Most Likely Heteroscedastic Gaussian Process Regression

Kristian Kersting KERSTING@CSAIL.MIT.EDU
CSAIL, Massachusetts Institute of Technology, 32 Vassar Street, Cambridge, MA, 02139-4307, USA

Christian Plagemann PLAGEM@INFORMATIK.UNI-FREIBURG.DE
Patrick Pfaff PFAFF@INFORMATIK.UNI-FREIBURG.DE
Wolfram Burgard BURGARD@INFORMATIK.UNI-FREIBURG.DE
Department of Computer Science, University of Freiburg, Georges-Koehler-Allee 079, 79110 Freiburg, Germany

Abstract

This paper presents a novel Gaussian process (GP) approach to regression with input-dependent noise rates. We follow Goldberg et al.'s approach and model the noise variance using a second GP in addition to the GP governing the noise-free output value. In contrast to Goldberg et al., however, we do not use a Markov chain Monte Carlo method to approximate the posterior noise variance but a most likely noise approach. The resulting model is easy to implement and can directly be used in combination with various existing extensions of the standard GPs such as sparse approximations. Extensive experiments on both synthetic and real-world data, including a challenging perception problem in robotics, show the effectiveness of most likely heteroscedastic GP regression.

1. Introduction

Gaussian processes (GPs) have emerged as a powerful yet practical tool for solving various machine learning problems such as non-linear regression or multi-class classification (Williams, 1998). The increasing popularity is due to the fact that non-linear 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. An important practical problem, that has been addressed in the recent literature, is to relax the assumption of constant noise made in the standard GP model. In many real-world problems, the local noise rates are important features of data distributions that have to be modeled accurately. Consider for example the *Motorcycle* benchmark dataset depicted in Fig. 1. While the standard GP regression model quite accurately estimates the mean of the sought after distribution, it clearly overestimates the data variance in some areas and underestimates it in others. In contrast, taking the input-dependent noise into account the variance in the flat regions becomes low. The main contribution of the present paper is a novel GP treatment of input-dependent noise. More precisely, we follow Goldberg et al.'s (1998) approach and model the noise variance using a second GP in addition to the GP governing the noise-free output value. In contrast to Goldberg et al., however, we do not apply a time consuming Markov chain Monte Carlo method to approximate the posterior noise variance but replace it with an approximative most likely noise approach. This treatment allows us to develop a fast (hard) EM-like procedure for learning both the hidden noise variances and, in contrast to other approaches, also the kernel parameters. Experiments on synthetic and real-world data sets show that our most likely noise approach clearly outper-

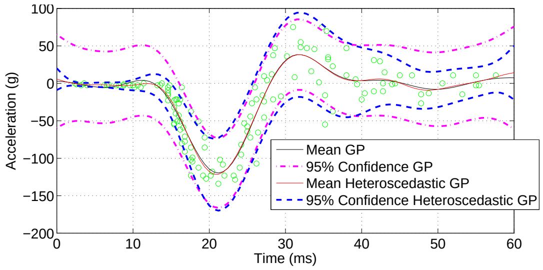

Figure 1. Silverman's (1985) motorcycle benchmark is an example for input dependent noise. It consists of a sequence of accelerometer readings through time following a simulated motor-cycle crash.

Appearing in *Proceedings of the 24th International Conference on Machine Learning*, Corvallis, OR, 2007. Copyright 2007 by the author(s)/owner(s).

forms standard GP regression and is competitive with existing heteroscedastic regression approaches. At the same time, our approach is substantially less complex than previous ones and has the additional advantage of fully staying within the GP regression framework. Extensions to standard GPs such as online learning, dependent outputs, non-stationary covariance functions, and sparse approximation can directly be be adapted. In the present paper, we will exemplify this by combining our model with the *projected process* approximation (Rasmussen & Williams, 2006), which only represents a small subset of the data for parameter estimation and inference. As our experiments show, this can keep memory consumption low and speed up computations tremendously.

Aside from this, we discuss a challenging, new application area for heteroscedastic regression, namely the modeling of range sensors for robotics applications. Modeling range sensors is an important task in robotics and engineering. We will review this modeling problem in the experimental section and show that heteroscedastic GP regression outperforms standard GP regression as well as customized existing models when applied to the task of mobile robot localization. This establishes a new, interesting link between the machine learning and the robotics communities as encouraged by the NIPS 2005 workshop on “Open Problems in Gaussian Processes for Machine Learning”[1]. Actually, robotics applications go one step ahead as they typically call for non-standard settings such as periodic covariance functions and heteroscedasticity.

We proceed as follows. After reviewing related work in Section 2, we will develop our most likely heteroscedastic GP regression model in Section 3, discuss parameter adaptation in Section 4, and show how to achieve sparse approximations in Section 5. Before concluding, we will present the results of an extensive set of experiments including the mobile robot localization task.

2. Related Work

The non-linear regression problem has been extensively studied in research areas such as machine learning, statistics, or engineering. While many existing approaches to the problem assume constant noise throughout the domain, there is also a growing body of work addressing heteroscedasticity, i.e., varying levels of noise. Schölkopf et al. (2000) present an SVM based algorithm that takes a known variance function into account. Nott (1996) propose a Bayesian model based on penalized splines and give an MCMC algorithm for inferring the posterior. Chan et al. (2006) derive a similar model for the Gaussian case, which adapts the noise variances and also requires MCMC for inference. Edakunni et al. (2007) presents a mixture of local linear regression models that can be learned using variational Bayesian EM. Opsomer et al. (1997) present an iterative procedure for dealing with heteroscedasticity in the context of kriging. They assume a linear model for the mean that is fitted using generalized least squares. Snelson et al. (2003) propose a nonlinear transformation of the output space to model a kind of output-dependent noise variances. Yuan and Wahba (2004) also jointly estimate the mean and noise variances but do not deal with the problem of selecting the kernel function. Le et al. (2005) also estimate the variance non-parametrically along with the mean of the distribution. In contrast to other approaches, they propose a maximum-a-posteriori estimation of the natural parameters in the exponential family. This yields, for the case of given kernel parameters, a convex optimization problem that can be solved efficiently. Recently, Snelson and Ghahramani (2006) proposed to utilize the dependency of the predictive uncertainty on the density of input data points.

3. The Model

The non-linear regression problem is to recover a functional dependency $t_i = f(\mathbf{x}_i) + \epsilon_i$ from n observed data points $\mathcal{D} = \{(\mathbf{x}_i, t_i)\}_{i=1}^n$. Here, $t_i \in \mathbb{R}$ are the (noisy) observed output values at input locations $\mathbf{x}_i \in \mathbb{R}^d$. The task is to learn a model for $p(t^*|\mathbf{x}^*, \mathcal{D})$, i.e., the predictive distribution of new target values t^* indexed by $\mathbf{x}^*$ depending on the observed data set $\mathcal{D}$. If we assume independent, normally distributed noise terms $\epsilon_i \sim \mathcal{N}(0, \sigma_i)$, where the noise variances σ_i are modeled by $\sigma_i = r(\mathbf{x}_i)$, i.e., by a function of $\mathbf{x}$, we get a heteroscedastic regression problem as studied by Goldberg et al. (1998), where the noise rate is not assumed constant on the domain. By placing a Gaussian process prior on f and assuming a noise rate function $r(\mathbf{x})$, the predictive distribution $P(\mathbf{t}^*|\mathbf{x}_1^*, \ldots, \mathbf{x}_q^*)$ at the query points $\mathbf{x}_1^*, \ldots, \mathbf{x}_q^*$ is a multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}^*, \Sigma^*)$ with mean

$$\boldsymbol{\mu}^* = \mathrm{E}[\mathbf{t}^*] = K^* \left(K + R\right)^{-1} \mathbf{t} \tag{1}$$

and covariance matrix

$$\Sigma^* = \mathrm{var}[\mathbf{t}^*] = K^{**} + R^* - K^* \left(K + R\right)^{-1} K^{*T} \ . \tag{2}$$

In these equations, we have $K \in \mathbb{R}^{n \times n}$, $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$, $K^* \in \mathbb{R}^{q \times n}$, $K^*_{ij} = k(\mathbf{x}^*_i, \mathbf{x}_j)$, $K^{**} \in \mathbb{R}^{q \times q}$, $K^{**}_{ij} = k(\mathbf{x}^*_i, \mathbf{x}^*_j)$, $\mathbf{t} = (t_1, t_2, \ldots, t_n)^T$, $R = \mathrm{diag}(\mathbf{r})$ with $\mathbf{r} = (r(\mathbf{x}_1), r(\mathbf{x}_2), \ldots, r(\mathbf{x}_n))^T$, and $R^* = \mathrm{diag}(\mathbf{r}^*)$ with $\mathbf{r}^* = (r(\mathbf{x}^*_1), r(\mathbf{x}^*_2), \ldots, r(\mathbf{x}^*_q))^T$.

[1] See http://gp.kyb.tuebingen.mpg.de/

An integral part of this model is the covariance function $k(\mathbf{x}_i, \mathbf{x}_j)$ that specifies the covariance $\mathrm{cov}(t_i, t_j)$ of the corresponding targets. Common choices, that we also employ throughout this work, are the squared exponential covariance function

$$k_{SE}(\mathbf{x}_i, \mathbf{x}_j) = \sigma_f^2 \exp\left(-(\Delta_{ij}^2)/(2\ell^2)\right) \ , \qquad (3)$$

with $\Delta_{ij} = \|\mathbf{x}_i - \mathbf{x}_j\|$, which has a relatively strong smoothing effect, or instances of the Matern type of covariance functions, like $k_M(\mathbf{x}_i, \mathbf{x}_j) =$

$$\sigma_f^2 \left(1 + \frac{\sqrt{5}\Delta_{ij}}{\ell} + \frac{\sqrt{5}\Delta_{ij}^2}{3\ell^2}\right) \cdot \exp\left(-\frac{\sqrt{5}\Delta_{ij}}{\ell}\right) \ .$$

These two covariance functions are called *stationary*, since they only depend on the distance Δ_{ij} between input locations $\mathbf{x}_i$ and $\mathbf{x}_j$. In the definitions above, σ_f denotes the amplitude (or signal variance) and ℓ is the characteristic length-scale, see Rasmussen and Williams (2006) for a detailed discussion. These parameters are called hyper-parameters of the process. They are typically denoted as $\boldsymbol{\theta} = (\sigma_f, \ell)$.

Goldberg et al. (1998) do not specify a functional form for the noise level $r(\mathbf{x})$ but place a prior over it. More precisely, an independent GP is used to model the logarithms of the noise levels, denoted as $z(\mathbf{x}) = \log(r(\mathbf{x}))$. This z-process is governed by a different covariance function k_z, parameterized by $\boldsymbol{\theta}_z$. The locations $\mathbf{x}_1, \ldots \mathbf{x}_n$ of the training data points $\mathbf{z} = \{z_1, z_2, \ldots, z_n\}$ for the z-process can be chosen arbitrarily, however, for notational convenience, we set them to coincide with the ones of the t-process here.

Since the noise rates z_i are now independent latent variables in the combined regression model, the predictive distribution for $\mathbf{t}^*$, i.e., the vector of regressands at points $\mathcal{X}^* = \{\mathbf{x}_1^*, \ldots, \mathbf{x}_q^*\}$, changes to $P(\mathbf{t}^*|\mathcal{X}^*, \mathcal{D}) =$

$$\int\int P(\mathbf{t}^*|\mathcal{X}^*, \mathbf{z}, z^*, \mathcal{D}) \ \cdot P(\mathbf{z}, z^*|\mathcal{X}^*, \mathcal{D}) \ d\mathbf{z} \ dz^* \ . \quad (4)$$

Given $(\mathbf{z}, z^*)$, the prediction $P(\mathbf{t}^*|\mathcal{X}^*, \mathbf{z}, z^*, \mathcal{D})$ is Gaussian with mean and variance as defined by (1) and (2). The problematic term is indeed $P(\mathbf{z}, z^*|\mathcal{X}^*, \mathcal{D})$ as it makes the integral difficult to handle analytically. Therefore, Goldberg et al. proposed a Monte Carlo approximation. More precisely, given a representative sample $\{(\mathbf{z}_1, z_1^*), \ldots, (\mathbf{z}_k, z_k^*)\}$ of (logarithmic) noise rates the integral (4) can be approximated by $\frac{1}{k}\sum_{j=1}^{k} P(t^*|\mathcal{X}^*, \mathbf{z}_j, z_j^*, \mathcal{D})$. The sampling is quite time consuming and the expectation can be approximated by the most likely noise levels $(\tilde{\mathbf{z}}, \tilde{z}^*)$. That is, we approximate the predictive distribution by $P(\mathbf{t}^*|\mathcal{X}^*, \mathcal{D}) \approx P(\mathbf{t}^*|\mathcal{X}^*, \tilde{\mathbf{z}}, \tilde{z}^*, \mathcal{D})$, where $(\tilde{\mathbf{z}}, \tilde{z}^*) = \arg\max_{(\tilde{\mathbf{z}}, \tilde{z}^*)} P(\tilde{\mathbf{z}}, \tilde{z}^*|\mathcal{X}^*, \mathcal{D})$. This will be a good approximation if most of the probability mass of $P(\mathbf{z}, z^*|\mathcal{X}^*, \mathcal{D})$ is concentrated around $(\tilde{\mathbf{z}}, \tilde{z}^*)$. Moreover, computing the most likely noise level and $P(\mathbf{t}^*|\mathcal{X}^*, \mathcal{D})$ now requires only standard GP inference, which is faster than the fully Bayesian treatment.

4. Optimization

So far, we have described our model and how to make predictions assuming that we have the parameters $\boldsymbol{\theta}_z$ of the z-process and the parameters $\boldsymbol{\theta}$ of the noise-free t-process, which uses the predictions of the z-process as noise variances at the given points. In practice, we are unlikely to have these parameters a-priori and, instead, we would like to estimate them from data.

The basic observation underlying our approach is very similar to the one underlying the (hard) EM algorithm: learning would be easy if we knew the noise level values for all the data points. Therefore, we iteratively perform the following steps to find the parameters:

1. Given the input data $\mathcal{D} = \{(\mathbf{x}_i, t_i)\}_{i=1}^n$, we estimate a standard, homoscedastic GP G_1 maximizing the likelihood for predicting t from $\mathbf{x}$.

2. Given G_1, we estimate the empirical noise levels for the training data, i.e., $z_i' = \log(\mathrm{var}[t_i, G_1(\mathbf{x}_i, \mathcal{D})])$, forming a new data set $\mathcal{D}' = \{(\mathbf{x}_1, z_1'), (\mathbf{x}_2, z_2'), \ldots, (\mathbf{x}_n, z_n')\}$.

3. On $\mathcal{D}'$, we estimate a second GP G_2.

4. Now we estimate the combined GP G_3 on $\mathcal{D}$ using G_2 to predict the (logarithmic) noise levels r_i.

5. If not converged, we set $G_1 = G_3$ and go to step 2.

To summarize the procedure, we take the current noise model and complete the data, i.e., make the noise levels observed. We then fix the completed data cases and use them to compute the maximum likelihood parameters of G_3. This process is repeated until convergence. Like the hard EM, the algorithm is not guaranteed to improve the likelihood in each step and can start oscillating as it considers most-likely completions of the data only. In our experminents, however, this happened only occasionally and only at reasonably accurate estimates.

Step 2, i.e., the empirical estimation of the noise levels is most crucial step o the procedure: **given** the data $\mathcal{D} = \{(\mathbf{x}_i, t_i)\}_{i=1}^n$ and the predictive distribution of the current GP estimates, **find** an estimate of the noise levels $\mathrm{var}[t_i, G_1(\mathbf{x}_i, \mathcal{D})]$ at each $\mathbf{x}_i$. Consider a sample t_i^j from the predictive distribution induced by the current GP at $\mathbf{x}_i$. Viewing t_i and t_i^j as two independent observations of the same noise-free, unknown target, their arithmetic mean $(t_i - t_i^j)^2/2$ is a natural estimate for the noise level at $\mathbf{x}_i$. Indeed, we can improve the estimate by taking the expectation with respect to the predictive distribution. This yields

$$\mathrm{var}[t_i, G_1(\mathbf{x}_i, \mathcal{D})] \approx s^{-1} \sum\nolimits_{j=1}^{s} 0.5 \cdot (t_i - t_i^j)^2$$

where s is the sample size and the t_i^j are samples from the predictive distribution induced by the current GP at $\mathbf{x}_i$. This minimizes the average distance between the predictive distribution and the prototype value t_i. For large enough number of samples ($s > 100$), this will be a good estimate for the noise levels.

5. Sparse Approximation

The heteroscedastic regression model presented in the previous section can directly be combined with various extensions of the GP model, like online learning, dependent outputs, non-stationary covariance functions, and sparse approximations. To exemplify this, we discuss how the *projected process* approximation (Rasmussen & Williams, 2006) can be applied to our model to increase its efficiency for large data sets. Section 6.3 also gives experimental results for this extension.

Several approximative models have been proposed for GPs in order to deal with the high time and storage requirements for large training data sets. In general, existing approaches select a subset $\mathcal{I}$, $|\mathcal{I}| = m$, of data points (the support set) from the full training set $\mathcal{D}$, $|\mathcal{D}| = n$, to reduce the complexity of learning, model representation, and inference. In contrast to simpler approaches that discard $\mathcal{D} \backslash \mathcal{I}$ completely, the so called *projected process* (PP) approximation considers a projection of the m-dimensional space of $\mathcal{I}$ up to n dimensions in order to be able to involve all available data points. The key idea is to only represent $m < n$ latent function values, denoted as $\mathbf{f}_m$ with $f_i = f(\mathbf{x}_i), \mathbf{x}_i \in \mathcal{I}$, which leads to smaller matrices that have to be stored and calculated. Then, in the homoscedastic case where a constant noise level σ_n^2 is assumed, the 'discarded' points $\mathbf{t}_{n-m}$ from $\mathcal{D} \setminus \mathcal{I}$ are modeled by

$$\mathbf{t}_{n-m} \sim \mathcal{N}(\mathrm{E}[\mathbf{f}_{n-m}|\mathbf{f}_m], \sigma_n^2 I) \;. \qquad (5)$$

As detailed in (Rasmussen & Williams, 2006), this leads to an easy to implement modification of the predictive distribution $p(t^*|\mathbf{x}^*, \mathcal{D})$. For our heteroscedastic model, we replace $\sigma_n^2 I$ in Equation (5) by the input noise rate matrix R (as defined in Section 3), which leads to a straightforward modification of the approximated predictive distribution of the homoscedastic case. An issue not discussed so far is how to select the active set $\mathcal{I}$ from $\mathcal{D}$. While existing approaches make informed selection decisions based on the information gain or on the predictive variance at prospective points in a greedy fashion, we employed a simple random sampling strategy for the experiments reported in Section 6.3. Due to this, the results reported there can be seen as a lower bound for the performance of our model under the PP approximation. More importantly, it takes only time $\mathcal{O}(m^2 n)$ to carry out the necessary matrix computations. For a fixed m, this is linear in n as opposed to $\mathcal{O}(n^3)$ for standard GPs.

6. Experiments

The goal of our experimental evaluation was to investigate to which extent most likely heteroscedastic GP regression is able to handle input-dependent noise:

(Q1) Is there a gain over standard GP regression?
(Q2) Can it rediscover the hidden noise function?
(Q3) Can it deal with non-smooth noise?
(Q4) Can sparse GP techniques be employed?
(Q5) Are there real-world applications in which it is useful and outperforms standard GP regression?

We conducted several experiments on benchmark data sets as well as in the context of mobile robot localization. We implemented our approach in `Matlab` using Rasmussen and Williams's (2006) GP toolbox as well as in `C++` for the robotics experiment. The benchmark data set experiments were run on a PowerBook G4 using `Matlab` 7.2 using a squared exponential covariance function. The mobile robot localization experiments were run on a 2.1GHz P4 Dual Core workstation using Linux using a Matern covariance function for the constant noise process and a squared exponential for the noise process. The parameters were always initialized randomly. As performance measures, we used two different losses. For traditional reasons, we report on the normalized mean squared error MSE $= \frac{1}{n}\sum_{i=1}^{n} \frac{(t_i - m_i)^2}{\mathrm{var}(y)}$, where m_i is the mean of the estimated predictive distribution $p(t_i|x_i)$ and $\mathrm{var}(t)$ is the empirical variance of the data, which only takes a point prediction into account that minimizes squared errors. A better loss for our task is the average negative log estimated predictive density NLPD $= \frac{1}{n}\sum_{i=1}^{n} - \log p(t_i|\mathbf{x}_i)$, which penalizes over-confident predictions as well as under-confident ones.

6.1. Benchmark Data Sets

We evaluated most likely heteroscedastic GP regression on the following benchmark data sets known from the literature, which have been used to empirically in-

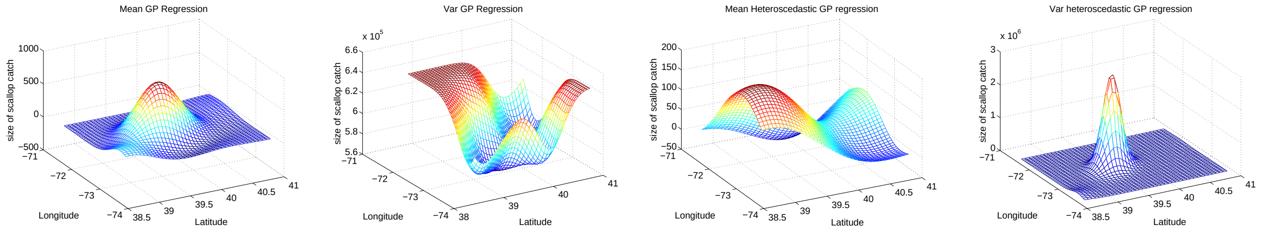

Figure 2. Size of scallop catch prediction. (Left) mean and variance estimate for standard GP regression. (Right) mean and variance prediction for most likely heteroscedastic GP regression. Note the difference in both the mean and the variance prediction. Standard GP regression is unable to adapt to the increase in noise at the location of higher variance.

Table 1. Mean test set MSE and NLPD over 10 reruns. In each run, the data set was randomly split into 90% training and 10% test data points. A '•' denotes a significant improvement (t-test, $p = 0.05$) over the corresponding value. Values are rounded to the second digit.

Data-	**MSE**		**NLPD**	
set	**GP**	**Het GP**	**GP**	**Het GP**
G	0.40 ± 0.20	0.40 ± 0.19	1.57 ± 0.31	1.46 ± 0.30•
Y	0.88 ± 0.19	0.89 ± 0.18	1.66 ± 0.21	1.37 ± 0.26•
W	0.49 ± 0.30	0.49 ± 0.30	0.78 ± 0.35	0.35 ± 0.39•
L	0.49 ± 0.30	0.49 ± 0.30	0.78 ± 0.36	0.35 ± 0.39•

vestigate other heteroscedastic regression methods:

G: The synthetic data originally used by Goldberg et al. (1998): 100 points x_i have been chosen uniformly spaced in the interval $[0, 1]$ and the targets $t_i = 2\sin(2\pi x_i)$ have been corrupted with a Gaussian noise where the standard deviation increases linearly from 0.5 at $x = 0$ to 1.5 at $x = 1$.

Y: The synthetic data originally used by Yuan and Wahba (2004): 200 points x_i have been chosen uniformly spaced in $[0, 1]$. The targets were sampled from a Gaussian $t_i \sim \mathcal{N}(\mu(x_i), \exp(g(x_i)))$ with mean $\mu(x_i) = 2[\exp(-30(x_i - 0.25)^2) + \sin(\pi x_i^2)] - 2$ and the logarithm of the standard deviation $g(x_i) = \sin(2\pi x_i)$.

W: The synthetic data originally used by Williams (1996): 200 input x_i are drawn from a uniform distribution on $[0, \pi]$. The targets t_i are distributed according to a Gaussian with mean $\sin(2.5x_i) \cdot \sin(1.5x_i)$ and standard deviation $0.01 + 0.25(1 - \sin(2.5x_i))^2$.

L: The LIDAR data set (Sigrist, 1994) consists of 221 observations from a light detection and ranging experiment. The logarithm of the ratio of received light from two laser sources are given for several distances traveled before the light is reflected back to its source.

For each data set, we performed 10 independent runs. In each run, we randomly split the data into 90% for training and 10% for testing. Table 1 summarizes the experimental results on the test sets. As one can see, most likely heteroscedastic GP regression is always at least as good as GP regression and always significantly improves the estimated predictive distribution. We observed the same when we investigating Ecker and Heltshe's (1994) scallop data set. the data consist of 148 data points concerning scallop abundance and it is based on a 1990 survey cruise in the Atlantic continental shelf off Long Island, New York, USA. The input specifies the location (latitude and longitude) and the target is the size of scallop catch at this location. We performed a 20 times estimate on 129 randomly selected data points for training and tested the model on the remaining 19 points. On average, GP regression achieved a MSE of 1.93 ± 2.0 and a NLPD of 8.16 ± 0.64. Our heteroscedastic GP regression approach achieved a MSE of 1.03 ± 0.16 and a NLPD of 7.73 ± 1.78. The difference in NLPD is significant (t-test, $p = 0.07$), the one in MSE not. To summarize, the results clearly answer **Q1** in an affirmative way.

To investigate **Q2**, we ran experiments on all generated data sets, i.e., data sets **G**, **Y**, and **W**. In Fig. 3 (top and bottom-left), the average standard deviation of the inferred noises have been plotted. Notice how in all cases the estimated noise is in close agreement with the data generator. Moreover, they are in the range of the ones reported in the literature. Thus, our method is competitive with other heteroscedastic regression methods. This is clearly an affirmative answer to **Q2**.

To summarize, these results show that our method indeed improves GP regression, that it is able to rediscover the hidden noise function, and that it is competitive with other heteroscedastic regression approaches.

6.2. Non-Smooth Noise

Most likely heteroscedastic Gaussian processes assume the noise function to be smooth. Here, we will experi-

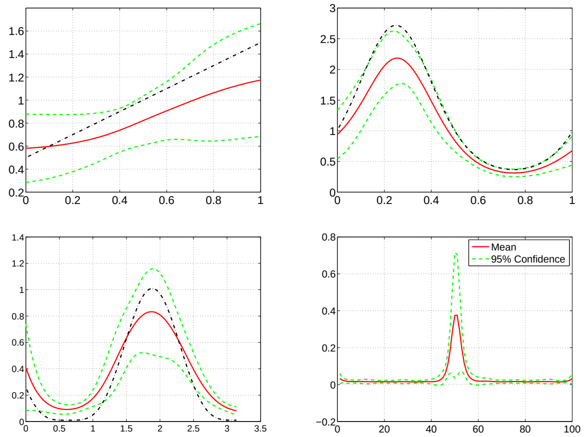

Figure 3. Solid curves give the averaged std. dev. of the noise and dashed curves the corresponding two-std.-dev. confidence interval. Dashed-dotted curves show the true noises. (Top-left) **G** data set: 30 runs à 60 samples. (Top-right) **Y** data set: 10 runs, 200 samples. (Bottom-left) **W** data set: 10 runs, 100 samples. (Bottom-right) Average variance for the step function (20 runs, 100 samples).

mentally investigate **Q3**, i.e., to which extent they can handle non-smooth noise functions. To this aim, we followed Cawley et al. (2006) and considered the step function on $[-1, 1]$: $f(x) = 1$ if $x > 0$ and 0 otherwise. 100 points have been chosen uniformly spaced in the interval $[-1, 1]$ and the targets have been corrupted with a Gaussian noise of standard deviation 0.1. The optimal predictive variance is very large around 0. A standard GP with stationary covariance function is in fact unable to model this. In contrast, the predictive variance of a most likely heteroscedastic GP captures the misfit around 0 well. Fig. 3 (bottom-right) shows the estimated variance averaged over 20 reruns. The peak is at zero and the average of 0.4 is the same as Cawley et al.'s (2006) result using 'leave-one-out heteroscedastic kernel regression'. The non-zero variance in the flat regions is directly related to the noise in the targets. This affirmatively answers **Q3**.

6.3. Sparse Approximations

In order to investigate **Q4**, i.e., sparse approximation techniques within most likely heteroscedastic GP regression, we ran three sets of experiments.

First, we reconsidered the benchmark data sets from Section 6.1. For the synthetic data sets, we sampled 1000 examples in each run; for the **L** data set, we used the original data set. The data was randomly split into 90% training and 10% test points. 100 random samples of the training set were used as support set.

Table 2. Mean test set MSE and NLPD over 10 runs of sparse approximation. A '•' denotes a significant improvement (t-test, $p = 0.05$) over the corresponding value. Values are rounded to the second digit.

	MSE		NLPD	
	GP	**Het GP**	**GP**	**Het GP**
G	0.73 ± 0.18	0.73 ± 0.17	2.02 ± 0.12	1.92 ± 0.16•
Y	0.88 ± 0.05	0.84 ± 0.05•	1.88 ± 0.14	1.46 ± 0.13•
W	0.59 ± 0.09	0.56 ± 0.11	0.90 ± 0.11	0.41 ± 0.18•
L	0.08 ± 0.04	0.08 ± 0.03	-1.03 ± 0.33	-1.35 ± 0.32•

Table 2 summarizes the results. As one can see, most likely heteroscedastic GP regression is again always at least as good as GP regression and always significantly improves the estimated predictive distribution.

Second, we investigated the `kin-8nh` data set. This data set was generated synthetically from a realistic simulation of the forward dynamics of an 8 link all-revolute robot arm[2]. The task is to predict the distance of the end-effector from a target. The inputs are 8 features describing quantities like joint positions, twist angles, etc. In total, there are 2000 training examples. We ran our approach 10 times and each time randomly selected a subset of 200 as support set. Standard GP regression achieved a MSE of 0.52 ± 0.03 and a NLPD of -0.23 ± 0.023 on the whole data set; the most likely heteroscedastic GP regression a MSE of 0.49 ± 0.03 and a NLPD of -0.26 ± 0.024. Both differences are significant (t-test, $p = 0.05$).

Third, we considered the Spatial Interpolation Comparison (SIC) 2004 competition. The target variable is ambient radioactivity measured in Germany. More precisely, the data are gamma dose rates reported by means of the national automatic monitoring network[3]. There are two scenarios: the "normal" and the "anomaly", which contains an anomaly in radiation at a specific location. We have focused on the "anomaly" scenario. As Le et al. (2005) point out, there is no reason to believe that radioactivity would exhibit highly nonuniform behavior. GP regression, however, is unable to cope with local noise due to the "step-like" anomaly. In contrast, heteroscedastic GP regression should adapt locally to the noise. To investigate this, we performed 10 random estimates using 400 of the 808 given examples as support set. The initial parameters were selected on the "normal" data set. On the complete data set, the standard GP achieved a MSE of 24.72 ± 8.51 and a NLPD of 6.84 ± 3.63, both averaged over the 10 runs. Our heteroscedastic approach achieved a MSE of 58.27 ± 29.17 and a NLPD

[2] See `http://www.cs.toronto.edu/~delve/`

[3] See `http://www.ai-geostats.org/events/sic2004/index.htm`

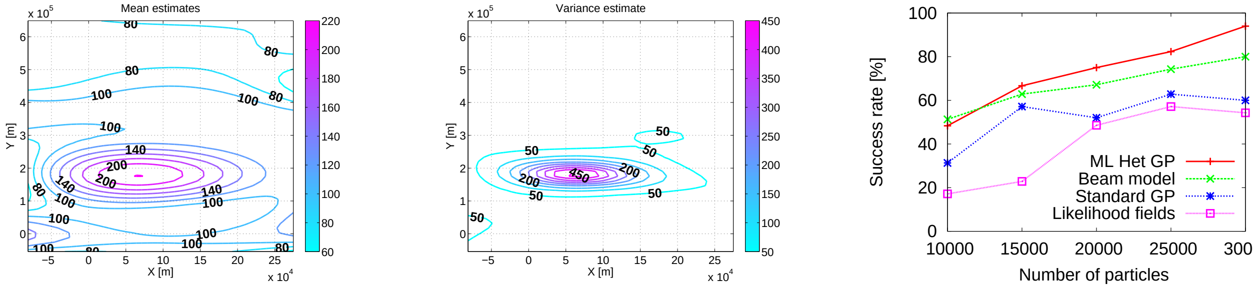

Figure 4. (Left and middle) Spatial interpolation Comparison (SIC) 2004 data. (Left) mean estimate and (middle) variance estimates for most likely heteroscedastic GP regression. Note the peak in variance at location of the outbreak. (Right) Pose estimation of a moving real robot: number of successful localizations after 8 integrated sensor readings for different numbers of particles used in the Monte Carlo localization algorithm.

of 4.21 ± 0.25. Thus, the most likely heteroscedastic GP modeled the predictive distribution significantly better but achieved a significantly worse MSE measure (t-test, $p = 0.05$). This is because the outbreak was identified as noise as shown in Fig. 4, which depicts a typical radioactivity prediction using our method. Actually, the estimated variance was only high at the location of the outbreak. This contrasts with standard GPs, which did not adapt to the local noise.

To summarize, the results of all three experiments affirmatively answer **Q4**, the SIC experiment also **Q3**. Furthermore, they confirmed the drop in runnning time from $\mathcal{O}(n^3)$ for standard GPs to $\mathcal{O}(m^2 n)$ for the projective process approximation.

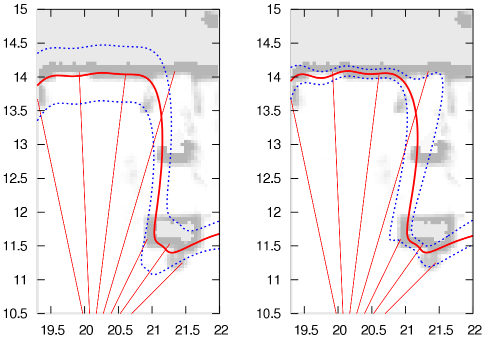

Figure 5. Standard GP regression (left), which assumes constant noise, and our most likely heteroscedastic GP regression (right), which deals with non-constant noise. The heteroscedastic approach yields lower predictive uncertainties at places with low expected noise levels such as the walls in front. Scales are given in meters. The red lines depict possible range measurements at this robot pose.

6.4. Mobile Robot Perception

We have applied our heteroscedastic regression framework to the problem of range sensor modeling for robotic applications. Here, the task is to interpret discrete sets of measured distances r_i along given beam directions α_i. An important type of range sensors are the so called laser range finders, which use a laser beam and a rotating mirror to determine the distances to reflective objects. In contrast to existing models (Thrun et al., 2005) that reason on the discrete set of measurements directly, we view the measurements as samples from a stochastic process and apply the heteroscedastic regression technique introduced in this paper. Fig. 5 illustrates the scenario of a mobile robot navigating in an office environment that uses a laser range finder to localize itself relative to a given map. The left diagram of this figure gives a typical predictive distribution of range measurements using standard GP regression, while the right diagram depicts the results using our approach. It can be seen from the diagrams, that the heteroscedastic model achieves physically more plausible predictions, where real test beams recorded at the same robot pose (visualized by red lines) receive higher observation likelihoods.

To quantitatively evaluate the benefits of our model in the robot perception domain, we ran a set of global localization experiments with a real robot. Here, the task was to find the pose of a moving robot within an environment using a stream of wheel encoder and laser measurements. The environment used consists of a long corridor and 8 rooms containing chairs, tables and other pieces of furniture. In total, the map is 20 meters long and 14 meters wide.

The results are summarized in the right diagram of Fig. 4, which gives the number of successful localizations after 8 integrated sensor readings for different numbers of particles used. We compared four models: the standard GP regression model, the most likely heteroscedastic GP regression model, and two stateof-the-art models used in the robotics community, i.e. the beam model and the likelihood fields model (Thrun et al., 2005). In this experiment, we assumed that the localization was achieved when the estimated pose was at most 30 cm apart from the true location of the robot. Our heteroscedastic model clearly outperforms the likelihood fields model and standard GPs, and it is slightly better than the beam-based model. Additional analysis using different data sets revealed that the heteroscedastic treatment is especially beneficial in highly cluttered environments, such as rooms containing many chairs and tables.

7. Conclusions

This paper has shown that effective Gaussian process (GP) regression with input-dependent noise can be fully implemented using standard GP techniques. In experimental tests, most likely heteroscedastic GP regression, the resulting approach, produced estimates that are significantly better than standard GPs and competitive with other heteroscedastic regression approaches. Furthermore, most likely heteroscedastic GP regression outperformed standard techniques on a challenging perception problem in robotics.

Directions for future work include studying online learning, classification, and applications within other learning tasks such as reinforcement learning. Furthermore, it would be interesting to investigate "almost surely convergence" along the lines of Bottou and Bengio (1995) and to understand it from a variational Bayes perspective.

Acknowledgements: The authors would like to thank the three anonymous reviewers for their helpful comments. This work has been supported by the EC under FP6-004250-CoSy, by the German Ministry for Education and Research (BMBF) through the DESIRE project, and by the German Science Foundation (DFG) within the Research Training Group 1103.

References

Bottou, L., & Bengio, Y. (1995). Convergence properties of the kmeans algorithm. *Advances in Neural Information Processing Systems*. MIT Press.

Cawley, G., Talbot, N., & Chapelle, O. (2006). Estimating Predictive Variances with Kernel Ridge Regression. *Machine Learning Challenges* (pp. 56–77).

Chan, D., Kohn, R., Nott, D., & Kirby, C. (2006). Locally-adaptive semiparametric estimation of the mean and variance functions in regression models. *Journal of Computational and Graphical Statistics*.

Ecker, M., & Heltshe, J. (1994). Geostatistical estimates of scallop abundance. In *Case studies in biometry*, 107–124. John Wiley and Sons.

Edakunni, N., Schaal, S., & Vijayakumar, S. (2007). Kernel carpentry for online regression using randomly varying coefficient model. *Proc. of IJCAI-07*.

Goldberg, P., Williams, C., & Bishop, C. (1998). Regression with input-dependent noise: A gaussian process treatment. *NIPS 1998*. MIT Press.

Le, Q., Smola, A., & Canu, S. (2005). Heteroscedastic gp regression. *Proc. of ICML-05* (pp. 489–496).

Nott, D. (1996). Semiparametric estimation of mean and variance functions for non-gaussian data. *Computational Statistics*, *21*, 603–620.

Opsomer, J., Ruppert, D., Wand, M., Holst, U., & Hossjer, O. (1997). *Kriging with nonparametric variance function estimation* (Technical Report).

Rasmussen, C., & Williams, C. (2006). *Gaussian processes for machine learning*. MIT Press.

Schölkopf, B., Smola, A. J., Williamson, R. C., & Bartlett, P. L. (2000). New support vector algorithms. *Neural Computation*, *12*, 1207–1245.

Sigrist, M. (1994). *Air Monitoring by Spectroscopic Techniques*, vol. 197 of *Chemical Analysis Series*. John Wiley and Sons.

Silverman, B. (1985). Some Aspects of the Spline Smoothing Approach to Non-parametric Regression Curve Fitting. *Journal of the Royal Statistical Society*, *47*, 1–52.

Snelson, E., & Ghahramani, Z. (2006). Variable noise and dimensionality reduction for sparse gaussian processes. *Proc. of UAI-06*.

Snelson, E., Rasmussen, C., & Ghahramani, Z. (2003). Warped gaussian processes. *NIPS 2003*.

Thrun, S., Burgard, W., & Fox, D. (2005). *Probabilistic robotics*. MIT Press.

Williams, C. (1998). Prediction with gaussian processes: From linear regression to linear prediction and beyond. In *Learning and inference in graphical models*, 599–621. Kluwer Acadamic.

Williams, P. (1996). Using neural networks to model conditional multivariate densities. *Neural Computation*, *8*, 843–854.

Yuan, M., & Wahba, G. (2004). Doubly penalized likelihood estimator in heteroscedastic regression. *Statistics and Probability Letter*, *69*, 11–20.