Uncertainty Disentanglement with Non-stationary Heteroscedastic Gaussian Processes for Active Learning

Zeel B Patel
IIT Gandhinagar, India

Nipun Batra
IIT Gandhinagar, India

Kevin Murphy
Google, USA

Abstract

Gaussian processes are Bayesian non-parametric models used in many areas. In this work, we propose a Non-stationary Heteroscedastic Gaussian process model which can be learned with gradient-based techniques. We demonstrate the interpretability of the proposed model by separating the overall uncertainty into aleatoric (irreducible) and epistemic (model) uncertainty. We illustrate the usability of derived epistemic uncertainty on active learning problems. We demonstrate the efficacy of our model with various ablations on multiple datasets.

1 Introduction

Gaussian processes (GPs) are Bayesian non-parametric models useful for many real-world regression and classification problems. The key object required to define a GP is the kernel function $\mathcal{K}(\mathbf{x}, \mathbf{x}')$, which measures the similarity of the input points. A common choice is the RBF kernel $\mathcal{K}(\mathbf{x}, \mathbf{x}'; \theta) = \sigma^2 \exp\left(-\frac{||\mathbf{x}-\mathbf{x}'||^2}{2\ell^2}\right)$ where ℓ is the length scale, and σ^2 is the signal variance. In regression problems, we also often have observation noise with variance ω^2. These three hyper-parameters, $\theta = (\ell, \sigma, \omega)$, are often learned by optimizing the negative log marginal likelihood.

However, this model uses a stationary kernel (depended only on the distance between locations) and homoskedastic noise (constant noise variance (ω^2)), and these assumptions might not hold in real-life applications such as environment modeling [1, 2]. In particular, non-stationary kernels are necessary if the similarity of two inputs may depend on their location in the input space. Similarly, heteroskedastic noise may be necessary if the quality of the measurements may vary across space.

In this short paper, we provide a computationally efficient way to create GPs with non-stationary kernels, and heteroskedastic noise, by using a Gibbs kernel [3]. Gibbs kernel can be considered the generalization of the RBF kernel where the hyper-parameters are input-dependent, i.e., $\theta(\mathbf{x}) = (\ell(\mathbf{x}), \sigma(\mathbf{x}), \omega(\mathbf{x}))$. These three hyper-parameter functions are themselves represented by a "latent" GP. In contrast to prior work, which uses EP [4] or HMC [5], we use inducing point approximations to approximate the latent GP (which is needed to evaluate the kernel). In addition, we show how modeling variation in all three hyper-parameters allows us to distinguish locations where the latent function value is uncertain (epistemic uncertainty), as opposed to locations where the observation noise is high (aleatoric uncertainty). This distinction is crucial for problems such as active learning and efficient sensor placement. (c.f. [6]). Our experiments can be found here[1]

[1] `https://github.com/patel-zeel/GP_Neurips_22`

2022 NeurIPS Workshop on Gaussian Processes, Spatiotemporal Modeling, and Decision-making Systems.

2 Methods

2.1 Non-stationary Heteroscedastic Gaussian processes

Given observations $\mathbf{y} \in \mathbb{R}^N$ at inputs $X \in \mathbb{R}^{N \times D}$, we assume the following model:

$$y(\mathbf{x}) = f(\mathbf{x}) + \varepsilon(\mathbf{x}), \quad \varepsilon(\mathbf{x}) \sim \mathcal{N}\left(0, \omega(\mathbf{x})^2\right) \tag{1}$$

$$f(\mathbf{x}) \sim GP(0, \mathcal{K}_f(\mathbf{x}, \mathbf{x}')) \tag{2}$$

where the kernel function is $\mathcal{K}(\mathbf{x}, \mathbf{x}') = \text{cov}(f(\mathbf{x}), f(\mathbf{x}'))$ and $\varepsilon(\mathbf{x})$ is zero mean noise. We use the following non-stationary kernel function [3]:

$$\mathcal{K}_f(\mathbf{x}, \mathbf{x}') = \sigma(\mathbf{x})\sigma(\mathbf{x}') \prod_{i=1}^{D} \sqrt{\frac{2\ell_i(x_i)\ell_i(x_i')}{\ell_i(x_i)^2 + \ell_i(x_i')^2}} \exp\left(-\frac{(x_i - x_i')^2}{\ell_i(x_i)^2 + \ell_i(x_i')^2}\right) \tag{3}$$

We assume all hyperparameters of the model, $\theta(\mathbf{x}) = (\ell_{1:D}(\mathbf{x}), \sigma(\mathbf{x}), \omega(\mathbf{x}))$, may be input dependent, to allow for non-stationarity and heteroscedasticity (Previous work [5] has shown that such a kernel is a valid positive semi-definite kernel). We assume these "hyper-functions" $h(\mathbf{x})$ (where $h(\cdot)$ represents either $\ell_i(\cdot)$, $\sigma(\cdot)$ or $\omega(\cdot)$) are smooth and model them by a latent GP on the log scale:

$$\tilde{h}(\mathbf{x}) \sim GP(\tilde{\mu}_h, \mathcal{K}_h(\mathbf{x}, \mathbf{x}'; \tilde{\ell}_h, \tilde{\sigma}_h)) \tag{4}$$

where $\tilde{h}(\cdot) = \log h(\cdot)$. These latent GPs are characterized by a constant mean $\tilde{\mu}_h$ and RBF kernels with parameters $(\tilde{\ell}_h, \tilde{\sigma}_h)$. (We assume noise-free latent functions, so $\omega_h = 0$.)[2] So each of the $D + 2$ hyper-functions h has hyper-parameters $\phi_h = \{\tilde{\mu}_h, \tilde{\ell}_h, \tilde{\sigma}_h\}$ for $h \in \{\ell, \sigma, \omega\}$. (We share hyper-parameters for each dimension of ℓ_i.)

To make learning the latent GPs efficient, we will use a set of M (shared) inducing points $\overline{\mathbf{X}} \in \mathbb{R}^{M \times D}$, which we treat as additional hyper-parameters. Let $\overline{\mathbf{z}}_h = \tilde{h}(\overline{\mathbf{X}}) \in \mathbb{R}^M$ be the (log) outputs at these locations for hyper-parameter h. Then we can infer the expected hyper-parameter value at any other location $\mathbf{x}$ using the usual GP prediction formula for the mean:

$$\tilde{h}(\mathbf{x}) = \mathcal{K}_h(\mathbf{x}, \overline{\mathbf{X}})\mathcal{K}_h(\overline{\mathbf{X}}, \overline{\mathbf{X}})^{-1}\overline{\mathbf{z}}_h \tag{5}$$

Then we can compute $\mathcal{K}_f(\mathbf{x}, \mathbf{x}')$ at any pair of inputs using Eq. (3), where $h(\mathbf{x}) = e^{\tilde{h}(\mathbf{x})}$.

2.2 Learning the hyper-parameters

In total, we can have up to $2MD + 2M + 9$ parameters: the inducing inputs $\overline{\mathbf{X}}_{1:M,1:D}$, the predicted length scales at each inducing point for each dimension, $\ell(\overline{\mathbf{X}})_{1:M,1:D}$, the predicted variances at each inducing point, $\sigma(\overline{\mathbf{X}})_{1:M}$ and $\omega(\overline{\mathbf{X}})_{1:M}$, and the 9 latent GP hyper-parameters $\{\tilde{\mu}_h, \tilde{\ell}_h, \tilde{\sigma}_h\}$ for $h \in \{\ell, \sigma, \omega\}$. We denote all the hyper parameters by ϕ, and we let $\phi^X = (\overline{\mathbf{X}}, \phi)$ represent all the model parameters. We can compute a MAP-type II estimate of these parameters by minimizing the following objective using gradient descent:

$$\hat{\phi}^X = \arg\min_{\phi^X} -[\log p(\mathbf{y}|\mathbf{X}, \phi, \overline{\mathbf{X}}) + \log p(\phi)] \tag{6}$$

where $p(\mathbf{y}|\mathbf{X}, \phi)$ is the marginal likelihood of the main GP (integrating out $\mathbf{f} = [f(\mathbf{x}_n)]_{n=1}^N$), and where $p(\phi)$ is the prior. To ensure smoothly varying latent GPs with a large length scale and low variance, we use a Gamma(5, 1) prior for each $\tilde{\ell}_h$ and a Gamma(0.5, 1) prior for each $\tilde{\sigma}_h$. To ease the optimization process, we use a non-centered parameterization of $\overline{\mathbf{z}}_h$ by learning a vector $\boldsymbol{\gamma}_h$ with independent values (prior for $\boldsymbol{\gamma}_h$ is $\mathcal{N}(0, 1)$) and we then deterministically compute $\overline{\mathbf{z}}_h = L\boldsymbol{\gamma}_h + \mu_h$, where, L is Cholesky decomposition of $\mathcal{K}_h(\overline{\mathbf{X}}, \overline{\mathbf{X}})$. We initialize ϕ by sampling from their respective priors, and initialize $\overline{\mathbf{X}}$ by randomly selecting M points from the dataset.

[2] In practice, we use a fixed value of $\omega_h = 10^{-4}$ for numerical stability.

2.3 Active learning

Active learning (see e.g., [7]) uses some measure of uncertainty to decide which points to label so as to learn the underlying function as quickly as possible. This can also be useful for tasks such as deciding where to place air quality (AQ) sensors (see e.g., [2]), where the goal is to infer the underlying AQ values at all spatial locations based on sensors of varying quality.

Often the GP predictive variance is used as the acquisition function. The overall predictive variance, $\text{var}(y(\mathbf{x}))$, is equal to the model or epistemic uncertainty, $\text{var}(f(\mathbf{x})) = \sigma(\mathbf{x})$, plus the observation noise, $\varepsilon(\mathbf{x}) \sim \mathcal{N}(0, \omega(\mathbf{x})^2)$, which is known as aleatoric uncertainty [8]. Aleatoric uncertainty is irreducible, and thus querying data-points where it is high is not likely to improve the model (c.f., [9]). Thus, we use the epistemic uncertainty $\text{var}(f(\mathbf{x}))$ for active learning. (We note that separating epistemic and aleatoric uncertainty is harder with other kinds of probabilistic model, such as Bayesian neural networks, c.f., [10, 11]).

3 Experiments

In this section, we experimentally compare our method to various baselines on various datasets.

Regression In Table 1, we compare various ablations of our method on various small regression tasks. In particular, we consider making one or more of the hyper-parameters corresponding to length scale (ℓ), variance (σ), and observation noise (ω) either input-dependent (using a latent GP) or fixed constants to be learned The metrics are averaged over five-fold cross-validation using the Adam optimizer (rate=0.05, epochs=1000). We use the following datasets for experiments: 1) Motorcycle helmet: It is a real-life non-stationary, heteroscedastic dataset used in related studies [5]; 2) NONSTAT-2D: It is a non-stationary 2D dataset simulated by input-depended lengthscale [12]. We add linearly increasing heteroscedastic noise to NONSTAT-2D to make it heteroscedastic; and 3) Jump1D: This is 1D dataset taken from [5] (mentioned as J in [5]), which is highly discontinuous from the center. In general, we get better results by making all these parameters input-dependent. We generate a dataset SYNTH-1D from a generative process of a GP where all three hyperparameters ($\ell(x), \sigma(x), \omega(x)$) are input-depended. We use 200 equally spaced inputs in range (-30, 30) with the following deterministic trends: $\ell(x) = 0.5\sin(x/8) + 1.5, \sigma(x) = 1.5\exp(\sin(0.2x)), \omega(x) = 2.5\log(1 + \exp(\sin(-0.2x)))$. We show in Fig. 1 that we are able recover the trends of underlying hyperparameter functions. We illustrate fit over a few more datasets in Appendix Section A Fig. 3.

	Jump		Motorcycle		NONSTAT-2D	
Model	NLPD	RMSE	NLPD	RMSE	NLPD	RMSE
Stationary Homoskedastic GP	4.98	0.26	11.96	0.44	-50.72	0.09
(ℓ)-GP	5.01	0.26	11.92	0.44	-65.13	0.06
(ω)-GP	3.82	0.22	5.21	0.44	-50.81	0.09
(σ)-GP	0.92	0.30	11.56	0.44	-56.66	0.07
(ℓ, ω)-GP	5.01	0.26	5.68	0.45	-65.31	0.06
(ℓ, σ)-GP	-2.18	0.22	11.54	**0.44**	-49.28	0.07
(σ, ω)-GP	0.92	0.22	4.21	0.46	-54.35	0.10
(ℓ, σ, ω)-GP	**-2.20**	**0.22**	**4.09**	0.45	**-73.74**	**0.05**

Table 1: Quantitative results on various datasets. Metrics are as follows (lower is better): Negative Log Predictive Density (NLPD), Root Mean Squared Error (RMSE). The rows represent different methods in which we either used fixed or input-dependent length scale ℓ, signal variance σ and observation noise ω. N-NONSTAT-2D, and other datasets are from [5]. (ℓ, σ, ω)-GP is the best or the second best across all datasets and metrics.

Active learning In Section 2.3, we argued that $\text{var}(f(\mathbf{x}))$ is a better measure of uncertainty compared to $\text{var}(y(\mathbf{x}))$, when performing active learning. We illustrate this point on a 1d synthetic example in Fig. 2. We first train the model on 30 initial train points and then select 50 points with active learning with the following acquisition functions: c) $\text{var}(y(\mathbf{x}))$ and d) $\text{var}(f(\mathbf{x}))$. Note that we do not retrain the hyperparameters of GP during active learning to fasten the process. We empirically (b) and visually (c, d) show that points chosen by $\text{var}(f(\mathbf{x}))$ are closer to the real function.

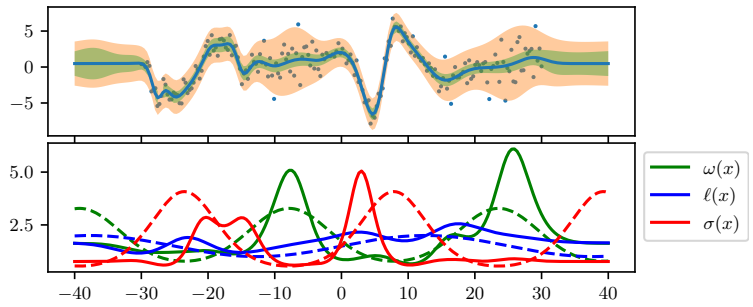

Figure 1: Our model fit on SYNTH-1D data (top) and recovered hyperparameter trends (bottom). Dotted lines show the true functions and continuous lines show the learned functions.

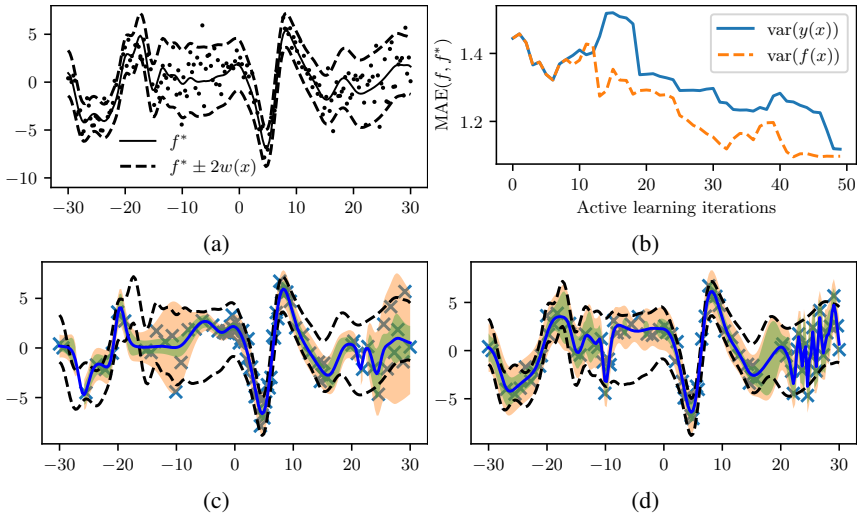

Figure 2: Fit after sampling 50 points from SYNTH-1D with active learning using (c) overall uncertainty; (d) epistemic uncertainty. We show in (b) that Mean Squared Error (MAE) between predicted function f and ground truth f^* is improving faster in (d) with epistemic uncertainty as compared to (c) overall uncertainty. Initial training was done on 30 points. While using epistemic uncertainty, we can capture better points that help GP learn a better fit. Black dots in (a) are data points. Green and orange regions show epistemic and overall uncertainty, respectively.

4 Related work

Multiple approaches to create non-stationary kernels include the multivariate generalization of the Gibbs kernel in [13], sparse spectral kernels [14, 15], the post-processing method of [16], deep kernel learning [17], and deep GPs [18]. In this paper, we adopt the Gibbs kernel approach, where the kernel parameters are themselves generated by latent GPs, as used in prior work [5]. The main difference is in the model fitting procedure. They optimize the latent hyper-parameters ϕ by grid search, and they compute MAP estimates of $\tilde{\ell} = [\tilde{\ell}(\mathbf{x}_n)]_{n=1}^N$, $\tilde{\sigma} = [\tilde{\sigma}(\mathbf{x}_n)]_{n=1}^N$, and $\tilde{\omega} = [\tilde{\omega}(\mathbf{x}_n)]_{n=1}^N$ using gradient descent. Thus they are optimizing a total of $3N+9$ parameters, namely the outputs of the three latent hyper-functions, and the latent hyper-parameters. By contrast, we optimize $2M(D+1)+9$ parameters. For the problems of interest to us (spatio-temporal modeling), D is often low dimensional, so $2M(D+1) \ll 3N$, which makes our approach faster and less prone to overfitting. An additional advantage of our approach is that it is easy to make predictions on new data since we can compute the kernel $\mathcal{K}$ for any pair of inputs. By contrast, in [5], they have to use a heuristic to extrapolate the latent hyper-functions to the test inputs, before using them to compute the kernel values at those locations.

References

[1] Paul D Sampson and Peter Guttorp. Nonparametric estimation of nonstationary spatial covariance structure. *Journal of the American Statistical Association*, 87(417):108–119, 1992.

[2] Zeel B Patel, Palak Purohit, Harsh M Patel, Shivam Sahni, and Nipun Batra. Accurate and scalable gaussian processes for Fine-Grained air quality inference. *AAAI*, 36(11):12080–12088, June 2022.

[3] Mark N Gibbs. *Bayesian Gaussian processes for regression and classification*. PhD thesis, Citeseer, 1998.

[4] Ville Tolvanen, Pasi Jylänki, and Aki Vehtari. Expectation propagation for nonstationary heteroscedastic gaussian process regression. In *2014 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6, September 2014.

[5] Markus Heinonen, Henrik Mannerström, Juho Rousu, Samuel Kaski, and Harri Lähdesmäki. Non-stationary gaussian process regression with Hamiltonian Monte Carlo. In *Artificial Intelligence and Statistics*, pages 732–740. PMLR, 2016.

[6] A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in Gaussian processes: Theory, efficient algorithms and empirical studies. *JMLR*, 9:235–284, 2008.

[7] Burr Settles. Active learning literature survey. 2009.

[8] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. *Mach. Learn.*, 110(3):457–506, March 2021.

[9] Ian Osband. Risk versus uncertainty in deep learning: Bayes, bootstrap and the dangers of dropout. In *NIPS workshop on Bayesian deep learning*, 2016.

[10] Matias Valdenegro-Toro and Daniel Saromo. A deeper look into aleatoric and epistemic uncertainty disentanglement. In *CVPR Workshop on LatinX in CV*, April 2022.

[11] Stefan Depeweg, Jose-Miguel Hernandez-Lobato, Finale Doshi-Velez, and Steffen Udluft. Decomposition of uncertainty in Bayesian deep learning for efficient and risk-sensitive learning. In Jennifer Dy and Andreas Krause, editors, *ICML*, volume 80 of *Proceedings of Machine Learning Research*, pages 1184–1193. PMLR, 2018.

[12] Christian Plagemann, Kristian Kersting, and Wolfram Burgard. Nonstationary gaussian process regression using point estimates of local smoothness. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 204–219. Springer, 2008.

[13] Christopher J Paciorek and Mark J Schervish. Spatial modelling using a new class of nonstationary covariance functions. *Environmetrics*, 17(5):483–506, 2006.

[14] Sami Remes, Markus Heinonen, and Samuel Kaski. Non-Stationary spectral kernels. In *NIPS*, May 2017.

[15] Tompkins, Oliveira, and Ramos. Sparse spectrum warped input measures for nonstationary kernel learning. In *NIPS*, 2020.

[16] Mark D Risser and Daniel Turek. Bayesian inference for high-dimensional nonstationary gaussian processes. *J. Stat. Comput. Simul.*, 90(16):2902–2928, November 2020.

[17] Andrew Gordon Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P Xing. Deep kernel learning. In *AIStats*, pages 370–378, 2016.

[18] Kalvik Jakkala. Deep gaussian processes: A survey. 2021.

A Model fit on various datasets

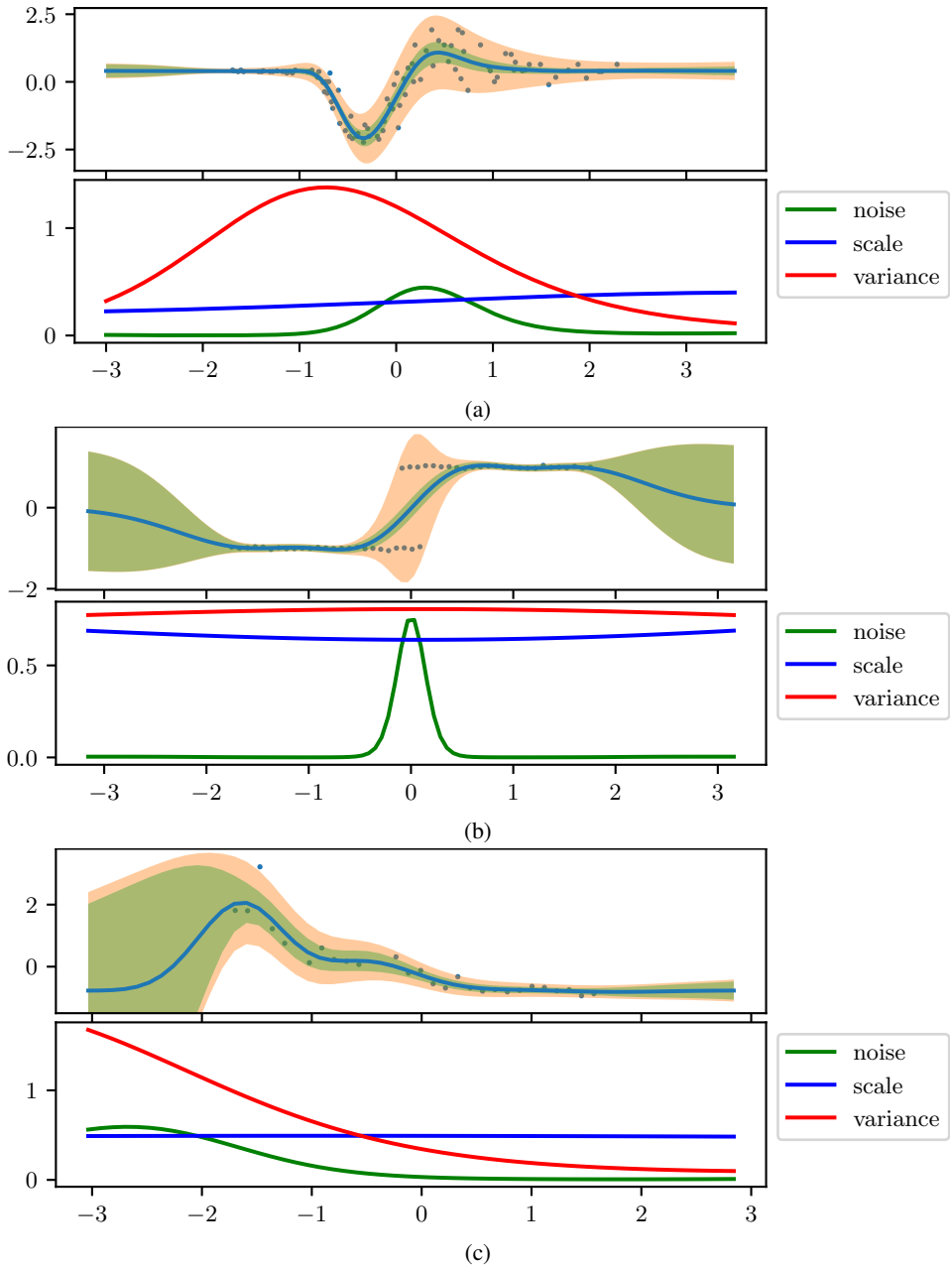

Figure 3: Fitted (ℓ, σ, ω)-GP to a) motorcycle helmet data; b) step data and c) Olympic 100-m race data. Top: predicted distribution. Bottom: learned latent GPs for the 3 hyper-parameters: noise (ω), scale (ℓ), variance (σ). The green region is 95% epistemic variance var$(f(\mathbf{x}))$ and orange region is overall 95% confidence.