frac{1}{4}} \circ \mathbf{P_c}^{\frac{1}{4}} \circ \frac{\partial (\mathbf{P_s}^{-\frac{1}{2}})}{\partial \bar{\boldsymbol{\ell}}} \circ \mathbf{E} \right) + \left(\mathbf{P_r}^{\frac{1}{4}} \circ \mathbf{P_c}^{\frac{1}{4}} \circ \mathbf{P_s}^{-\frac{1}{2}} \circ \frac{\partial (\mathbf{E})}{\partial \bar{\boldsymbol{\ell}}} \right) \right] \ .$$

The remaining simplifications can be achieved by substitution with the definitions given after Eq. (7) and by applying general rules for differentiation such as the chain rule

$$\frac{\partial f(g(\mathbf{X}))}{\partial \mathbf{x}} = \frac{\partial (f(\mathbf{U}) \colon)}{\partial \mathbf{U}} \cdot \frac{\partial g(\mathbf{x})}{\partial \mathbf{x}} \quad \Bigg|_{\mathbf{U}=g(\mathbf{X})}$$

where $\mathbf{X}\colon$ denotes the vectorization of a matrix by stacking its columns, e.g., as applied to a term containing element-wise division

$$\frac{\partial (\mathbf{A} \div \mathbf{B})}{\partial \mathbf{x}} = \mathbf{A} \circ \frac{\partial \ \mathrm{inv}(\mathbf{U})\colon}{\partial \mathbf{U}\colon} \cdot \frac{\partial \mathbf{B}\colon}{\partial \mathbf{x}} \quad \Bigg|_{\mathbf{U}=\mathbf{B}}$$

for a matrix $\mathbf{A}$ that does not depend on $\mathbf{x}$. Substituting the resulting partial derivatives in Eq. (8) yields the gradient $\partial L(\boldsymbol{\theta})/\partial \boldsymbol{\theta}$, which can be used in gradient-based optimization techniques, such as Møller's [7] scaled conjugate gradients (SCG), to jointly optimize the hyperparameters of $\mathcal{GP}_y$ and $\mathcal{GP}_\ell$.

5 Experiments

The goal of our experimental evaluation was to investigate to which extent the point estimate approach to nonstationary GP regression is able to handle input-dependent smoothness and to quantify the gains relative to the stationary model. Specifically, we designed several experiments to investigate whether the approach can solve standard regression problems from the literature. We also applied it to two hard and relevant regression problems from embedded systems and robotics. On the two standard test sets, we demonstrate that the prediction accuracy of our approach is comparable to the one achieved by the MCMC-based method by Paciorek and Schervish [8], which, compared to our algorithm, is substantially more demanding regarding the computational resources.

We have implemented and evaluated our approach in Matlab. Using the compact matrix notation for all derivations, the core algorithm is implemented in less than 150 lines of code and, more importantly, advanced optimization strategies like sparse matrix approximations or parallelization can be realized with virtually no additional implementation efforts. As optimization procedure, we applied Møller's scaled conjugate gradient (SCG) [7] approach. In all our experiments, the SCG converged after at most 20 iterations. To quantitatively evaluate the performance of our nonstationary regression technique, we ran 30 to 50 independent test runs for each of the following test cases. Each run consisted of (a) randomly selecting or generating training data, (b) fitting the nonstationary model, and (c) evaluating the predictive distribution of the learned model at independent test locations. The latter was done either using the known ground truth function values or by assessing the likelihood of independent observations in the cases in which the ground truth was not known (e.g., for the RFID and terrain mapping experiments).

In all test scenarios, we evaluate the accuracy of the mean predictions and also the fit of the whole predictive distribution using the **standardized mean squared error**

$$\mathrm{sMSE} := n^{-1} \sum\nolimits_{i=1}^{n} \mathrm{var}(y)^{-1} (y_i - m_i)^2$$

and the **negative log predictive density**

$$\mathrm{NLPD} := n^{-1} \sum\nolimits_{i=1}^{n} \log p_{\mathrm{model}}(y_i | \mathbf{x}_i)$$

respectively. Here, $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ denotes the test data set, $p_{\mathrm{model}}(\cdot|\mathbf{x}_i)$ stands for the predictive distribution at location $\mathbf{x}_i$, and $m_i := \mathbb{E}[p_{\mathrm{model}}(\cdot|\mathbf{x}_i)]$ denotes the predicted mean. Statistical significance was assessed using two-sample t-tests with 95% confidence intervals.

All experiments were conducted using Matlab on a Linux desktop PC with a single 2 GHz CPU. The typical runtime for fitting the full nonstationary model to 100 training points was in the order of 50 seconds. The runtime requirements of the MCMC-based approach [8] which does not employ any gradient information were reported to be in the order of hours for a C-implementation on standard hardware in year 2004. In the following, we term our nonstationary approach as LA-GP (Locally Adaptive GP), the standard model employing the isotropic, squared exponential covariance function as STD-GP and Paciorek and Schervish's MCMC-based approach as NS-GP (Nonstationary GP).

5.1 Simulation Results in 1D and 2D

First, we verified that our approach accurately solves standard regression problems described in the literature. To this aim, we considered the two simulated

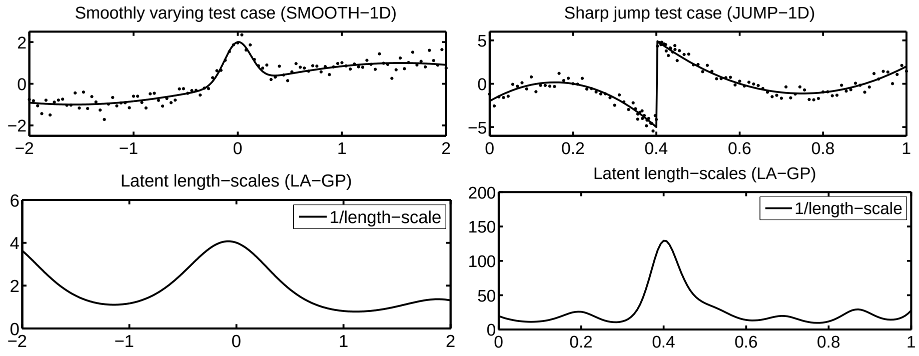

Fig. 2. Two standard nonstationary test cases SMOOTH-1D (top left) and JUMP-1D (top right) that were used for evaluation purposes in previous work [4] and [8]. The lower two plots give the inverse latent length-scales as optimized by our approach. Higher values in these plots indicate a larger local frequency.

functions shown in Figure 2. Both functions were also used for evaluation purposes by Dimatteo *et al.* [4] and in [8]. In the remainder, these test scenarios will be referred to as SMOOTH-1D and JUMP-1D. Whereas SMOOTH-1D is a smoothly varying function with a substantial "bump" close to 0, JUMP-1D has a sharp jump at 0.4. For SMOOTH-1D, we sampled 101 training points and 400 test points from the interval $(-2, 2)$. In the case of JUMP-1D, we sampled 111 training points and 111 for testing from $(0, 1)$. Table 2 gives the results for theses experiments (averaged over 50 independent runs). Additionally, this table contains results for a two-dimensional simulated function NONSTAT-2D, which is described further below in this sub-section.

The results can be summarized as follows: with respect to the sMSE, the accuracy of our approach is comparable to the MCMC-based method of Paciorek and Schervish. Note that values given here were taken from their publication [8]. Both approaches significantly (α=0.05) outperform standard GPs. Our approach

Table 2. Quantitative evaluation of the proposed nonstationary approach (LA-GP) and the standard Gaussian process (STD-GP) as well as the MCMC-based approach of [8] (NS-GP). We compare the prediction accuracies using the negative log predictive density (NLPD) and the standardized mean squared errors (sMSE), see text. Results marked by • differ significantly ($\alpha = 0.05$) from the others in their category.

Test Scenario	NLPD		sMSE		
	LA-GP	**STD-GP**	**LA-GP**	**STD-GP**	**NS-GP [8]**
SMOOTH-1D	-1.100	-1.026 (•)	0.0156	0.021 (•)	0.015
JUMP-1D	-0.375	-0.440 (•)	0.0268	0.123 (•)	0.026
NONSTAT-2D	-3.405	-3.315 (•)	0.0429	0.0572 (•)	-

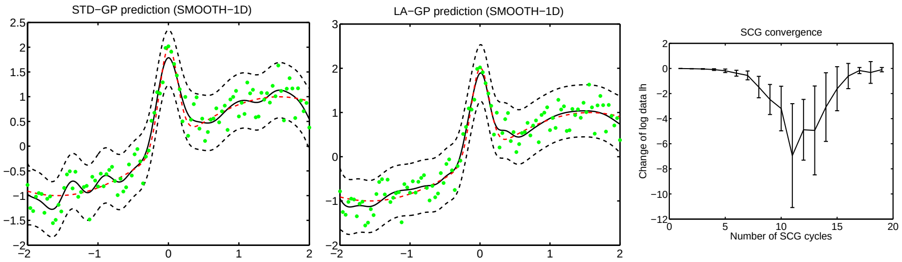

Fig. 3. Typical regression results in the SMOOTH-1D test scenario for the STD-GP model (left) and LA-GP (middle). The right diagram gives the statistics for changes of the objective function per SCG optimization cycle (in log data liklihood).

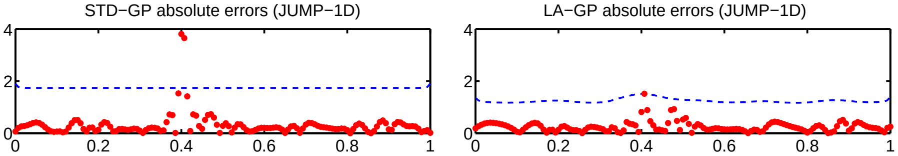

Fig. 4. Absolute distances of the test points from the predicted means in one run of the JUMP-1D scenario using the STD-GP model (left) and LA-GP (right). The model confidence bounds (2 standard deviations of the predictive distribution) are given by dashed lines.

also provides a significantly better performance compared to standard GPs with respect to the NLPD. For a visual comparison of the regression results, consider the left two diagrams in Figure 3. Whereas the standard GP (left plot) – having a constant length-scale for the whole domain – cannot adapt to all local properties well, our LA-GP accurately fits the bump and also the smoother parts (center plot). It should be noted that LA-GP tends to assign higher frequencies to the border regions of the training set, since there is less constraining data there compared to the center regions (see also the lower left plot in Figure 2).

The right diagram of Figure 3 provides statistics about the individual gains during the SCG cycles for 50 independent test runs. As can be seen from this plot, after about 20 cycles the objective function, which corresponds to the negative log data likelihood, does not change notably any more. Figure 4 compares the confidence bounds of the different regression models to the actual prediction errors made. It can be seen that the LA-GP model has more accurate bounds. It should be noted that the predictive variance of the STD-GP model depends only on the local data density and not on the target values and, thus, it is constant in the non-border regions.

We give the absolute average errors of the mean predictions in the different test cases in Figure 5. To highlight the more accurate confidence bounds of the LA-GP model, we also give the statistics for the 50% most confident predictions.

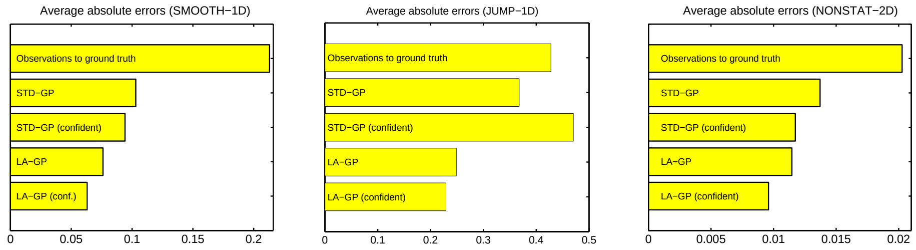

Fig. 5. Absolute average errors of the mean predictions in the SMOOTH-1D test scenario (left), JUMP-1D (middle), and NONSTAT-2D (right). We give the absolute distances of the simulated observations to the true function values, the overall average errors for the different models, and the average errors of the 50% most confidently predicted means respectively.

In addition to the two one-dimensional standard test cases, we evaluated the performance or our approach on a bivariate function (NONSTAT-2D). In particular, we simulated observations $y(x_1, x_2) \sim f(x_1, x_2) + \mathcal{N}(0, 0.025)$ using the noise-free bivariate function $f(x_1, x_2) = 1/10 \cdot (\sin(x_1\, b(x_1, x_2)) + \sin(x_2\, b(x_1, x_2))$ and the underlying bandwidth function $b(x_1, x_2) = \pi\ (2x_1 + 0.5x_2 + 1)$. This function and typical observations are depicted in the left diagram of Figure 6. During training, we sampled $11 \cdot 11 = 121$ points from a uniform distribution over $[-0.5, 1] \times [-0.5, 1]$ and corresponding simulated observations (the latter were drawn independently for each run). For testing, we uniformly sampled $31 \cdot 31 = 961$ points from $[-0.5, 1] \times [-0.5, 1]$ including their true function values. A typical example of the resulting optimized length-scales are visualized in the upper right contour plot of Figure 6. It can be seen that larger length-scales (which correspond to stronger smoothing) are assigned to the flat part of the surface around $(-0.5, 0)^T$ and smaller ones towards $(1, 1)^T$.

The quantitative results in terms of NLPD and sMSE for 30 independent test runs are given in Table 2. The absolute errors of the mean predictions are given in the right bar chart of Figure 5. The two lower plots of Figure 6 give a visual impression about the accuracy of the two regression models. We give the NLPD errors at equidistantly sampled test locations overlayed by contour plots of the predictive uncertainties. Note that the LA-GP model assigns higher confidence to the flat part of the function, which – given the uniform sampling of training points – can be reconstructed more accurately than the higher-frequency parts.

5.2 Modeling RFID Signal Strength

We have applied our nonstationary regression approach to the problem of learning the signal strength distribution of RFID (Radio Frequency Identification) tags. For this experiment, 21.794 signal strength measurements (logarithmic to the base of 10) have been recorded in a test setup at the University of Freiburg

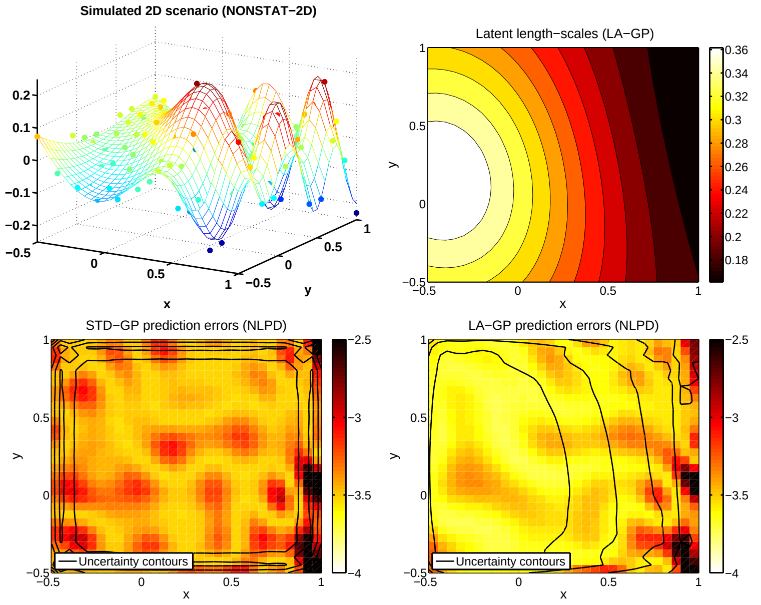

Fig. 6. The true function and noisy observations in the NONSTAT-2D test case (top left). Note the spatially varying oscillation frequency. The top right plot depicts the contours of the latent length-scales as estimated by our LA-GP model. In the two lower diagrams, we give the individual prediction errors (NLPD) of the Standard GP model (bottom left) and LA-GP (bottom right). The predictive uncertainty of the models is visualized using overlayed contours.

using a static antenna and a mobile, externally localized RFID tag. For efficiency reasons, only the left half-space of the antenna was sampled with real measurements and then mirrored along the respective axis. We randomly sampled 121 training points for learning the regression models and 500 different ones for evaluation. Note that although larger training sets lead to better models, we learn from this comparably small number of observations only to achieve faster evaluation runs. Table 3 gives the quantitative comparison to the standard GP model (STD-GP). As can be seen from the results, the standard model is outperformed by our nonstationary extension both in terms of sMSE and NLPD.

Table 3. Quantitative results for the RFID-2D experiment. Results marked by • differ significantly ($\alpha = 0.05$) from the others in their category.

	NLPD		sMSE	
Test Scenario	**LA-GP**	**STD-GP**	**LA-GP**	**STD-GP**
RFID-2D	-0.0101 (•)	0.1475	0.3352 (•)	0.4602

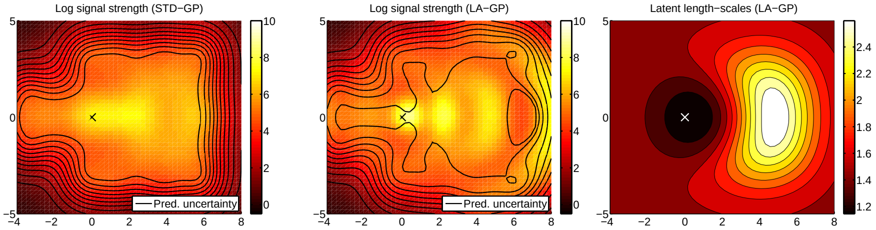

Fig. 7. Predicted mean log signal strengths of RFID tags using the standard GP (left) and the locally adapted GP (middle). The sensor location (0,0) is marked by a cross and the predictive uncertainties of the models are visualized by overlayed contours. The right plot visualizes the adapted latent length-scales of the LA-GP model. Coordinates are given in Meters.

Figure 7 shows predicted mean log signal strengths of the two models as color maps overlayed with contour plots of the corresponding predictive uncertainties. We also visualize the contours of the latent length-scales modeling higher frequencies in the proximity of the sensor location and lower ones in front of it.

5.3 Laser-based Terrain Mapping

We also applied our model to the particularly hard robotics problem of learning probabilistic terrain models from laser range measurements. In a joint project with the Massachusetts Institute of Technology, we have equipped a quadruped robot with a Hokuyo URG laser range sensor (see the left picture in Figure 8). The robot was programmed to perform a 'pushup' motion sequence in order to acquire a 3D scan of the local environment. For evaluation, we selected a $20 \times 20cm$ part of a rough terrain (with a maximum height of around 9 cm) including its front edge (see the middle plot of Figure 8). 4.282 laser end points of the 3D scan fall into this area.

We have trained the standard GP model and our nonstationary variant on 80 randomly selected training points from a noise-free simulation of the real terrain (TERSIM-2D) and evaluated the prediction accuracy for 500 test points

Table 4. Quantitative results for the simulated (TERSIM-2D) and the real (TERREAL-2D) terrain mapping experiment. Results marked by • differ significantly ($\alpha = 0.05$) from the others in their category.

	NLPD		sMSE	
Test Scenario	**LA-GP**	**STD-GP**	**LA-GP**	**STD-GP**
TERSIM-2D	-4.261 (•)	-4.198	0.127	0.126
TERREAL-2D	-3.652	-3.626	0.441 (•)	0.475

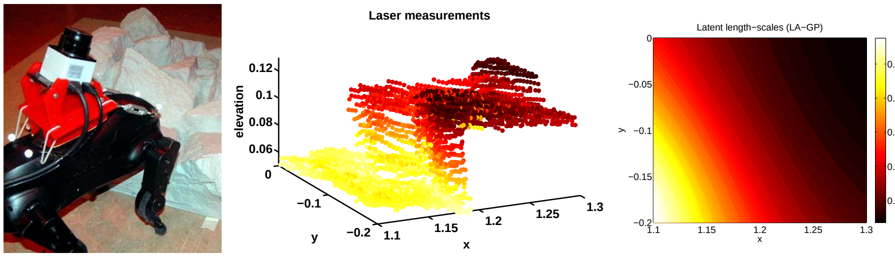

Fig. 8. A quadruped robot equipped with a laser sensor (left) acquires elevation measurements of a rough terrain surface (middle) by executing a 'pushup' motion. From a subset of elevation samples, our LA-GP approach learns a predictive model that captures the nonstationary nature of the data set (right).

(averaged over 30 independent runs). We repeated the same procedure on the real data (TERREAL-2D) and evaluated the prediction accuracy for other, independently selected test points from the real scan. Thus, the latter evaluation quantifies how well the models are able to predict other samples from the same distribution while the former gives the prediction errors relative to a known ground truth function. Table 4 gives the quantitative results for these two experiments. The right colormap in Figure 8 depicts the optimized length-scales of the LA-GP model. It can be seen that the flat part of the terrain is assigned larger local kernels compared to the rougher parts.

6 Conclusions

This paper has shown that GP regression with nonstationary covariance functions can be realized efficiently using point estimates of the latent local smoothness. The experimental results have shown that the resulting locally adaptive GPs perform significantly better than standard GPs and that they have the potential to solve hard learning problems from robotics and embedded systems.

There are several interesting directions for future work. First, the idea of optimizing the parameters of the latent and the observed process jointly should be applied to GP regression with input-dependent noise. In robotics applications, one is likely to encounter both, input-dependent noise and variable smoothness. Hence, the joint treatment of both should be addressed. Another direction is the extensions of our approach to the pseudo-noise setting introduced by Snelson and Ghahramani, see e.g. [15], so that the locations of the length-scale support values are learned from data, too. Finally, one should investigate multi-task learning e.g. along the lines of Yu *et al.* [17] to generalize e.g. across different types of terrains.

Acknowledgments The authors would like to thank the anonymous reviewers for their helpful comments. Many thanks to Christopher Paciorek for helpful discussions on nonstationary GP regression. The RFID data set was acquired by Dominik Joho at the University of Freiburg. Many thanks to Nicholas Roy, Sam Prentice, and Russ Tedrake at MIT CSAIL and to the DARPA Learning Locomotion program for making the robot terrain mapping experiment possible. Kristian Kersting was supported by the Fraunhofer Attract fellowship STREAM. Christian Plagemann and Wolfram Burgard were supported by the EC under contract number FP6-004250-CoSy and by the German Ministry for Education and Research (BMBF) through the DESIRE project.

References

1. A. Brooks, A. Makarenko, and B. Upcroft. Gaussian process models for sensor-centric robot localisation. In *Proc. of ICRA*, 2006.
2. Mike Brooks. The matrix reference manual. http://www.ee.ic.ac.uk/hp/staff/dmb/matrix/intro.html.
3. D. Cornford, I. Nabney, and C. Williams. Adding constrained discontinuities to gaussian process models of wind fields. In *Advances in Neural Information Processing Systems 11 (NIPS)*. Cambridge, MA, 1999.
4. I. Dimatteo, C.R. Genovese, and R.E. Kass. Bayesian curve-fitting with free-knot splines. *Biometrika*, 88(4):1055–1071, Dec. 2001.
5. K. Kersting, C. Plagemann, P. Pfaff, and W. Burgard. Most likely heteroscedastic gaussian processes regression. In Zoubin Gharahmani, editor, *ICML 07*, 2007.
6. T. Lang, C. Plagemann, and W. Burgard. Adaptive non-stationay kernel regression for terrain modelling. In *Proc. of Robotics: Science and Systems (RSS)*, 2007.
7. M. Møller. A Scaled Conjugate Gradient Algoritm for Fast Supervised Learning. *Neural Networks*, 6:525–533, 1993.
8. C. Paciorek and M. Schervish. Nonstationary covariance functions for Gaussian process regression. In S. Thrun, L. Saul, and B. Schoelkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.
9. T. Pfingsten, M. Kuss, and C.E. Rasmussen. Nonstationary gaussian process regression using a latent extension of the input space. In *Extended Abstract in Proc. of ISBA Eighth World Meeting on Bayesian Statistics*, Valencia, Spain, 2006.
10. C. Plagemann, D. Fox, and W. Burgard. Efficient failure detection on mobile robots using particle filters with gaussian process proposals. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, Hyderabad, India, 2007.
11. C. E. Rasmussen and C. K.I. Williams. *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. The MIT Press, 01 2006.
12. P.D. Sampson and P. Guttorp. Nonparametric estimation of nonstationary spatial covariance structure. *Journal of the American Stat. Association*, 87:108–119, 1992.
13. A.M. Schmidt and A. OHagan. Bayesian inference for nonstationary spatial covariance structure via spatial deformations. *JRSS, series B*, 65:745–758, 2003.
14. A. Schwaighofer, M. Grigoras, V. Tresp, and C. Hoffmann. A Gaussian process positioning system for cellular networks. In *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.
15. E. Snelson and Z. Ghahramani. Variable noise and dimensionality reduction for sparse gaussian processes. In *UAI*, 2006.
16. O. Williams. A switched Gaussian process for estimating disparity and segmentation in binocular stereo. In *Neural Info. Proc. Systems (NIPS)*, 2006.
17. K. Yu, V. Tresp, and A. Schwaighofer. Learning gaussian processes from multiple tasks. In *ICML 07*, 2007.