Learning Non-stationary System Dynamics Online Using Gaussian Processes*

Axel Rottmann and Wolfram Burgard

Department of Computer Science, University of Freiburg, Germany

Abstract. Gaussian processes are a powerful non-parametric framework for solving various regression problems. In this paper, we address the task of learning a Gaussian process model of non-stationary system dynamics in an online fashion. We propose an extension to previous models that can appropriately handle outdated training samples by decreasing their influence onto the predictive distribution. The resulting model estimates for each sample of the training set an individual noise level and thereby produces a mean shift towards more reliable observations. As a result, our model improves the prediction accuracy in the context of non-stationary function approximation and can furthermore detect outliers based on the resulting noise level. Our approach is easy to implement and is based upon standard Gaussian process techniques. In a real-world application where the task is to learn the system dynamics of a miniature blimp, we demonstrate that our algorithm benefits from individual noise levels and outperforms standard methods.

1   Introduction

Accurately modeling the characteristics of a system is fundamental in a wide range of research and application fields, and it becomes more important as the systems grow more complex and less constrained. A common modeling approach is to use probabilities to represent the dependencies between the system's variables and apply machine learning techniques to learn the parameters of the model from collected data. Consider for example the task of learning the system dynamics of a small blimp. The majority of existing approaches assume stationary systems and equally weight all the training data. The flight characteristics of the blimp, however, are affected by many unconsidered factors that change over time. A common approach to deal with non-stationary systems is to assign higher weights to newer training samples.

In this paper we present a probabilistic regression framework that can accurately describe a system even when its characteristics change over time. More concretely, we extend the Gaussian process (GP) framework to be able to handle training samples with different weights. GPs are a state-of-the-art non-parametric Bayesian regression framework that has been successfully applied
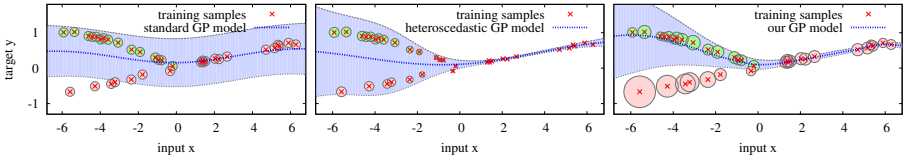

Fig. 1. Different observation noise assumptions in the data lead to different GP models. From left to right: standard approach assuming uniform noise levels, heteroscedastic GP model assuming input-dependent noise levels, and our approach assuming unrestricted noise levels where the samples circled green (the smaller circles) are assigned with higher weights.

for solving various regression problems. One limitation of standard GPs, however, is that the noise in the training data is assumed to be uniform over the whole input domain (homoscedasticity). This assumption is not always valid and different approaches have recently been proposed to deal with varying noise in the data. The main idea behind these approaches is to assume that the noise can be described by a function of the input domain (heteroscedasticity) so that adjacent data is supposed to have similar noise. However, both of these approaches effectively regard all training data as having the same weight. In this paper, we present a general extension of these GP approaches. Our model is able to deal with individual, uncorrelated observation noise levels for each single training sample and thereby is able to weight the samples individually. This flexibility allows us to apply our framework, for example, to an online learning scenario where the underlying function being approximated may change during data collection.

Figure 1 illustrates how different assumptions about the observation noise in the data lead to different predictive distribution of GP models. In the figure, the predicted mean, the 95% confidence interval, and the training samples are shown. The size of the circle around each point corresponds to its estimated noise; the bigger the radius, the larger the noise. The left plot corresponds to the most restrictive, standard GP model where the noise is assumed to be constant. In the plot in the middle, the noise is assumed to depend on the input domain. This corresponds to the more flexible heteroscedastic GP models. This model, however, still has limited flexibility since it does not allow us to deal with unrestricted noise levels. The approach presented in this paper is able to weight the training samples individually by assigning different noise levels. The right plot in Fig. 1 corresponds to a GP learned using our approach. Assuming the samples circled red are outdated information and assigned with lower weights than those in green circles our model selects the noise levels correspondingly. Note that our model as well assigns smaller noise levels to the samples circled red on the right side as there is no newer information available. Finally, the predictive mean function is shifted towards the higher weighted samples and the predictive variance reproduces the distribution of the training samples.

The main contribution of this paper is a novel GP framework that estimates individual, uncorrelated noise levels based on weights assigned to the training samples. Considering unrestricted noise levels allows us to increase the prediction accuracy compared to previous approaches as we can assign higher noise levels to inaccurate observations, so that their influence onto the regression is reduced.

This paper is organized as follows. After reviewing related work in Section 2 we give a short introduction of GP regression in Section 3. Afterward, in Section 4, we introduce our approach of estimating unrestricted noise levels. Finally, in Section 5 we provide several experiments demonstrating the advantages of our method, followed by a conclusion.

2   Related Work

Gaussian process regression has been intensively studied in the past and applied in a wide range of research areas such as statistics and machine learning. A general introduction to GPs and a survey of the enormous approaches of the literature is given in the book of Rasmussen and Kuss [1]. In most GP frameworks a uniform noise distribution throughout the domain is assumed. In contrast to this, a heteroscedastic noise prediction has as well been intensively studied. For instance, the approaches of Goldberg *et al.* [2] and Kersting *et al.* [3] deal with input-dependent noise rates. Both use two separate GPs to model the data. One predicts the mean as a regular GP does, whereas the other is used to model the prediction uncertainty. In contrast to our approach, they predict input-dependent noise levels. We extend their approach and additionally estimate for each single training sample an individual noise value. Heteroscedasticity has as well been applied in other regression models. For example, Schölkopf *et al.* [4] integrated a known variance function into an SVM-based algorithm and Bishop and Quazaz [5] investigated input-dependent noise assumptions for parametric models such as neuronal networks.

GPs have been also successfully applied to different learning tasks. Due to the limited space, we simply refer to some approaches which are closely related to the experiments performed in this paper. Ko *et al.* [6] presented an approach to improve a motion model of a blimp derived from aeronautic principles by using a GP to model the residual. Furthermore, Rottmann *et al.* [7] and Deisenroth *et al.* [8] learned control policies of a completely unknown system in a GP framework. In these approaches stationary underlying functions were assumed. As already discussed above, in real applications this is normally not the case and estimating an individual noise level for each observation can improve the prediction accuracy.

Additionally, alternative regression techniques have been successfully applied to approximate non-stationary functions. D'Souza *et al.* [9] used Locally Weighted Projection Regression to learn the inverse kinematics of a humanoid robot. The authors include a forgetting factor in their model to improve the influence of newer observations. In contrast to their approach, we obtain an observation noise estimate of each single training sample which, for instance, can be used to remove outdated observations from the data set.

3   Gaussian Process Regression

Gaussian processes (GPs) are a powerful non-parametric framework for regression and provide a general tool to solve various machine learning problems [1]. In the context of regression, we are given a training set $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^N$ of N, d-dimensional states $\boldsymbol{x}_i$ and target values y_i. We aim to learn a GP to model the dependency $y_i = f(\boldsymbol{x_i}) + \epsilon_i$ for the unknown underlying function $f(\boldsymbol{x})$ and, in case of a homoscedastic noise assumption, independent and identically, normally distributed noise terms $\epsilon_i \sim \mathcal{N}(0, \sigma_n^2)$.

A GP is fully specified by its mean $m(\boldsymbol{x})$ and covariance function $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Typical choices are a zero mean function and a parametrized covariance function. In this work, we apply $k(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sigma_f^2 \exp\left(-\frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{x}_j)^T \Lambda^{-1}(\boldsymbol{x}_i - \boldsymbol{x}_j)\right)$, where σ_f^2 is the signal variance and $\Lambda = \mathrm{diag}(\ell_1, \ldots, \ell_d)$ is the diagonal matrix of the length-scale parameters.

Given a set of training samples $\mathcal{D}$ for the unknown function and the hyperparameters $\boldsymbol{\theta} = (\Lambda, \sigma_f^2, \sigma_n^2)$ a predictive distribution $P(f^* \mid \boldsymbol{x^*}, \mathcal{D}, \boldsymbol{\theta})$ for a new input location $\boldsymbol{x}^*$ is again a Gaussian with

$$f_\mu^* = m(\boldsymbol{x^*}) + \boldsymbol{k}(x, \boldsymbol{x}^*)^T (K + R)^{-1} (\boldsymbol{y} - m(x)) \tag{1a}$$

$$f_{\sigma^2}^* = k(\boldsymbol{x}^*, \boldsymbol{x}^*) - \boldsymbol{k}(x, \boldsymbol{x}^*)^T (K + R)^{-1} \boldsymbol{k}(x, \boldsymbol{x}^*). \tag{1b}$$

Here, $K \in \mathbb{R}^{N \times N}$ is the covariance matrix for the training points with $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $R = \sigma_n^2 I$ is the observation noise.

4   GP Model with Individual Noise Levels

In general, GP regression can be seen as a generalization of weighted nearest neighbor regression and thus can be applied directly to model non-stationary underlying functions. As more and more accurate observations become available the predictive mean function is shifted towards the more densely located training samples. However, assigning lower weights to outdated samples improves the approximation accuracy regarding the actual underlying function. For this purpose, we assign a weighting value $w(\boldsymbol{x}_i), i = 1, \ldots, N$, to each single training sample $\boldsymbol{x}_i$. In case of GPs the weight of a sample and thus the importance on the predictive distribution can be regulated by adapting the observation noise correspondingly. Therefore, given the weights the presented GP framework estimates individual noise levels for each training samples to obtain the most likely prediction of the training samples. Obviously, the prediction accuracy of the actual underlying function highly depends on the progress of the weight values. However, in practical applications such values can easily be established and, even without having knowledge about the optimal values, raising the weights of only a few samples result in a significant improvement of the approximation. In an online learning task, the influence of subsequent observations can be boosted by monotonically increasing values over time. D'Souza *et al.* [9], for instance, apply a constant forgetting factor. Throughout our experiments we set

$$w(\boldsymbol{x}_i) = \begin{cases} 0.1 & \text{if } i < N - \Delta \\ 1.0 & \text{otherwise} \end{cases} \quad , i = 1, \ldots, N \,, \tag{2}$$

where N is the total number of training samples and the parameter Δ specifies how many of the more recent observations are assumed to have higher importance. Although we are only using a simple step function to define the weights, our GP framework is not restricted to any fixed distribution of these values.

To implement individual noise levels the noise matrix R of the GP model is replaced by the diagonal matrix $R_D = \text{diag}(\sigma_1^2, \ldots, \sigma_N^2)$ of the individual noise levels for each training point $x_1, \ldots, x_N$. In general, as the global noise rate σ_n^2 is simply split into individual levels $\sigma_1^2, \ldots, \sigma_N^2$ the GP model remains unaffected and the additional parameters can be added to the set of hyper-parameters $\boldsymbol{\theta}$. Given that, the noise levels can be estimated in the same fashion as the other hyper-parameters. In our current system, we employ leave-one-out cross-validation to adapt the hyper-parameters. Alternative cross-validation approaches in the context of GPs are described by Sundararajan and Keerthi [10]. In general, one seeks to find hyper-parameters that minimize the average loss of all training samples given a predefined optimization criterion. Possible criteria are the negative marginal data likelihood (GPP), the predictive mean squared error (GPE), and the standard mean squared error (CV). The weights $w(\boldsymbol{x}_i)$ can easily be integrated into cross-validation by adding the value of each training sample as an additional factor to the corresponding loss term. This procedure is also known as importance-weighted cross-validation [11].

From (1a) and (1b) we see, that for an arbitrary scaling of the covariance function the predictive mean remains unaffected whereas the predictive variance depends on the scaling. Initial experiments showed that the GPP and GPE criteria scale the covariance function to be zero as they minimize the average loss of all training points. The predictive mean remains unchanged whereas the predictive variance is successively decreased. Therefore, we employ the CV criterion, which is given as

$$CV(\boldsymbol{\theta}) = \frac{1}{w_N} \sum_{i=1}^{N} w(\boldsymbol{x}_i) (y_i - \mu_i^*)^2 \,. \tag{3}$$

Here, μ_i^* denotes the predicted mean value of $P\left(f_i^* \mid \boldsymbol{x}_i, \mathcal{D}^{(i)}, \boldsymbol{\theta}\right)$, $\mathcal{D}^{(i)}$ is obtained from $\mathcal{D}$ by removing the ith sample, and $w_N = \sum_{i=1}^{N} w(\boldsymbol{x}_i)$ is the normalization term of the importance values.

Using the CV criterion to optimize the hyper-parameters, we obtain individual, uncorrelated noise levels and a mean prediction — we will refer to it as f_μ^{CV} — which is optimal with respect to the squared error. However, an adequate variance prediction of the training samples is not obtained. This is based on the fact that the CV criterion simply takes the predictive mean into account. To achieve this, we employ, in a final step, the obtained mean function as a fixed mean function $m(\boldsymbol{x}) = f_\mu^{CV}$ of a second GP. The predicted mean of the first GP becomes a latent variable for the second one, so the predictive distribution can be written as $P\left(f^* \mid \boldsymbol{x}^*, \mathcal{D}, \boldsymbol{\theta}\right) = \int P\left(f^* \mid \boldsymbol{x}^*, \mathcal{D}, \boldsymbol{\theta}, f_\mu^{CV}\right) \cdot P\left(f_\mu^{CV} \mid \boldsymbol{x}^*, \mathcal{D}, \boldsymbol{\theta}\right) \, df_\mu^{CV}$.

Given the mean f_μ^{CV} the first term is Gaussian with mean and variance as defined by (1a) and (1b), and $m(\boldsymbol{x}) = f_\mu^{CV}$. To simplify the integral, we approximate the expectation of the second term by the most likely predictive mean $\tilde{f}_\mu^{CV} \approx \arg\max_{\tilde{f}_\mu^{CV}} P\left(\tilde{f}_\mu^{CV} \mid \boldsymbol{x}^*, \mathcal{D}, \boldsymbol{\theta}\right)$. This is a proper approximation as most of the probability mass of $P\left(f_\mu^{CV} \mid \boldsymbol{x}^*, \mathcal{D}, \boldsymbol{\theta}\right)$ is concentrated around the mean which minimizes the mean squared error to the training samples.

To adjust the hyper-parameters of the second GP — and to define the final predictive distribution — all we need to do is to apply a state-of-the-art GP approach. The individual noise levels have already been estimated in the first GP and they do not have to be considered in the covariance function of the second GP. Therefore, depending on the noise assumption of the training samples, one can choose a homoscedastic or heteroscedastic GP model. The final predictive variance of our model is defined by the second GP only whereas the individual noise levels $\sigma_1^2, \ldots, \sigma_N^2$ are taken into account in the predictive mean. Still, the hyper-parameters of the second model must be optimized with respect to the weights $w(\boldsymbol{x}_i)$. Therefore, we employ leave-one-out cross-validation based on the GPP criterion with included weights instead of the original technique of the chosen model.

5 Experimental Results

The goal of the experiments is to demonstrate that the approach above outperforms standard GP regression models given a non-stationary underlying function. We consider the task to learn the system dynamics of a miniature indoor blimp [12]. The system is based on a commercial 1.8 m blimp envelope and is steered by three motors. To gather the training data the blimp was flown in a simulation, where the visited states as well as the controls were recorded. To obtain a realistic movement of the system a normal distributed noise term was added to the control commands to simulate outer influence like gust of wind. The system dynamics of the blimp were derived based on standard physical aeronautic principles [13] and the parameters were optimized based on a series of trajectories flown with the real blimp. To evaluate the predictive accuracy, we used 500 randomly sampled points and determined the mean squared error (RMS) of the predictive mean of the corresponding GP model relative to the ground truth prediction calculated by the simulator.

The dynamics obtained from a series of states indexed by time can be written as $\boldsymbol{s}(t+1) = \boldsymbol{s}(t) + \boldsymbol{h}(\boldsymbol{s}(t), \boldsymbol{a}(t))$, where $\boldsymbol{s} \in S$ and $\boldsymbol{a} \in A$ are states and actions, respectively, t is the time index, and $\boldsymbol{h}$ the function which describes the system dynamics given state $\boldsymbol{s}$ and action $\boldsymbol{a}$. Using a GP model to learn the dynamics the input space consists of the state space S and the actions A and the targets represent the difference between two consecutive states $\boldsymbol{s}(t+1) - \boldsymbol{s}(t)$. Then, we learn for each dimension S_c of the state space $S = (S_1, \ldots, S_{|S|})$ a Gaussian process $GP_c : S \times A \to S_c$. Throughout our experiments, we used a fixed time interval $\Delta t = 1\,\text{s}$ between two consecutive states.

We furthermore carried out multiple experiments on benchmark data sets to evaluate the accuracy of our predictive model given a stationary underlying function. Throughout the experimental section, we use three different GP models: a standard GP model (StdGP) of [1], a heteroscedastic GP model (HetGP) of [3], and our GP model with individual, uncorrelated noise levels (InGP). As mentioned in Section 4, our approach can be combined with a homoscedastic as well as a heteroscedastic variance assumption for the data. In the individual experiments we always specify which specific version of our model was applied. We implemented our approach in `Matlab` using the GP toolbox of [1].

5.1   Learning the System Dynamics

In the first experiment, we analyzed if the integration of individual noise levels for each single training sample yields an improvement of the prediction accuracy. We learned the system dynamics of the blimp using our approach assuming a homoscedastic variance of the data and a standard GP model. To evaluate the performance we carried out several test runs. In each run, we collected 800 observations and calculated the RMS of the final prediction. To simulate a non-stationary behavior, we manually modified the characteristics of the blimp during each run. More precisely, after $320\,\mathrm{s}$ we increased the mass of the system to simulate a loss of buoyancy.

We additionally evaluated different distributions of the weights (2) by adjusting the parameter Δ, that specifies how many of the more recent observations have higher importance. Table 1 summarizes the results for different values of Δ. For each dimension of the state vector we plot the residual. The state vector of the blimp contains the forward direction X, the vertical direction Z, and the heading φ as well as the corresponding velocities $\dot{X}$, $\dot{Z}$, and $\dot{\varphi}$. As can be seen from Table 1, the vertical direction is mostly affected by modifying the mass. Also, as expected, increasing Δ raises the importance of more reliable observations (which are the more recent ones in this experiment). Consequently, we obtain a better prediction accuracy. Note that this already happens for values of Δ that are substantially smaller than the optimal one, which would be 480.

Table 1. Prediction accuracy of the system dynamics of the blimp using a standard GP model and our approach with a homoscedastic variance assumption of the data

model	$X_{(\mathrm{mm})}$	$\dot{X}_{(\mathrm{mm/s})}$	$Z_{(\mathrm{mm})}$	$\dot{Z}_{(\mathrm{mm/s})}$	$\varphi_{(\mathrm{deg})}$	$\dot{\varphi}_{(\mathrm{deg/s})}$
StdGP	53.7	40.7	128.4	18.4	3.3	1.2
LWPR	43.4	33.9	27.9	9.1	3.7	1.1
InGP, $\Delta = 200$	51.1	35.4	30.4	10.1	3.3	1.3
InGP, $\Delta = 300$	42.7	38.4	27.5	10.1	3.6	1.3
InGP, $\Delta = 400$	44.3	33.7	26.2	8.8	3.3	1.2
optimal prediction	50.4	32.1	12.5	7.1	3.0	1.3

For comparison, we trained a standard GP model based on stationary dynamics. The accuracy of this model, which corresponds to the optimum, is given in the bottom of Table 1. Furthermore, we applied the Locally Weighted Projection Regression (LWPR) algorithm [9] to the data set and obtained equivalent results compared to our approach with $\Delta = 400$. LWPR is a powerful method to learn non-stationary functions. In contrast to their approach, however, we obtain for each input value a Gaussian distribution over the output values. Thus, the prediction variance reproduce the distribution of the trained samples. Further outputs of our model are individual, uncorrelated noise levels which are estimated based on the location and the weights of the training samples. These levels are a meaningful rating for the training samples. The higher the noise level the less informative is the target to reproduce the underlying function. We analyze this additional information in the following experiment.

5.2   Identifying Outliers

This experiment is designed to illustrate the advantage of having an individual noise estimate for each observation. A useful property of our approach is that it automatically increases this level if a point is not reflecting the underlying function. Depending on the estimated noise value, we can determine whether the corresponding point should be removed from the training set. The goal of this experiment is to show that this way of identifying and removing outliers can significantly improve the prediction accuracy.

To perform this experiment, we learned the dynamics of the blimp online and manually modified the behavior of the blimp after 50 s by increasing the mass. As soon as 10 new observations were received, we add them to the existing training set. To increase the importance of the subsequent observations, we used (2) with $\Delta = 10$. Then, according to the estimated noise levels $\sigma_1^2, \ldots, \sigma_N^2$ we labeled observations with a value exceeding a given threshold ν as an outlier and removed it from the data set. Throughout this experiment, we used the standard deviation of the noise levels: $\nu = \xi \cdot \sqrt{\sum_{i=1}^{N} \sigma_i^2}$. After that, we learned a standard GP model on the remaining points.

Figure 2 shows the learning rate of the vertical direction Z averaged over multiple runs. Regarding the prediction accuracy, the learning progress based on removing outliers is significantly improvement ($p = 0.05$) compared to the standard GP model which uses the complete data set. Additionally, we evaluated our InGP model with rejected outliers for the second GP and this model performs like the standard GP model with removed outliers. This may be based on the fact that we only took the predictive mean function into account. To have a baseline, we additionally trained a GP model based only on observations that correspond to the currently correct model. The prediction of this model can be regarded as optimal. In a second experiment, we evaluated final prediction accuracies after 200 s for different values of ξ. The results are shown in Table 2. As can be seen, the improvement is robust against variations of ξ. In our experiment, keeping 95% of the points on average lead to the best behavior.
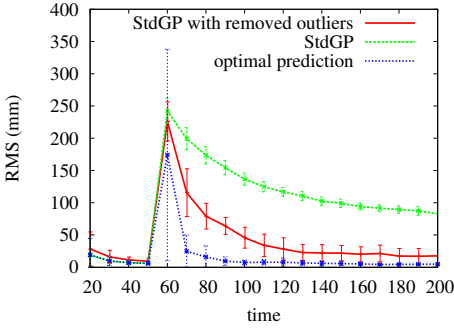
Table 2. Prediction accuracies of the system dynamics using different thresholds ξ to identify outliers

model	RMS$_{(mm)}$
StdGP	83.6 $\pm$ 5.6
removing outliers, $\xi = 1.5$	22.4 $\pm$ 12.2
removing outliers, $\xi = 2.0$	18.7 $\pm$ 12.1
removing outliers, $\xi = 2.5$	20.4 $\pm$ 12.4
removing outliers, $\xi = 3.0$	21.6 $\pm$ 12.3
optimal prediction	5.7 $\pm$ 2.7

Fig. 2. Prediction accuracies plotted over time using $\xi = 2.0$

5.3 Benchmark Test

We also evaluated the performance of our approach based on different data sets frequently used in the literature. We trained our GP model with a homoscedastic as well as a heteroscedastic noise assumption of the data and compared the prediction accuracy to the corresponding state-of-the-art approaches. We assigned uniform weights to our model as the underlying function of each data set is stationary. For each data set, we performed 20 independent runs. In each run, we separated the data into 90% for training and 10% for testing and calculated the negative log predictive density $NLPD = \frac{1}{N} \sum_{i=1}^{N} -logP(y_i \mid \boldsymbol{x}_i, \mathcal{D}, \boldsymbol{\theta})$. Table 3 shows typical results for two synthetic data sets $\mathcal{A}$ and $\mathcal{B}$; and a data set of a simulated motor-cycle crash $\mathcal{C}$, which are introduced in detail in [2], [14], and [15], respectively. As can be seen, our model with individual noise levels achieves an equivalent prediction accuracy compared to the alternative approaches.

Table 3. Evaluation of GP models with different noise assumptions

data set	homoscedastic noise		heteroscedastic noise	
	StdGP	InGP	HetGP	InGP
$\mathcal{A}$	1.506 $\pm$ 0.263	1.491 $\pm$ 0.255	1.455 $\pm$ 0.317	1.445 $\pm$ 0.276
$\mathcal{B}$	1.834 $\pm$ 0.245	1.827 $\pm$ 0.262	1.496 $\pm$ 0.279	1.512 $\pm$ 0.269
$\mathcal{C}$	4.528 $\pm$ 0.189	4.515 $\pm$ 0.233	4.315 $\pm$ 0.474	4.277 $\pm$ 0.530

6   Conclusions

In this paper we presented a novel approach to increase the accuracy of the predicted Gaussian process model based on a non-stationary underlying function. Using individual, uncorrelated noise levels the uncertainty of outdated observation is increased. Our approach is an extension of previous models and easy to

implement. In several experiments, in which we learned the system dynamics of a miniature blimp robot, we show that the prediction accuracy is improved significantly. Furthermore, we showed that our approach, when applied to data sets coming from stationary underlying functions, performs as good as standard Gaussian process models.

References

1. Rasmussen, C.E., Williams, C.: Gaussian Processes for Machine Learning. MIT Press, Cambridge (2006)
2. Goldberg, P., Williams, C., Bishop, C.: Regression with input-dependent noise: A Gaussian process treatment. In: Proc. of the Advances in Neural Information Processing Systems, vol. 10, pp. 493–499. MIT Press, Cambridge (1998)
3. Kersting, K., Plagemann, C., Pfaff, P., Burgard, W.: Most likely heteroscedastic Gaussian process regression. In: Proc. of the Int. Conf. on Machine Learning, pp. 393–400 (2007)
4. Schölkopf, B., Smola, A., Williamson, R., Bartlett, P.: New support vector algorithms. Neural Computation 12, 1207–1245 (2000)
5. Bishop, C., Quazaz, C.: Regression with input-dependent noise: A bayesian treatment. In: Proc. of the Advances in Neural Information Processing Systems, vol. 9, pp. 347–353. MIT Press, Cambridge (1997)
6. Ko, J., Klein, D., Fox, D., Hähnel, D.: Gaussian processes and reinforcement learning for identification and control of an autonomous blimp. In: Proc. of the IEEE Int. Conf. on Robotics & Automation, pp. 742–747 (2007)
7. Rottmann, A., Plagemann, C., Hilgers, P., Burgard, W.: Autonomous blimp control using model-free reinforcement learning in a continuous state and action space. In: Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems, pp. 1895–1900 (2007)
8. Deisenroth, M., Rasmussen, C., Peters, J.: Gaussian process dynamic programming. Neurocomputing 72(7-9), 1508–1524 (2009)
9. D'Souza, A., Vijayakumar, S., Schaal, S.: Learning inverse kinematics. In: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 298–303 (2001)
10. Sundararajan, S., Keerthi, S.: Predictive approaches for choosing hyperparameters in Gaussian processes. Neural Computation 13(5), 1103–1118 (2001)
11. Sugiyama, M., Krauledat, M., Müller, K.R.: Covariate shift adaptation by importance weighted cross validation. J. Mach. Learn. Res. 8, 985–1005 (2007)
12. Rottmann, A., Sippel, M., Zitterell, T., Burgard, W., Reindl, L., Scholl, C.: Towards an experimental autonomous blimp platform. In: Proc. of the 3rd Europ. Conf. on Mobile Robots, pp. 19–24 (2007)
13. Zufferey, J., Guanella, A., Beyeler, A., Floreano, D.: Flying over the reality gap: From simulated to real indoor airships. Autonomous Robots 21(3), 243–254 (2006)
14. Yuan, M., Wahba, G.: Doubly penalized likelihood estimator in heteroscedastic regression. Statistics and Probability Letter 69(1), 11–20 (2004)
15. Silverman, B.: Some aspects of the spline smoothing approach to non-parametric regression curve fitting. Journal of the Royal Statistical Society 47(1), 1–52 (1985)