Noname manuscript No.
(will be inserted by the editor)

Distributed Robust Gaussian Process Regression

Sebastian Mair · Ulf Brefeld

Received: 17 Sep 2016 / Revised: 21 Apr 2017 / Accepted: 29 May 2017

Abstract We study distributed and robust Gaussian Processes where robustness is introduced by a Gaussian Process prior on the function values combined with a Student-t likelihood. The posterior distribution is approximated by a Laplace Approximation and together with concepts from Bayesian Committee Machines, we efficiently distribute the computations and render robust GPs on huge data sets feasible. We provide a detailed derivation and report on empirical results. Our findings on real and artificial data show that our approach outperforms existing baselines in the presence of outliers by using all available data.

Keywords Robust regression · Gaussian Process regression · Student-t likelihood · Laplace Approximation · Distributed computation

1 Introduction

Gaussian Processes (Rasmussen and Williams 2006) (GPs) are the method of choice for many real-world regression problems. Being non-parametric models, they do not rely on low-level assumptions like the class of the function to be inferred, adapt accurately to the data at-hand and additionally provide handy confidence bounds when it comes to interpreting the results.

Their computation however is quite demanding; e.g., training and prediction scales cubically and quadratically, respectively, in the number of training instances. Thus, applying Gaussian Processes to state-of-the-art data sets is simply infeasible. In addition, real data is usually distorted by extreme observations that are often considered outliers or anomalies. These observations may arise from various sources such as broken sensors or transcription errors and deteriorate the predictive performance of GPs due to the Gaussian likelihood model which is not able to reject extreme observations (O'Hagan 1979).

Scaling and robustness issues of Gaussian Processes have been studied for some time and remedies have been proposed. A straight forward idea is to use only a sparse set instead of the entire data set to compute the GPs (Quiñonero-Candela and Rasmussen 2005; Titsias 2009; Hensman et al 2013; Gal et al 2014). However, these so-called sparse approximations come with other issues. These, for example, include a sampling strategy to assemble a small subset from very many data points, the optimization of inducing data points which are not part of the original

✉ Sebastian Mair · Ulf Brefeld
Leuphana University of Lüneburg, Germany
E-mail: {sebastian.mair,ulf.brefeld}@leuphana.de

data as well as understanding the trade-off between the quality of the approximation and the size of that training set. It also questions storing all data in the first place as the better part is eventually ignored anyway. Alternative, and perhaps more appropriate, approaches exploit the independence of local GP experts on random partitions of the data and distribute the computation to several machines (Tresp 2000; Cao and Fleet 2014; Deisenroth and Ng 2015).

Robustness, on the other hand, can be obtained by two-component mixture noise models (Jaynes and Bretthorst 2003; Naish-Guzman and Holden 2008) or heavy-tailed observation models like Laplace (Kuss 2006) and Student-t distributions (Neal 1997; Vanhatalo et al 2009; Jylänki et al 2011). Naturally, these modifications come at a cost. For instance, they often lead to non-Gaussian posterior distributions and integrals for the predictive distribution cannot be solved analytically anymore. Consequentially, one has either to resort to time-consuming sampling approaches or (possibly inaccurate) approximations.

In this paper, we leverage ideas from Deisenroth and Ng (2015) and Vanhatalo et al (2009) and present a distributed and robust Gaussian Process regression that is based on Student-t observation distributions. We propose an efficient distributed computation scheme that works on independent (i.e., distributed) subsets to process all available data. Inference is performed by an efficient Laplace Approximation that proves as effective as variational approaches and Markov Chain Monte Carlo strategies when using Student-t observation models (Vanhatalo et al 2009). We provide a detailed derivation of the proposed approach and empirically compare its performance with vanilla and distributed Gaussian Processes. Our findings show that the proposed approach significantly outperforms its peers in predictive performance while having a run-time that depends only on the number of available machines.

The remainder is structured as follows. Section 2 reviews related work and Section 3 introduces basic concepts. The main contribution is resented in Section 4. Section 5 reports on our empirical results and Section 6 concludes.

2 Related Work

A prominent technique to scale up Gaussian Process regression is to use a sparse approximation. The idea is to focus on only a small set of size $m \ll n$. This effectively lowers the computational complexity to $\mathcal{O}(m^3)$ (Hensman et al 2013) or $\mathcal{O}(nm^2)$ (Quiñonero-Candela and Rasmussen 2005; Titsias 2009), depending on the respective approach. Although sparse approximations can be distributed using the Map-Reduce architecture (Gal et al 2014), choosing m and selecting or optimizing representative candidates for a problem at-hand is difficult and often requires complex additional computations.

Alternative approaches study distributing Gaussian Process regression to leverage all available data. By distributing the computations as well as the data, the computational power of computer clusters is effectively exploited and only the hardware constrains the processable quantities of data. The underlying idea of the Product-of-Experts (PoE) family grounds on factorizations of the likelihood by exploiting inherent independence assumptions (Deisenroth and Ng 2015). The data $\mathcal{D}$ is accordingly split into $M \in \mathbb{N}$ disjoint parts $\mathcal{D}^{(k)}$ of (roughly) equal size n_k where $1 \leq k \leq M$, and a GP is trained on every split $\mathcal{D}^{(k)}$ where hyperparameters $\boldsymbol{\theta}$ are shared among all models. By doing so, a (practically infeasible) global model is being approximated.

Several different approaches have been studied. Cao and Fleet (2014) propose generalized-Product-of-Experts (gPoE) where independent local GPs are weighted according to the difference in entropy between prior and posterior for a given test point, before merging individual predictions into a joint one. The Bayesian Committee Machine (BCM) (Tresp 2000) assumes that the subsets are conditionally independent given the function values. Furthermore, it directly incorporates the Gaussian Process prior into the prediction to be able to fall back to prior belief in regions that are far away from the training data. Standard (g)PoE models fail to use prior information in those areas or require additional normalizations (Deisenroth and Ng 2015), while gPoE tends to over-estimate variances near training instances and frequently acts too conservative. The BCM on the other hand performs similar to a full GP near training instances but the model suffers from weak experts in terms of the predictive mean estimates.

To overcome these weaknesses Deisenroth and Ng (2015) introduce robust Bayesian Committee Machines (rBCM) by merging the generalized-Product-of-Experts with Bayesian Committee Machines. Their approach constitutes a generalized unification and includes the former two as special cases. By using the rBCM, one is able to distribute all available data as well as the computations on a computer cluster. Furthermore, all computations are straightforward and can be performed analytically. Note that the term *robust* Bayesian Committee Machines is somewhat misleading in our context as *robust* does not imply robustness against outliers but refers to the ability of performing consistent predictions.

Robustness is usually achieved by using a heavy-tailed likelihood for instance realized by Cauchy, Laplace or Student-*t* distributions. Another way of constructing heavy tails is to use a two-component mixture noise model (Jaynes and Bretthorst 2003; Naish-Guzman and Holden 2008); with high probability, an observation is considered regular according to a Gaussian distribution with moderate variance while there is a small probability that renders an instance an outlier that has been drawn from a Gaussian possessing a much higher variance.

Robust regression using a Laplace observation model is used in Kuss (2006). By choosing a non-Gaussian observation model the posterior distribution in Gaussian Process regression is no longer analytically tractable and approximate inference is needed. Kuss (2006) describes a Markov Chain Monte Carlo as well as an Expectation Propagation method for approximate inference.

The Student-*t* distribution is another common choice when studying robust observation models (Jylänki et al 2011). Together with GPs, a great variety of approximate inference strategies has been proposed for Student-*t* observation models, including Markov Chain Monte Carlo (MCMC) (Geweke 1993; Neal 1997), Laplace Approximations (LA) (Vanhatalo et al 2009), Factorizing Variational Approximations (Tipping and Lawrence 2005; Kuss 2006), Variational Bounds (Nickisch and Rasmussen 2008), and Expectation Propagation (Jylänki et al 2011).

3 Preliminaries

3.1 Problem Setting

We study noisy regression problems with target variables $y_i = f(\boldsymbol{x}_i) + \varepsilon_i \in \mathbb{R}$ where $i = 1, 2, \ldots, n$. The goal is to infer the latent function $f : \mathbb{R}^d \to \mathbb{R}$ given some training data $\mathscr{D} = (\boldsymbol{X}, \boldsymbol{y})$ of size $n \in \mathbb{N}$, where the d-dimensional inputs $\boldsymbol{x}_i \in \mathbb{R}^d$ are aggregated into a design matrix $\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^n$ and the noisy observations are stacked into a n-dimensional vector $\boldsymbol{y} = \{y_i\}_{i=1}^n$. Furthermore, the latent function values $f_i = f(\boldsymbol{x}_i)$ will be stacked into a n-dimensional vector $\boldsymbol{f} = \{f(\boldsymbol{x}_i)\}_{i=1}^n$. Typically, the noise term is assumed to be i.i.d. Gaussian, i.e., $\varepsilon_i \sim \mathscr{N}(0, \sigma_\varepsilon^2)$, centered at zero with a static variance of σ_ε^2. In the presence of outliers, some targets y_i take on extreme values and lie far away from the remaining normal ones.

3.2 Gaussian Processes

A Gaussian Process (GP) is a collection of random variables, any finite number of which have a joint Gaussian distribution. Because of the properties of a Gaussian distribution, a GP is completely specified by its mean $m(\boldsymbol{x})$ and covariance or kernel function k. Without loss of generality, we will assume a zero mean function, i.e., $m(\boldsymbol{x}) = \boldsymbol{0}$. A common choice for the kernel function is the squared exponential kernel, which is given by

$$k(\boldsymbol{x}_p, \boldsymbol{x}_q) = \sigma_f^2 \exp\left(-\frac{1}{2}(\boldsymbol{x}_p - \boldsymbol{x}_q)^\top \boldsymbol{\Lambda} (\boldsymbol{x}_p - \boldsymbol{x}_q) \right). \tag{1}$$

In this paper we will focus on $\boldsymbol{\Lambda} = \operatorname{diag}(\ell_1^{-2}, \ldots, \ell_d^{-2})$ as it implements an automatic relevance determination (ARD) (Rasmussen and Williams 2006). A GP is trained by optimizing its hyperparameters $\boldsymbol{\theta}$, that is, $\boldsymbol{\theta} = (\ell_1^2, \ldots, \ell_d^2, \sigma_f^2, \sigma_\varepsilon^2)$. This can be done, for instance, by maximizing the log marginal likelihood

$$\boldsymbol{\theta}^\star = \arg\max_{\boldsymbol{\theta}} \ln p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} -\frac{1}{2}\boldsymbol{y}^\top (\boldsymbol{K} + \sigma_\varepsilon^2 \boldsymbol{I})^{-1}\boldsymbol{y} - \frac{1}{2}\ln|\boldsymbol{K} + \sigma_\varepsilon^2 \boldsymbol{I}| - \frac{n}{2}\ln 2\pi,$$

where $\boldsymbol{K}$ denotes the $n \times n$ kernel matrix given by $K_{pq} = k(\boldsymbol{x}_p, \boldsymbol{x}_q)$. After training, the prediction $f(\boldsymbol{x}_*)$ of a test input $\boldsymbol{x}_* \in \mathbb{R}^d$ can be inferred using the posterior predictive distribution that is given by

$$p(f_*|\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{x}_*) = \int p(f_*|\boldsymbol{X}, \boldsymbol{x}_*, \boldsymbol{f}) \cdot p(\boldsymbol{f}|\boldsymbol{X}, \boldsymbol{y})\, \mathrm{d}\boldsymbol{f}, \tag{2}$$

where the posterior over the latent variables is

$$p(\boldsymbol{f}|\boldsymbol{X}, \boldsymbol{y}) = \frac{p(\boldsymbol{y}|\boldsymbol{f}) \cdot p(\boldsymbol{f}|\boldsymbol{X})}{p(\boldsymbol{y}|\boldsymbol{X})} \propto \underbrace{p(\boldsymbol{y}|\boldsymbol{f})}_{\text{likelihood}} \cdot \underbrace{p(\boldsymbol{f}|\boldsymbol{X})}_{\text{prior}}. \tag{3}$$

With a Gaussian prior $p(\boldsymbol{f}|\boldsymbol{X})$ and likelihood $p(\boldsymbol{y}|\boldsymbol{f})$, the posterior remains Gaussian and the computation is straightforward. The predictive distribution in Equation (2) is then fully specified by the mean and variance estimates given by

$$\mathbb{E}[f_*] = \boldsymbol{k}_*^\top (\boldsymbol{K} + \sigma_\varepsilon^2 \boldsymbol{I})^{-1}\boldsymbol{y}, \tag{4}$$

$$\mathbb{V}[f_*] = k_{**} - \boldsymbol{k}_*^\top (\boldsymbol{K} + \sigma_\varepsilon^2 \boldsymbol{I})^{-1}\boldsymbol{k}_*, \tag{5}$$

respectively, where $\boldsymbol{k}_* = k(\boldsymbol{X}, \boldsymbol{x}_*)$ and $\boldsymbol{k}_{**} = k(\boldsymbol{x}_*, \boldsymbol{x}_*)$.

3.3 The Student-t Distribution

To obtain robust GPs, we aim to replace the Gaussian observation model $p(\boldsymbol{y}|\boldsymbol{f})$ by a Student-t one. Before we go into details, we briefly introduce the Student-t distribution.

Definition 1 The probability density function of the one-dimensional Student-t distribution (see Gelman et al 2013, page 578ff.) of a random variable ϕ is given by

$$t_\nu(\phi|\mu, \sigma^2) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)\sqrt{\nu\pi\sigma^2}} \left(1 + \frac{1}{\nu}\frac{(\phi - \mu)^2}{\sigma^2}\right)^{-\frac{\nu+1}{2}}, \tag{6}$$

where $\mu \in \mathbb{R}$ is the location parameter, $\sigma \in \mathbb{R}^+$ the scale parameter and $\nu \in \mathbb{R}^+$ the degree of freedom. ■

In the limit $\nu \to \infty$, the Student-t distribution approaches a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ and for $\nu = 1$, the Cauchy distribution is obtained as special case. The Student-t can be seen as a mixture of Gaussian distributions with a common mean and variances distributed as scaled Inv-χ^2 (compare Gelman et al 2013, page 437). Suppose $y_i \sim t_\nu(\mu, \sigma^2)$, then the following system is equivalent:

$$\begin{aligned} y_i|V_i &\sim \mathcal{N}(\mu, V_i), \\ V_i &\sim \text{Inv-}\chi^2(\nu, \sigma^2). \end{aligned} \tag{7}$$

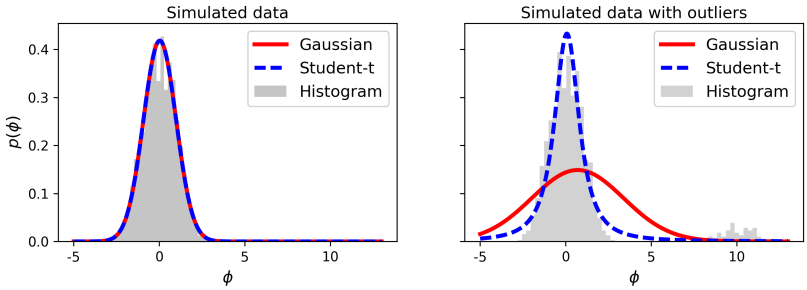

Fig. 1 Illustration of fitting a Gaussian as well as a Student-t distribution to a number of standard normal distributed samples with and without outliers.

This representation will be convenient later on. In contrast to the Gaussian, the Student-t has longer tails which is beneficial in the presence of outliers. Therefore, the Student-t distribution is often used in place of a Gaussian for robust analyses (see Gelman et al 2013, page 582).

To show why Student-t makes sense in outlier-prone scenarios, we define the important properties of outlier-proneness and outlier-resistance and state some implications according to O'Hagan (1979). We begin with a proper definition of outlier-proneness.

Definition 2 Let $y_1,\dots,y_n,y_{n+1}$ be independently and identically distributed. The observation model is defined to be *outlier-prone* of order n, if $p(f|y_1,\dots,y_n,y_{n+1}) \to p(f|y_1,\dots,y_n)$ as $y_{n+1} \to \pm\infty$. ■

The definition says that a new observation y_{n+1} will be taken into account as long as it is consistent with the previous n observations and rejected if it tends towards $\pm\infty$, in which case it is considered an outlier. A distribution satisfying this property is the family of Student-t distributions for $\nu > 0$ (O'Hagan 1979).

Proposition 1 *The Student-t distributions t_ν are outlier-prone of order 1.* ■

If the density is bounded, it follows that outlier-proneness of order n implies outlier-proneness of order $n+1$ and therefore outlier-proneness of order 1 is the strongest property. It can also be shown that up to m outliers can be rejected by an outlier-prone distribution if there are at least $2m$ observations (O'Hagan 1979). The counterpart of outlier-proneness is outlier-resistance. The definition is as follows:

Definition 3 Let $y_1,\dots,y_n,y_{n+1}$ be independently and identically distributed. The observation model is defined to be *outlier-resistant*, if $p(f|y_1,\dots,y_n,y_{n+1})$ is a decreasing function of y_{n+1} for all $n \in \mathbb{N}$ and $y_1,\dots,y_n$. ■

That is, every distribution satisfying the property of outlier-resistance will take positive account of every observation, however extreme it may be. Unfortunately, the well-known Gaussian distribution fulfills this property.

Proposition 2 *The Gaussian distribution is outlier-resistant.* ■

The consequences of Proposition 2 are illustrated in Figure 1 where only a few extreme measurements lead to a dubiously shaped Gaussian. The figure thus also serves as a motivation for using the Student-t distributions instead of Gaussians in outlier-prone scenarios.

However, in case of a Student-t likelihood $p(\boldsymbol{y}|\boldsymbol{f})$, the posterior in Equation (3) is no longer Gaussian and the integral in Equation (2) becomes analytically intractable. Therefore, an approximation is required. We resort to a Laplace Approximation (Vanhatalo et al 2009) for two reasons. Firstly, the resulting distribution is again Gaussian leading to a Gaussian predictive distribution which will turn out convenient in the remainder. Secondly, there is no

significant difference in performance between LA and its peers while LA is the fastest (Vanhatalo et al 2009). In the following section, a Laplace Approximation for the posterior will be derived based on Vanhatalo et al (2009) and Rasmussen and Williams (2006).

3.4 Laplace Approximation of the Posterior

With Laplace's method, the posterior in Equation (3) is approximated with a Gaussian using a second order Taylor expansion around the maximum $\hat{\boldsymbol{f}}$ of the posterior. We have

$$\begin{aligned}\ln p(\boldsymbol{f}|\boldsymbol{X},\boldsymbol{y}) \equiv \Psi(\boldsymbol{f}) &\approx \Psi(\hat{\boldsymbol{f}}) + \underbrace{\nabla\Psi(\hat{\boldsymbol{f}})(\boldsymbol{f}-\hat{\boldsymbol{f}})}_{=0} + \frac{1}{2}(\boldsymbol{f}-\hat{\boldsymbol{f}})^\top \nabla^2\Psi(\hat{\boldsymbol{f}})(\boldsymbol{f}-\hat{\boldsymbol{f}}) \\ &= \Psi(\hat{\boldsymbol{f}}) + \frac{1}{2}(\boldsymbol{f}-\hat{\boldsymbol{f}})^\top \nabla^2\Psi(\hat{\boldsymbol{f}})(\boldsymbol{f}-\hat{\boldsymbol{f}}).\end{aligned}$$

The linear term $\nabla\Psi(\hat{\boldsymbol{f}})(\boldsymbol{f}-\hat{\boldsymbol{f}})$ disappears since the gradient of Ψ is zero at the mode $\hat{\boldsymbol{f}}$. Comparing this with the log-version of a Gaussian displays the similarity,

$$\ln \mathcal{N}(\boldsymbol{f}|\boldsymbol{\mu},\boldsymbol{\Sigma}) = -\frac{1}{2}(\boldsymbol{f}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{f}-\boldsymbol{\mu}) - \frac{1}{2}\ln|\boldsymbol{\Sigma}| - \frac{n}{2}\ln 2\pi.$$

Therefore, by setting $\boldsymbol{\mu} = \hat{\boldsymbol{f}}$ and $\boldsymbol{\Sigma}^{-1} = -\nabla^2\Psi(\hat{\boldsymbol{f}})$ we obtain a Gaussian approximation $q(\boldsymbol{f}|\boldsymbol{X},\boldsymbol{y})$ of the posterior $p(\boldsymbol{f}|\boldsymbol{X},\boldsymbol{y})$. The actual approximation is given by

$$p(\boldsymbol{f}|\boldsymbol{X},\boldsymbol{y}) \approx q(\boldsymbol{f}|\boldsymbol{X},\boldsymbol{y}) = \mathcal{N}(\boldsymbol{f}|\hat{\boldsymbol{f}},\boldsymbol{\Sigma}),$$

with $\hat{\boldsymbol{f}} = \arg\max_{\boldsymbol{f}}\ p(\boldsymbol{f}|\boldsymbol{X},\boldsymbol{y})$ and $\boldsymbol{\Sigma}^{-1} = -\nabla^2 \ln p(\boldsymbol{f}|\boldsymbol{X},\boldsymbol{y})\big|_{\boldsymbol{f}=\hat{\boldsymbol{f}}}$. To obtain these equations, we first consider the log posterior and the definition of $\Psi(\boldsymbol{f})$,

$$\ln p(\boldsymbol{f}|\boldsymbol{X},\boldsymbol{y}) \overset{(3)}{=} \ln \frac{p(\boldsymbol{y}|\boldsymbol{f})\cdot p(\boldsymbol{f}|\boldsymbol{X})}{p(\boldsymbol{y}|\boldsymbol{X})} = \underbrace{\ln p(\boldsymbol{y}|\boldsymbol{f}) + \ln p(\boldsymbol{f}|\boldsymbol{X})}_{:=\Psi(\boldsymbol{f})} - \ln p(\boldsymbol{y}|\boldsymbol{X}).$$

Note, that $\Psi(\boldsymbol{f})$ does not include $\ln p(\boldsymbol{y}|\boldsymbol{X})$ since it does not depend on $\boldsymbol{f}$. Using a Gaussian prior yields

$$\Psi(\boldsymbol{f}) = \ln p(\boldsymbol{y}|\boldsymbol{f}) + \ln p(\boldsymbol{f}|\boldsymbol{X}) = \ln p(\boldsymbol{y}|\boldsymbol{f}) - \frac{1}{2}\boldsymbol{f}^\top \boldsymbol{K}^{-1}\boldsymbol{f} - \frac{1}{2}\ln|\boldsymbol{K}| - \frac{n}{2}\ln 2\pi. \tag{8}$$

Now the first and second order derivatives of $\Psi(\boldsymbol{f})$ can be computed with respect to $\boldsymbol{f}$,

$$\nabla\Psi(\boldsymbol{f}) = \nabla \ln p(\boldsymbol{y}|\boldsymbol{f}) + \nabla \ln p(\boldsymbol{f}|\boldsymbol{X}) = \nabla \ln p(\boldsymbol{y}|\boldsymbol{f}) - \boldsymbol{K}^{-1}\boldsymbol{f}, \tag{9}$$

$$\nabla^2\Psi(\boldsymbol{f}) = \nabla^2 \ln p(\boldsymbol{y}|\boldsymbol{f}) + \nabla^2 \ln p(\boldsymbol{f}|\boldsymbol{X}) = \nabla^2 \ln p(\boldsymbol{y}|\boldsymbol{f}) - \boldsymbol{K}^{-1} = -(\boldsymbol{K}^{-1} + \boldsymbol{W}), \tag{10}$$

where $\boldsymbol{W} = -\nabla^2 \ln p(\boldsymbol{y}|\boldsymbol{f})$ is the negative Hessian of the log likelihood $p(\boldsymbol{y}|\boldsymbol{f})$. Since the likelihood factorizes, $\boldsymbol{W}$ is diagonal and given by

$$W_{ij} = \begin{cases} -(\nu+1)\frac{(y_i-f_i)^2-\nu\sigma^2}{(\nu\sigma^2+(y_i-f_i)^2)^2}, & \text{for } i=j, \\ 0, & \text{otherwise.} \end{cases}$$

To find the maximum $\hat{\boldsymbol{f}}$ of $\Psi(\boldsymbol{f})$, we need to solve

$$\nabla\Psi(\boldsymbol{f}) = \nabla \ln p(\boldsymbol{y}|\boldsymbol{f}) - \boldsymbol{K}^{-1}\boldsymbol{f} \stackrel{!}{=} 0 \iff \hat{\boldsymbol{f}} = \boldsymbol{K}\Big(\nabla \ln p(\boldsymbol{y}|\hat{\boldsymbol{f}})\Big). \tag{11}$$

Equation (11) cannot be solved directly, since $\nabla \ln p(\boldsymbol{y}|\hat{\boldsymbol{f}})$ is a non-linear function of $\hat{\boldsymbol{f}}$. Rasmussen and Williams (2006) state, that Newton's method can be used to find the maximum of $\Psi(\boldsymbol{f})$ whereas Vanhatalo et al (2009) suggest to use the Expectation Maximization (EM) algorithm (Dempster et al 1977) for robustness and efficiency.

The EM algorithm utilizes the scale mixture representation of a Student-t distribution given in Equation (7). The E-step computes the expectations

$$\mathbb{E}\left[\frac{1}{V_i}\middle| y_i, f_i^{\text{old}}, \nu, \sigma\right] = \frac{\nu+1}{\nu\sigma^2 + (y_i - f_i^{\text{old}})^2},$$

that are aggregated in a diagonal matrix $\boldsymbol{V}^{-1}$ and the M-step provides an updated estimate for the mode

$$\hat{\boldsymbol{f}}^{\text{new}} = (\boldsymbol{K}^{-1} + \boldsymbol{V}^{-1})^{-1}\boldsymbol{V}^{-1}\boldsymbol{y}. \tag{12}$$

In practice, inverting the kernel matrix $\boldsymbol{K}$ is prohibitive since the eigenvalues may be close to zero which renders the inversion numerically unstable. Instead, the matrix inversion lemma (Rasmussen and Williams 2006; Vanhatalo et al 2009) is used. We introduce the matrix $\boldsymbol{G}$ to rewrite the inverse in Equation (12) as follows

$$\begin{aligned}(\boldsymbol{K}^{-1} + \boldsymbol{V}^{-1})^{-1} &= \boldsymbol{K} - \boldsymbol{K}(\boldsymbol{K}+\boldsymbol{V})^{-1}\boldsymbol{K} = \boldsymbol{K} - \boldsymbol{K}(\underbrace{\boldsymbol{V}^{\frac{1}{2}}\boldsymbol{V}^{-\frac{1}{2}}}_{=\boldsymbol{I}}\boldsymbol{K}\underbrace{\boldsymbol{V}^{-\frac{1}{2}}\boldsymbol{V}^{\frac{1}{2}}}_{=\boldsymbol{I}} + \underbrace{\boldsymbol{V}^{\frac{1}{2}}\boldsymbol{I}\boldsymbol{V}^{\frac{1}{2}}}_{=\boldsymbol{V}})^{-1}\boldsymbol{K} \\ &= \boldsymbol{K} - \boldsymbol{K}\boldsymbol{V}^{-\frac{1}{2}}(\underbrace{\boldsymbol{V}^{-\frac{1}{2}}\boldsymbol{K}\boldsymbol{V}^{-\frac{1}{2}} + \boldsymbol{I}}_{:=\boldsymbol{G}})^{-1}\boldsymbol{V}^{-\frac{1}{2}}\boldsymbol{K} = \boldsymbol{K} - \boldsymbol{K}\boldsymbol{V}^{-\frac{1}{2}}\boldsymbol{G}^{-1}\boldsymbol{V}^{-\frac{1}{2}}\boldsymbol{K}.\end{aligned} \tag{13}$$

Note that $\boldsymbol{V}$ is a positive diagonal matrix and therefore computing $\boldsymbol{V}^{\pm\frac{1}{2}}$ is straight forward. The matrix $\boldsymbol{G} = \boldsymbol{V}^{-\frac{1}{2}}\boldsymbol{K}\boldsymbol{V}^{-\frac{1}{2}} + \boldsymbol{I}$ is symmetric and positive definite by definition and therefore a Cholesky decomposition $\boldsymbol{H}\boldsymbol{H}^\top = \boldsymbol{G}$ can be applied. Such a decomposition is useful for the computation of $\boldsymbol{G}^{-1} = (\boldsymbol{H}^{-1})^\top\boldsymbol{H}^{-1}$ and $|\boldsymbol{G}| = \prod_i H_{ii}^2$. Using Equation (13) allows to reformulate the M-step as

$$\begin{aligned}\hat{\boldsymbol{f}}^{\text{new}} &\stackrel{(12)}{=} (\boldsymbol{K}^{-1} + \boldsymbol{V}^{-1})^{-1}\boldsymbol{V}^{-1}\boldsymbol{y} \stackrel{(13)}{=} (\boldsymbol{K} - \boldsymbol{K}\boldsymbol{V}^{-\frac{1}{2}}\boldsymbol{G}^{-1}\boldsymbol{V}^{-\frac{1}{2}}\boldsymbol{K})\boldsymbol{V}^{-1}\boldsymbol{y} \\ &= \boldsymbol{K}\boldsymbol{V}^{-1}\boldsymbol{y} - \boldsymbol{K}\boldsymbol{V}^{-\frac{1}{2}}\boldsymbol{G}^{-1}\boldsymbol{V}^{-\frac{1}{2}}\boldsymbol{K}\boldsymbol{V}^{-1}\boldsymbol{y} = \boldsymbol{K}\underbrace{(\boldsymbol{V}^{-1}\boldsymbol{y} - \boldsymbol{V}^{-\frac{1}{2}}\boldsymbol{G}^{-1}\boldsymbol{V}^{-\frac{1}{2}}\boldsymbol{K}\boldsymbol{V}^{-1}\boldsymbol{y})}_{:=\boldsymbol{a}} = \boldsymbol{K}\boldsymbol{a}.\end{aligned} \tag{14}$$

The vector $\boldsymbol{a}$ will serve as an intermediate result for further computations. The target vector $\boldsymbol{y}$ can be used as an initialization of the mode of the latent variables $\hat{\boldsymbol{f}}$. Except for outliers, $\boldsymbol{y}$ it should not be to far away as it is similarly distributed, an additive noise term $\boldsymbol{\varepsilon}$ being the only difference. Finally, the expectation and maximization steps have to be iterated until convergence of $\Psi(\hat{\boldsymbol{f}})$. One problem of finding the mode $\hat{\boldsymbol{f}}$ is, that the Student-t distribution is not log-concave and the posterior may be multimodal. Although choosing a unimodal Laplace Approximation for a possibly multimodal distribution may seem inappropriate, recall that any other unimodal approximation will face the same problem.

4 Distributed Robust Gaussian Process Regression

We now present our main contribution. We devise robust Gaussian Processes that can be applied to large data sets. The idea is to distribute the computations and effectively maintain several distributed independent GPs that are trained jointly on disjoint subsets of the data. The training data $\mathscr{D} = (\boldsymbol{X}, \boldsymbol{y})$ is split into $M \in \mathbb{N}$ smaller data sets $\mathscr{D}^{(k)} = (\boldsymbol{X}^{(k)}, \boldsymbol{y}^{(k)})$ of size $n_k \ll n$, where $k = 1, 2, \ldots, M$. The splits are supposed to be random and the subsets are disjoint and approximately equal in size. Every subset $\mathscr{D}^{(k)}$ is processed by a robust Gaussian Process similar to the previous section. However, the robust GPs are being trained jointly and share the same set of hyperparameters $\boldsymbol{\theta}$. After the training process, like in all other Product-of-Experts models, every GP produces a mean and variance prediction independently for its subset and all predictions will be combined using ideas from robust Bayesian Committee Machines (Deisenroth and Ng 2015). In the remainder of this section, we detail training and prediction phases.

4.1 Training

Exploiting independence allows to approximate the marginal likelihood by the product of M GP experts,

$$p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}) \approx \prod_{k=1}^{M} p_k\big(\boldsymbol{y}^{(k)}|\boldsymbol{X}^{(k)}, \boldsymbol{\theta}\big). \tag{15}$$

For notational simplicity we will drop the superscript (k) but keep the subscript k hereafter. Each of the M experts is defined by the conditional

$$p_k(\boldsymbol{y}|\boldsymbol{X}) = \int p_k(\boldsymbol{y}|\boldsymbol{f}) \cdot p_k(\boldsymbol{f}|\boldsymbol{X})\, \mathrm{d}\boldsymbol{f}. \tag{16}$$

In case of a Gaussian likelihood $p_k(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{f})$ and prior $p_k(\boldsymbol{f}|\boldsymbol{X})$ the product is proportional to a Gaussian and therefore the integral can be computed analytically. Instead, we will again make use of the Student-t likelihood we need to resort to an approximation. We will thus again use the Laplace method from Section 3.4.

For a Student-t likelihood the product inside the integral of Equation (16) is equal to $\exp(\Psi(\boldsymbol{f}))$, compare also with Equation (8). A second order Taylor Approximation of $\Psi(\boldsymbol{f})$ around the maximum $\hat{\boldsymbol{f}}$ yields

$$\begin{aligned}
p_k(\boldsymbol{y}|\boldsymbol{X}) &= \int \exp\Big(\Psi(\boldsymbol{f})\Big) d\boldsymbol{f} \\
&\overset{\text{TA}}{\approx} \int \exp\Big(\Psi(\hat{\boldsymbol{f}}) + \frac{1}{2}(\boldsymbol{f}-\hat{\boldsymbol{f}})^\top \nabla^2\Psi(\hat{\boldsymbol{f}})(\boldsymbol{f}-\hat{\boldsymbol{f}})\Big)\, \mathrm{d}\boldsymbol{f} \\
&= \exp\Big(\Psi(\hat{\boldsymbol{f}})\Big) \cdot \int \exp\Big(+\frac{1}{2}(\boldsymbol{f}-\hat{\boldsymbol{f}})^\top \nabla^2\Psi(\hat{\boldsymbol{f}})(\boldsymbol{f}-\hat{\boldsymbol{f}})\Big)\, \mathrm{d}\boldsymbol{f} \\
&\overset{(10)}{=} \exp\Big(\Psi(\hat{\boldsymbol{f}})\Big) \cdot \int \exp\Big(-\frac{1}{2}(\boldsymbol{f}-\hat{\boldsymbol{f}})^\top (\boldsymbol{K}^{-1}+\boldsymbol{W})(\boldsymbol{f}-\hat{\boldsymbol{f}})\Big)\, \mathrm{d}\boldsymbol{f} \\
&= q_k(\boldsymbol{y}|\boldsymbol{X})
\end{aligned} \tag{17}$$

and Equation (15) can be rewritten as

$$p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}) \approx \prod_{k=1}^{M} q_k\big(\boldsymbol{y}^{(k)}|\boldsymbol{X}^{(k)}, \boldsymbol{\theta}\big). \tag{18}$$

As in standard Gaussian Process regression, we rely on the log version of the marginal likelihood for training as it is easier to optimize,

$$\ln p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{\theta}) \approx \sum_{k=1}^{M} \ln q_k\big(\boldsymbol{y}^{(k)}|\boldsymbol{X}^{(k)},\boldsymbol{\theta}\big). \tag{19}$$

Taking the log on the approximation $q_k(\boldsymbol{y}|\boldsymbol{X})$ yields

$$\ln p_k(\boldsymbol{y}|\boldsymbol{X}) \approx \ln q_k(\boldsymbol{y}|\boldsymbol{X}) = \Psi(\hat{\boldsymbol{f}}) + \ln \int \exp\Big(-\frac{1}{2}(\boldsymbol{f}-\hat{\boldsymbol{f}})^\top(\boldsymbol{K}^{-1}+\boldsymbol{W})(\boldsymbol{f}-\hat{\boldsymbol{f}})\Big)\,\mathrm{d}\boldsymbol{f}, \tag{20}$$

where the integral on the right-hand side can be evaluated by first augmenting it with a constant term C to make it a Gaussian:

$$\begin{aligned}&\int \underbrace{\frac{1}{(2\pi)^{n/2}\sqrt{|(\boldsymbol{K}^{-1}+\boldsymbol{W})^{-1}|}}}_{=C} \exp\Big(-\frac{1}{2}(\boldsymbol{f}-\hat{\boldsymbol{f}})^\top(\boldsymbol{K}^{-1}+\boldsymbol{W})(\boldsymbol{f}-\hat{\boldsymbol{f}})\Big)\,\mathrm{d}\boldsymbol{f}\\ &= \int \mathcal{N}(\boldsymbol{f}|\hat{\boldsymbol{f}},(\boldsymbol{K}^{-1}+\boldsymbol{W})^{-1})\,\mathrm{d}\boldsymbol{f} = 1.\end{aligned}$$

This is feasible since $\boldsymbol{K}^{-1}+\boldsymbol{W}$ remains positive definite. The constant C does not depend on $\boldsymbol{f}$ and can be moved outside of the integral. Hence, the value of the integral is equal to

$$\int \exp\Big(-\frac{1}{2}(\boldsymbol{f}-\hat{\boldsymbol{f}})^\top(\boldsymbol{K}^{-1}+\boldsymbol{W})(\boldsymbol{f}-\hat{\boldsymbol{f}})\Big)\,\mathrm{d}\boldsymbol{f} = C^{-1} = (2\pi)^{n/2}\sqrt{|(\boldsymbol{K}^{-1}+\boldsymbol{W})^{-1}|}.$$

Equation (20) now simplifies with the value of the integral and the definition of $\Psi(\boldsymbol{f})$ from Equation (8) to (compare also Vanhatalo et al (2009))

$$\begin{aligned}\ln p_k(\boldsymbol{y}|\boldsymbol{X}) &\approx \ln q_k(\boldsymbol{y}|\boldsymbol{X})\\ &\overset{(20)}{=} \Psi(\hat{\boldsymbol{f}}) + \ln\left((2\pi)^{n/2}\sqrt{|(\boldsymbol{K}^{-1}+\boldsymbol{W})^{-1}|}\right)\\ &= \Psi(\hat{\boldsymbol{f}}) + \frac{n}{2}\ln 2\pi - \frac{1}{2}\ln|\boldsymbol{K}^{-1}+\boldsymbol{W}|\\ &\overset{(8)}{=} \overbrace{\ln p(\boldsymbol{y}|\hat{\boldsymbol{f}}) - \frac{1}{2}\hat{\boldsymbol{f}}^\top\boldsymbol{K}^{-1}\hat{\boldsymbol{f}} - \frac{1}{2}\ln|\boldsymbol{K}| \underbrace{-\frac{n}{2}\ln 2\pi + \frac{n}{2}\ln 2\pi}_{=0}}^{=\Psi(\hat{\boldsymbol{f}})} - \frac{1}{2}\ln|\boldsymbol{K}^{-1}+\boldsymbol{W}|\\ &= \ln p(\boldsymbol{y}|\hat{\boldsymbol{f}}) - \frac{1}{2}\hat{\boldsymbol{f}}^\top\boldsymbol{K}^{-1}\hat{\boldsymbol{f}} - \frac{1}{2}\ln|\boldsymbol{K}| - \frac{1}{2}\ln|\boldsymbol{K}^{-1}+\boldsymbol{W}|.\end{aligned} \tag{21}$$

In practice, we get the optimal hyperparameters $\boldsymbol{\theta}^\star$ by maximizing the (approximate) log marginal likelihood given in Equation (19), i.e.,

$$\boldsymbol{\theta}^\star = \arg\max_{\boldsymbol{\theta}} \sum_{k=1}^{M} \ln q_k\big(\boldsymbol{y}^{(k)}|\boldsymbol{X}^{(k)},\boldsymbol{\theta}\big),$$

where each approximate log marginal likelihood $\ln q_k\left(\boldsymbol{y}^{(k)}|\boldsymbol{X}^{(k)},\boldsymbol{\theta}\right)$ for the corresponding subset of the data $\mathscr{D}^{(k)} = (\boldsymbol{X}^{(k)},\boldsymbol{y}^{(k)})$ is given by Equation (21). The approximate log marginal likelihood in Equation (21) is differentiable with respect to $\boldsymbol{\theta}$. The maximum can be found with any gradient-based optimizer. Due to the sum rule of the partial derivative, the computation of the gradient by its partial derivatives

$$\begin{aligned}\frac{\partial}{\partial\theta_j}\ln p(\boldsymbol{y}|\boldsymbol{X},\boldsymbol{\theta}) &\approx \frac{\partial}{\partial\theta_j}\prod_{k=1}^{M}\ln p_k(\boldsymbol{y}^{(k)}|\boldsymbol{X}^{(k)},\boldsymbol{\theta}) = \frac{\partial}{\partial\theta_j}\sum_{k=1}^{M}\ln p_k\left(\boldsymbol{y}^{(k)}|\boldsymbol{X}^{(k)},\boldsymbol{\theta}\right)\\ &= \sum_{k=1}^{M}\frac{\partial}{\partial\theta_j}\ln p_k\left(\boldsymbol{y}^{(k)}|\boldsymbol{X}^{(k)},\boldsymbol{\theta}\right) \approx \sum_{k=1}^{M}\frac{\partial}{\partial\theta_j}\ln q_k\left(\boldsymbol{y}^{(k)}|\boldsymbol{X}^{(k)},\boldsymbol{\theta}\right)\end{aligned}$$

can be also distributed and therefore optimization is straightforward. The partial derivatives of the approximate log marginal likelihood in Equation (21) are provided in the appendix.

4.2 Prediction

After training, we end up with M robust GPs, each of which has been trained on its subset of the data $\mathscr{D}^{(k)}$ for $k = 1,2,\ldots,M$. Whereas the training procedure is the same among the Product-of-Experts family, the combination of the individual predictions is not. We briefly introduce the idea of the original Product-of-Experts model (Cao and Fleet 2014) before we detail a robust Bayesian Committee Machine (Deisenroth and Ng 2015) approach for GPs with Student-t observation models.

Consider the prediction $f_* = f(\boldsymbol{x}_*)$ of a test input $\boldsymbol{x}_* \in \mathbb{R}^d$ as well as its variance prediction which is computed by the posterior predictive distribution which factorizes due to the independence assumption into

$$p(f_*|\mathscr{D},\boldsymbol{x}_*) = \prod_{k=1}^{M} p_k(f_*|\mathscr{D}^{(k)},\boldsymbol{x}_*).$$

In a vanilla GP setting, all experts $p_k(f_*|\mathscr{D}^{(k)},\boldsymbol{x}_*)$ follow a Gaussian distribution. When predicting we end up having M independent mean predictions $\mu_k(\boldsymbol{x}_*)$ as well as M variance predictions $\sigma_k^2(\boldsymbol{x}_*)$. Since the product of Gaussians is proportional to a Gaussian the individual predictions can be combined to a single prediction by

$$\begin{aligned}\mu_*^{\mathrm{PoE}}(\boldsymbol{x}_*) &= \left(\sigma_*^{\mathrm{PoE}}\right)^2\sum_{k=1}^{M}\frac{\mu_k(\boldsymbol{x}_*)}{\sigma_k^2(\boldsymbol{x}_*)},\\ \left(\sigma_*^{\mathrm{PoE}}\right)^{-2}(\boldsymbol{x}_*) &= \sum_{k=1}^{M}\sigma_k^{-2}(\boldsymbol{x}_*).\end{aligned}$$

Cao and Fleet (2014) propose the generalized-Product-of-Experts (gPoE) model, where a weight β_k is assigned to each expert. By contrast, we propose an approach that is based on robust Bayesian Committee Machines (rBCM) (Deisenroth and Ng 2015). The predictive distribution is thus given by

$$p(f_*|\mathscr{D},\boldsymbol{x}_*) = \frac{\prod_{k=1}^{M} p_k^{\beta_k}(f_*|\mathscr{D}^{(k)},\boldsymbol{x}_*)}{p^{-1+\sum_k\beta_k}(f_*|\boldsymbol{x}_*)},$$

and it includes Cao and Fleet (2014) as well as Tresp (2000) as special cases. The corresponding mean and precision estimates are given by

$$\mu_*^{\text{rBCM}}(\boldsymbol{x}_*) = \left(\sigma_*^{\text{rBCM}}\right)^2 \sum_{k=1}^{M} \beta_k \cdot \frac{\mu_k(\boldsymbol{x}_*)}{\sigma_k^2(\boldsymbol{x}_*)},$$
$$\left(\sigma_*^{\text{rBCM}}\right)^{-2}(\boldsymbol{x}_*) = \sum_{k=1}^{M} \beta_k \cdot \frac{1}{\sigma_k^2(\boldsymbol{x}_*)} + \left(1 - \sum_{k=1}^{M} \beta_k\right) \cdot \sigma_{**}^{-2},$$

respectively. Here, the weights β_k are set as suggested by Cao and Fleet (2014) to be the differential entropy between the prior $p(f_*|\boldsymbol{x}_*)$ and the posterior $p_k(f_*|\mathscr{D}^{(k)}, \boldsymbol{x}_*)$, which can be computed as

$$\beta_k = \frac{1}{2}\Big(\log \sigma_{**}^2 - \log \sigma_k^2(\boldsymbol{x}_*)\Big),$$

to determine the importance of expert k. Like in the Bayesian Committee Machine, σ_{**}^2 is meant to be the prior variance. The Laplace Approximation which was used for approximative inference in the previous section yields a Gaussian predictive distribution. This constitutes a perfect match as the rBCM requires the experts having Gaussian predictive distributions. Hence, prediction is straightforward. In the following two sections we derive the mean $\mu_k(\boldsymbol{x}_*)$ and variance $\sigma_k^2(\boldsymbol{x}_*)$ for the predictors of expert k. Once again, we focus on a single expert and therefore drop the superscript (k) but keep the subscript k hereafter.

4.2.1 Mean Prediction of an Expert

Since the approximate posterior distribution is Gaussian, mean and variance predictions possess solutions in closed-form. For the mean, we obtain the following result with the help of the auxiliary variable $\boldsymbol{a}$,

$$\mathbb{E}_{q(\boldsymbol{f}|\boldsymbol{X},\boldsymbol{y})}\Big[f_*|\boldsymbol{X},\boldsymbol{y},\boldsymbol{x}_*\Big] \;=\; \boldsymbol{k}_*^\top \underbrace{\boldsymbol{K}^{-1}\hat{\boldsymbol{f}}}_{=\boldsymbol{a}} \overset{(14)}{=} \boldsymbol{k}_*^\top \boldsymbol{a}. \tag{22}$$

4.2.2 Variance Prediction of an Expert

Apart from matrix $\boldsymbol{W}$, the variance prediction of the robust GP is similar to that of regular Gaussian Process regression in Equation (5); we have,

$$\begin{aligned}
\mathbb{V}_{q(\boldsymbol{f}|\boldsymbol{X},\boldsymbol{y})}\Big[f_*|\boldsymbol{X},\boldsymbol{y},\boldsymbol{x}_*\Big] &= \underbrace{\mathbb{E}_{p(f_*|\boldsymbol{X},\boldsymbol{f},\boldsymbol{x}_*)}\Big[(f_* - \mathbb{E}[f_*|\boldsymbol{X},\boldsymbol{f},\boldsymbol{x}_*])^2\Big]}_{=k_{**}-\boldsymbol{k}_*^\top\boldsymbol{K}^{-1}\boldsymbol{k}_*} \qquad (23)\\
&\quad + \mathbb{E}_{q(\boldsymbol{f}|\boldsymbol{X},\boldsymbol{y})}\Big[(\mathbb{E}[f_*|\boldsymbol{X},\boldsymbol{f},\boldsymbol{x}_*] - \mathbb{E}[f_*|\boldsymbol{X},\boldsymbol{y},\boldsymbol{x}_*])^2\Big]\\
&= k_{**} - \boldsymbol{k}_*^\top\boldsymbol{K}^{-1}\boldsymbol{k}_* + \mathbb{E}_{q(\boldsymbol{f}|\boldsymbol{X},\boldsymbol{y})}\Big[(\mathbb{E}[f_*|\boldsymbol{X},\boldsymbol{f},\boldsymbol{x}_*] - \mathbb{E}[f_*|\boldsymbol{X},\boldsymbol{y},\boldsymbol{x}_*])^2\Big]\\
&= k_{**} - \boldsymbol{k}_*^\top\boldsymbol{K}^{-1}\boldsymbol{k}_* + \boldsymbol{k}_*^\top\boldsymbol{K}^{-1}(\boldsymbol{K}^{-1}+\boldsymbol{W})^{-1}\boldsymbol{K}^{-1}\boldsymbol{k}_*\\
&= k_{**} - \boldsymbol{k}_*^\top(\boldsymbol{K}+\boldsymbol{W}^{-1})^{-1}\boldsymbol{k}_*. \qquad (24)
\end{aligned}$$

Like in Equation (12), inverting the kernel matrix $\boldsymbol{K}$ to compute the predictive variance is prohibitive. By setting $\boldsymbol{B} = \boldsymbol{W}^{\frac{1}{2}}\boldsymbol{K}\boldsymbol{W}^{\frac{1}{2}} + \boldsymbol{I}$, we solve the inverse of $\boldsymbol{K} + \boldsymbol{W}^{-1}$ by inverting the matrix $\boldsymbol{B}$,

$$\begin{aligned}(\boldsymbol{K}+\boldsymbol{W}^{-1})^{-1} &= \underbrace{\boldsymbol{W}^{\frac{1}{2}}\boldsymbol{W}^{-\frac{1}{2}}}_{=\boldsymbol{I}}(\boldsymbol{K}+\boldsymbol{W}^{-1})^{-1}\underbrace{\boldsymbol{W}^{-\frac{1}{2}}\boldsymbol{W}^{\frac{1}{2}}}_{=\boldsymbol{I}} = \boldsymbol{W}^{\frac{1}{2}}(\boldsymbol{W}^{\frac{1}{2}}\boldsymbol{K}\boldsymbol{W}^{\frac{1}{2}} + \underbrace{\boldsymbol{W}^{\frac{1}{2}}\boldsymbol{W}^{-1}\boldsymbol{W}^{\frac{1}{2}}}_{=\boldsymbol{I}})^{-1}\boldsymbol{W}^{\frac{1}{2}} \\ &= \boldsymbol{W}^{\frac{1}{2}}(\underbrace{\boldsymbol{W}^{\frac{1}{2}}\boldsymbol{K}\boldsymbol{W}^{\frac{1}{2}} + \boldsymbol{I}}_{=\boldsymbol{B}})^{-1}\boldsymbol{W}^{\frac{1}{2}} = \boldsymbol{W}^{\frac{1}{2}}\boldsymbol{B}^{-1}\boldsymbol{W}^{\frac{1}{2}}. \end{aligned} \tag{25}$$

However, the computation of $\boldsymbol{W}^{\frac{1}{2}}$ might be problematic, because the diagonal matrix $\boldsymbol{W}$ may contain negative entries corresponding to negative variances. This is due to the non-log-concave likelihood caused by the choice of a Student-t distribution. At the positions where the negative values occur, it is unclear, whether the observation is an outlier or not and therefore the posterior is less certain than the prior and likelihood. A common way to solve this issue is to simply set all negative entries to a very small positive number, i.e., $W_{ii} = 10^{-6}$ if $W_{ii} < 0$. Using a Cholesky decomposition $\boldsymbol{L}\boldsymbol{L}^\top = \boldsymbol{B}$, the predictive variance can now be rewritten as

$$\begin{aligned}\mathbb{V}_{q(\boldsymbol{f}|\boldsymbol{X},\boldsymbol{y})}\left[f_*|\boldsymbol{X},\boldsymbol{y},\boldsymbol{x}_*\right] &\overset{(24)}{=} k_{**} - \boldsymbol{k}_*^\top(\boldsymbol{K}+\boldsymbol{W}^{-1})^{-1}\boldsymbol{k}_* \overset{(25)}{=} k_{**} - \boldsymbol{k}_*^\top\boldsymbol{W}^{\frac{1}{2}}\boldsymbol{B}^{-1}\boldsymbol{W}^{\frac{1}{2}}\boldsymbol{k}_* \\ &= k_{**} - \boldsymbol{k}_*^\top\boldsymbol{W}^{\frac{1}{2}}(\boldsymbol{L}^{-1})^\top\underbrace{\boldsymbol{L}^{-1}\boldsymbol{W}^{\frac{1}{2}}\boldsymbol{k}_*}_{:=\boldsymbol{w}} = k_{**} - \boldsymbol{w}^\top\boldsymbol{w},\end{aligned}$$

where $\boldsymbol{w} = \boldsymbol{L}^{-1}\boldsymbol{W}^{\frac{1}{2}}\boldsymbol{k}_*$.

5 Empirical Results

In this section, we empirically evaluate our distributed robust GPs (*DRGP*) on several real and artificial data sets. We compare the performance to vanilla Gaussian Processes (*GP*) (Rasmussen and Williams 2006), robust Gaussian Processes using Student-t observation models (*RGP*) (Vanhatalo et al 2009), and distributed Gaussian Processes (*DGP*) (Deisenroth and Ng 2015), respectively.

5.1 Runtime

We begin by studying the runtime of the different approaches by measuring the time needed to compute the (approximative) negative log marginal likelihood and its gradients for a one-dimensional toy problem with more than 10 million instances. The generated data is sampled from a random linear function with a Gaussian distributed noise term, i.e., $\varepsilon_i \sim \mathcal{N}(0, 0.35)$. For the robust versions, we fix the number of iterations of the Expectation Maximization algorithm to 20. All experiments run on a cluster with 60 cores. Figure 2 shows the resulting runtimes for varying data set sizes in a log-log scale.

Gaussian Processes (GP) exhibit a linear trend which corresponds to an exponential growth because of the double log-scale. Their robust counterpart (RGP) performs similarly and requires slightly more time due to the additional Expectation Maximization algorithm. The distributed GPs are tested with several configurations by instantiating local experts of size 512, 256, and 128.

For the distributed GPs, the required time is nearly constant up to a certain break point and then continues linearly. The resulting exponential continuations, however, possess smaller growth rates than the stand-alone peers. The break point indicates that the size of the data exceeds the number of instances that can be processed in parallel; that is, using

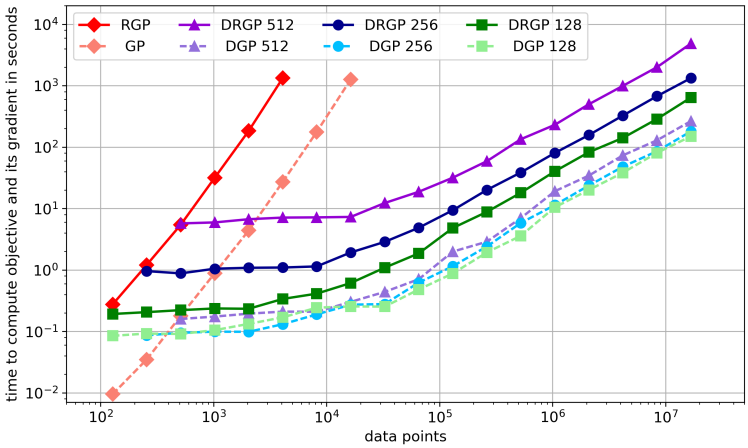

Fig. 2 Time to compute the (approximate) log marginal likelihood and its gradient. Axes in log-log scale.

60 cores and experts of size 512 allows to process $60 \cdot 512 = 30,720$ data points in a single run. Larger quantities render multiple repetitions necessary and the computation time grows proportionally. Similar to the stand-alone peers, the robust distributed GPs need slightly more time than their peers with Gaussian observation models.

5.2 Aimpeak

The Aimpeak data (Chen et al 2013) consists of 41,850 measurements along 775 segments of an urban road network. The task is to predict the traffic speed in kilometers per hour. We use the first $n = 36,000$ observations as training data and the remaining $n_* = 5,850$ observations as test set. To visualize the impact of the size n_k of the M experts, we vary the expert size from $n_k = 75$ ($M = 480$ experts) to $n_k = 600$ ($M = 60$ experts). We measure performance in terms of root mean square error (RMSE), mean average error (MAE), and mean negative log probability (MNLP). We use these three measures as RMSE assumes errors to be Gaussian and hence puts much weight to outliers that may even dominate the RMSE (Willmott and Matsuura 2005; Chai and Draxler 2014). The MAE treats all errors equally which appears more appropriate for studying robust regression. Finally, MNLP takes the predictive variance into account and sheds light on the confidence of the regressors. Distributed algorithms use random partitions of the training data. We report on averages over 25 runs, error bars show standard error.

Figure 3 shows the results. As expected, the MAE and RMSE errors decrease for both approaches with an increasing expert size n_k. We credit this finding to a better approximation of the global kernel matrix by larger values of n_k as shown in Figure 4. The proposed method yields better MAE and RMSE errors than vanilla distributed Gaussian Process regression especially for small expert sizes n_k while the DGP catches up and achieves slightly better RMSEs for large expert sizes $n_k \gtrsim 420$. Furthermore, the DGP results in better MNLP-values for all configurations which may indicate that its predictive variance is larger than that of the DRGPs.

5.3 Boston Housing

We include the Boston Housing data (Harrison and Rubinfeld 1978) in our experiments as it has become a benchmark for testing robustness. For instance, previous findings on Boston Housing consistently show that a Robust GP with

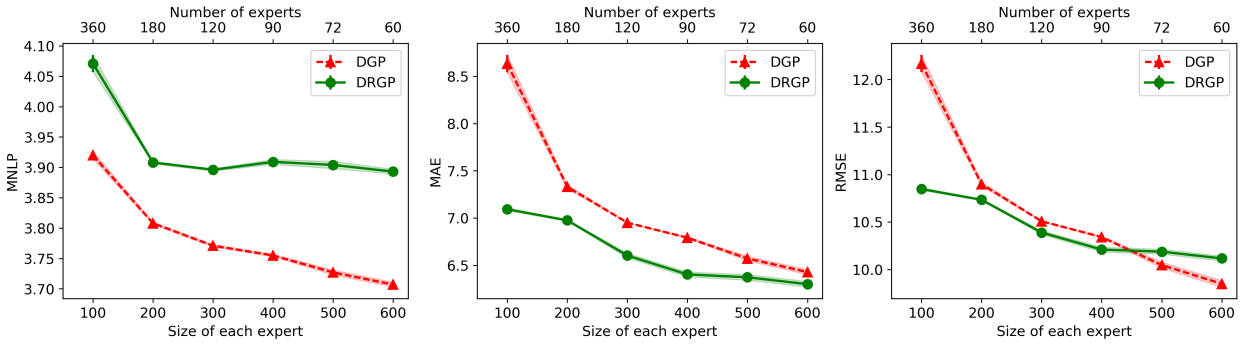

Fig. 3 Results for the Aimpeak data.

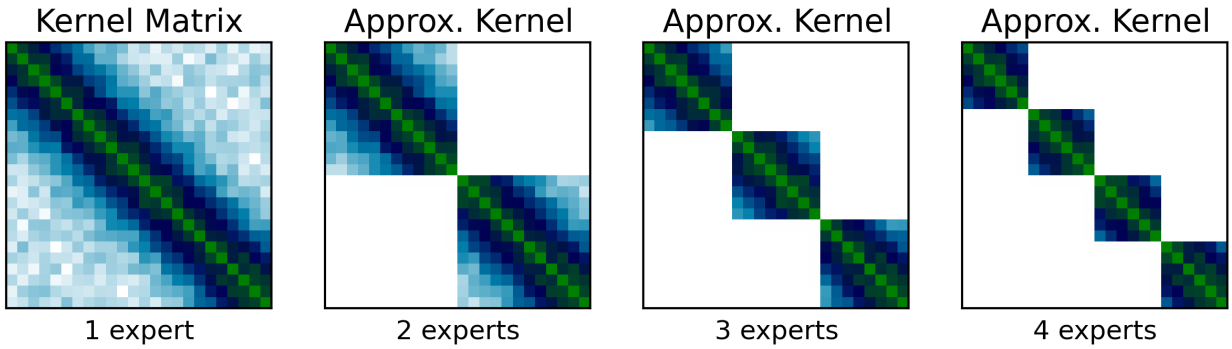

Fig. 4 The effect of smaller expert sizes. Fewer experts process more data and the approximation of the kernel matrix is closer to the global kernel matrix.

a Student-t likelihood model performs better than a vanilla Gaussian Process regression (Kuss 2006; Vanhatalo et al 2009; Jylänki et al 2011).

The data consists of 506 data points in 13 dimensions and the task is to predict the median value of owner-occupied homes. Due to the size of the data set we are able to compare against full models. After normalization, the first $n = 455$ observations form the training and the remaining $n_* = 51$ observations the test set. We apply random partitions of the training data for distributed settings. We compare the distributed GPs (vanilla and robust) form a single expert, i.e., $M = 1$, ($n_k = 455$) up to $M = 6$ experts ($n_k \approx 75$). Note that we obtain the stand-alone variants on the full data set as special-cases for maintaining only a single expert ($M = 1$). We report again on averages of 25 repetitions; error bars indicate standard errors.

The results are shown in Figure 5. The (distributed) robust GPs clearly outperform their peers with Gaussian observation models in all settings. In addition, the standard errors of the proposed method are consistently much smaller, indicating robust and reliable predictive distributions. Once again, the DRGP prove superior to DGP particularly for small expert sizes n_k and many experts M.

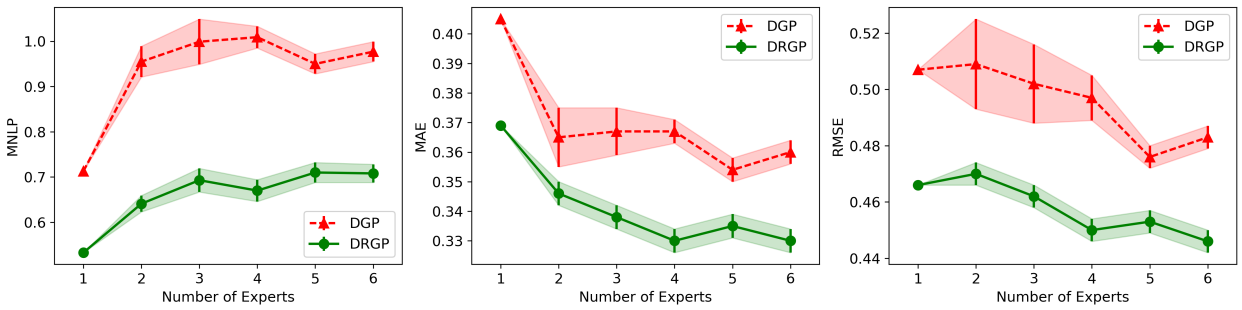

Fig. 5 Results for the Boston Housing data.

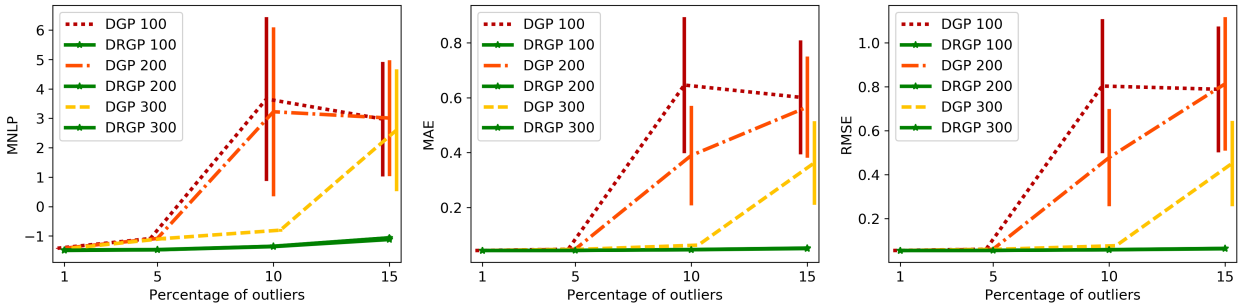

Fig. 6 Results for the artificial data.

5.4 Large-scale Artificial Data

We now compare the approaches on larger scales using artificial data to control the amount of outliers. To do so, we use a simple linear model. The data is generated as follows. We sample a design matrix of size $202,000 \times 5$ from a standard normal distribution and a five dimensional parameter vector from a uniform distribution such that only three of those five dimensions are informative. Gaussian noise ε_i is added to the target values using a mean of zero and a standard deviation of $\sigma_\varepsilon = 2$. Therefore, the remaining two dimensions only carry white noise. We sample $n = 200,000$ training and $n_* = 2,000$ test instances where instances are randomly turned into outliers by adding ten standard deviations such that a predefined ratio of outlier/all points is obtained. The predefined ratio ranges from 1 to 15 percent. We focus on distributed GPs and our robust variants thereof and vary the size of experts from $n_k = 100$ ($M = 2,000$ experts) to $n_k = 300$ ($M = 667$ experts) and report on averages including standard errors over 10 runs using different random partitions of the data.

Figure 6 shows the respective performances for different amounts of outliers. Unsurprisingly, a few outliers do not harm performance and vanilla and robust distributed GPs perform similarly. However, the more outliers are contained in the data, the more difficult the learning problem and performances consequentially deteriorate for the vanilla DGPs. Consequentially, the take-home message is twofold: Firstly, distributed GPs with Gaussian observation models deteriorate at significantly faster rates than our Student-t-based approach. Secondly, the MAE which may be seen as the most appropriate performance measure in this scenario as it weights all instances equally, irrespectively

of whether it is an outlier or not, remains constant for our approach. The latter shows that our distributed robust GPs constitute effective means in noisy settings with many outliers.

6 Conclusion

We presented distributed and robust Gaussian Process regression. We argued that the Gaussian observation model in vanilla GPs is unable to reject extreme observations that deteriorate predictive performance. As a remedy, we incorporated a Student-*t* observation model that was optimized using a Laplace Approximation for computation time and predictive power. As a side effect, the Gaussian posterior is preserved and the resulting predictive distribution could straight forwardly be distributed using ideas from robust Bayesian Committee Machines. Our approach thus allowed for distributing data and algorithm and rendered robust regression feasible at large-scales. Empirically, our approach performed orders of magnitude faster than stand-alone competitors and needs only slightly more time than distributed counterpart with a Gaussian observation model. It also proved significantly better in terms of predictive performance than vanilla (distributed) Gaussian Processes, especially for small expert sizes.

References

Cao Y, Fleet DJ (2014) Generalized Product of Experts for Automatic and Principled Fusion of Gaussian Process Predictions. In: Modern Nonparametrics 3: Automating the Learning Pipeline workshop at NIPS

Chai T, Draxler RR (2014) Root mean square error (RMSE) or mean absolute error (MAE)?–Arguments against avoiding RMSE in the literature. Geoscientific Model Development 7(3):1247–1250

Chen J, Cao N, Low KH, Ouyang R, Tan CKY, Jaillet P (2013) Parallel gaussian process regression with low-rank covariance matrix approximations. In: Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence, AUAI Press, pp 152–161

Deisenroth MP, Ng JW (2015) Distributed Gaussian Processes. In: Proceedings of the 32nd International Conference on Machine Learning (ICML-15), pp 1481–1490

Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. Journal of the royal statistical society Series B (methodological) pp 1–38

Gal Y, van der Wilk M, Rasmussen C (2014) Distributed variational inference in sparse Gaussian process regression and latent variable models. In: Advances in Neural Information Processing Systems, pp 3257–3265

Gelman A, Carlin JB, Stern HS, Dunson DB, Vehtari A, Rubin DB (2013) Bayesian Data Analysis. CRC Press

Geweke J (1993) Bayesian treatment of the independent student-t linear model. Journal of Applied Econometrics 8(S1):S19–S40

Harrison D, Rubinfeld DL (1978) Hedonic housing prices and the demand for clean air. Journal of environmental economics and management 5(1):81–102

Hensman J, Fusi N, Lawrence ND (2013) Gaussian processes for big data. In: Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence, AUAI Press, pp 282–290

Jaynes E, Bretthorst G (2003) Probability theory: the logic of science. Cambridge university press

Jylänki P, Vanhatalo J, Vehtari A (2011) Robust Gaussian process regression with a Student-t likelihood. The Journal of Machine Learning Research 12:3227–3257

Kuss M (2006) Gaussian Process Models for Robust Regression, Classification, and Reinforcement Learning. PhD thesis, Technische Universität Darmstadt

Naish-Guzman A, Holden S (2008) Robust regression with twinned Gaussian processes. In: Advances in Neural Information Processing Systems, pp 1065–1072

Neal R (1997) Monte Carlo Implementation of Gaussian Process Models for Bayesian Regression and Classification. Technical report (University of Toronto .Dept. of Statistics), University of Toronto

Nickisch H, Rasmussen CE (2008) Approximations for binary Gaussian process classification. Journal of Machine Learning Research 9(10)

O'Hagan A (1979) On outlier rejection phenomena in Bayes inference. Journal of the Royal Statistical Society Series B (Methodological) pp 358–367

Quiñonero-Candela J, Rasmussen CE (2005) A unifying view of sparse approximate Gaussian process regression. The Journal of Machine Learning Research 6:1939–1959

Rasmussen C, Williams C (2006) Gaussian Processes for Machine Learning. Adaptive Computation and Machine Learning, MIT Press, Cambridge, MA, USA, URL `http://mitpress.mit.edu/026218253X`

Tipping ME, Lawrence ND (2005) Variational inference for Student-t models: Robust Bayesian interpolation and generalised component analysis. Neurocomputing 69(1):123–141

Titsias MK (2009) Variational learning of inducing variables in sparse Gaussian processes. In: International Conference on Artificial Intelligence and Statistics, pp 567–574

Tresp V (2000) A Bayesian committee machine. Neural Computation 12(11):2719–2741

Vanhatalo J, Jylänki P, Vehtari A (2009) Gaussian process regression with Student-t likelihood. In: Advances in Neural Information Processing Systems, pp 1910–1918

Willmott CJ, Matsuura K (2005) Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. Climate research 30(1):79

Appendix

Before providing the derivation of the partial derivatives of the approximate log marginal likelihood in Equation (21), we introduce the matrix $\boldsymbol{R}$, which will be convenient later on.

$$\boldsymbol{R} = (\boldsymbol{W}^{-1} + \boldsymbol{K})^{-1} \overset{(25)}{=} \boldsymbol{W}^{\frac{1}{2}} (\underbrace{\boldsymbol{I} + \boldsymbol{W}^{\frac{1}{2}} \boldsymbol{K} \boldsymbol{W}^{\frac{1}{2}}}_{=\boldsymbol{B}=\boldsymbol{L}\boldsymbol{L}^\top})^{-1} \boldsymbol{W}^{\frac{1}{2}} = \boldsymbol{W}^{\frac{1}{2}} \boldsymbol{B}^{-1} \boldsymbol{W}^{\frac{1}{2}}. \tag{26}$$

Using the matrix $\boldsymbol{R}$ as well as the matrix inversion lemma allows to reformulate the inverse of $\boldsymbol{K}^{-1} + \boldsymbol{W}$ as a sum of the kernel matrix $\boldsymbol{K}$ and a new matrix $\boldsymbol{J}$,

$$\begin{aligned}\left(\boldsymbol{K}^{-1} + \boldsymbol{W}\right)^{-1} &= \boldsymbol{K} - \boldsymbol{K} \underbrace{\left(\boldsymbol{K} + \boldsymbol{W}^{-1}\right)^{-1}}_{=\boldsymbol{R}} \boldsymbol{K} \overset{(26)}{=} \boldsymbol{K} - \boldsymbol{K}\boldsymbol{R}\boldsymbol{K} \overset{(26)}{=} \boldsymbol{K} - \boldsymbol{K}\boldsymbol{W}^{\frac{1}{2}} \boldsymbol{B}^{-1} \boldsymbol{W}^{\frac{1}{2}} \boldsymbol{K} \\ &= \boldsymbol{K} - \underbrace{\boldsymbol{K}\boldsymbol{W}^{\frac{1}{2}} (\boldsymbol{L}^{-1})^\top}_{=\boldsymbol{J}^\top} \underbrace{\boldsymbol{L}^{-1} \boldsymbol{W}^{\frac{1}{2}} \boldsymbol{K}}_{:=\boldsymbol{J}} = \boldsymbol{K} - \boldsymbol{J}^\top \boldsymbol{J}.\end{aligned} \tag{27}$$

Recall that there are kernel as well as the likelihood hyperparameters. We focus on a squared exponential kernel with automatic relevance detection parametrized by the signal noise σ_f and the length scales ℓ_i for all $i = 1, 2, \ldots, d$ dimensions. The likelihood is parametrized by the scale σ_t and the degree of freedom ν.

Partial Derivatives with respect to the kernel hyperparameters

The partial derivatives with respect to the kernel hyperparameters are given by

$$\begin{aligned}\frac{\partial \ln q(\boldsymbol{y}|\boldsymbol{X})}{\partial \theta_j} &\overset{(21)}{=} \frac{\partial}{\partial \theta_j} \left(\ln p(\boldsymbol{y}|\hat{\boldsymbol{f}}) - \frac{1}{2} \hat{\boldsymbol{f}}^\top \boldsymbol{K}^{-1} \hat{\boldsymbol{f}} - \frac{1}{2} \ln |\boldsymbol{B}| \right) \\ &= \underbrace{\frac{\partial \ln q(\boldsymbol{y}|\boldsymbol{X})}{\partial \theta_j}}_{\text{explicit}} + \underbrace{\sum_{i=1}^{n} \frac{\partial \ln q(\boldsymbol{y}|\boldsymbol{X})}{\partial \hat{f}_i} \frac{\partial \hat{f}_i}{\partial \theta_j}}_{\text{implicit}},\end{aligned} \tag{28}$$

which consists of an explicit and an implicit term. The implicit term is caused by the dependence of $\hat{\boldsymbol{f}}$ and $\boldsymbol{W}$ on $\boldsymbol{K}$ and therefore depends on the hyperparameters. The first part of the explicit term

$$\frac{\partial}{\partial \theta_j} \ln p(\boldsymbol{y}|\hat{\boldsymbol{f}}) = 0$$

is equal to zero. For the second term we use the intermediate result $\boldsymbol{a}$ from Equation (14) to obtain

$$\frac{\partial}{\partial\theta_j}\left(-\frac{1}{2}\hat{\boldsymbol{f}}^\top\boldsymbol{K}^{-1}\hat{\boldsymbol{f}}\right)=\frac{1}{2}\underbrace{\hat{\boldsymbol{f}}^\top\boldsymbol{K}^{-1}}_{=\boldsymbol{a}^\top}\frac{\partial\boldsymbol{K}}{\partial\theta_j}\underbrace{\boldsymbol{K}^{-1}\hat{\boldsymbol{f}}}_{=\boldsymbol{a}}\overset{(14)}{=}\frac{1}{2}\boldsymbol{a}^\top\frac{\partial\boldsymbol{K}}{\partial\theta_j}\boldsymbol{a},$$

and for the third term we get

$$\begin{aligned}\frac{\partial}{\partial\theta_j}\left(-\frac{1}{2}\ln|\boldsymbol{B}|\right)&=-\frac{1}{2}\operatorname{tr}\left(\boldsymbol{B}^{-1}\frac{\boldsymbol{B}}{\partial\theta_j}\right)=-\frac{1}{2}\operatorname{tr}\left(\boldsymbol{B}^{-1}\frac{\partial}{\partial\theta_j}\left(\boldsymbol{I}+\boldsymbol{W}^{\frac{1}{2}}\boldsymbol{K}\boldsymbol{W}^{\frac{1}{2}}\right)\right)\\&=-\frac{1}{2}\operatorname{tr}\left(\boldsymbol{B}^{-1}\boldsymbol{W}^{\frac{1}{2}}\frac{\partial\boldsymbol{K}}{\partial\theta_j}\boldsymbol{W}^{\frac{1}{2}}\right)=-\frac{1}{2}\operatorname{tr}\left(\boldsymbol{W}^{\frac{1}{2}}\boldsymbol{B}^{-1}\boldsymbol{W}^{\frac{1}{2}}\frac{\partial\boldsymbol{K}}{\partial\theta_j}\right)\\&\overset{(26)}{=}-\frac{1}{2}\operatorname{tr}\left((\boldsymbol{W}^{-1}+\boldsymbol{K})^{-1}\frac{\partial\boldsymbol{K}}{\partial\theta_j}\right)\overset{(26)}{=}-\frac{1}{2}\operatorname{tr}\left(\boldsymbol{R}\frac{\partial\boldsymbol{K}}{\partial\theta_j}\right)\end{aligned}$$

by using the definitions of the matrices $\boldsymbol{B}$ and $\boldsymbol{R}$ and the fact that circular rotation of matrix products does not change the trace of the product. Therefore, the explicit part of the partial derivative is given by

$$\left.\frac{\partial\ln q(\boldsymbol{y}|\boldsymbol{X})}{\partial\theta_j}\right|_{\text{explicit}}=\frac{1}{2}\boldsymbol{a}^\top\frac{\partial\boldsymbol{K}}{\partial\theta_j}\boldsymbol{a}-\frac{1}{2}\operatorname{tr}\left(\boldsymbol{R}\frac{\partial\boldsymbol{K}}{\partial\theta_j}\right).$$

Now we take care of the implicit part of the partial derivative. The derivation of the first two parts is equivalent to the derivation of $\Psi(\hat{\boldsymbol{f}})$, which is for $\hat{\boldsymbol{f}}$ equal to zero,

$$\frac{\partial}{\partial\hat{\boldsymbol{f}}}\left(\ln p(\boldsymbol{y}|\hat{\boldsymbol{f}})-\frac{1}{2}\hat{\boldsymbol{f}}^\top\boldsymbol{K}^{-1}\hat{\boldsymbol{f}}\right)\equiv\frac{\partial}{\partial\hat{\boldsymbol{f}}}\Psi(\hat{\boldsymbol{f}})=0.$$

The third term of the partial derivative is the derivation of the log determinant of $\boldsymbol{B}$. Using the definition of the matrices $\boldsymbol{B}$ and $\boldsymbol{J}$ yields

$$\begin{aligned}\frac{\partial}{\partial\hat{f}_i}\left(-\frac{1}{2}\ln|\boldsymbol{B}|\right)&=-\frac{1}{2}\operatorname{tr}\left(\boldsymbol{B}^{-1}\frac{\boldsymbol{B}}{\partial\hat{f}_i}\right)=-\frac{1}{2}\operatorname{tr}\left(\left(\boldsymbol{I}+\boldsymbol{K}\boldsymbol{W}\right)^{-1}\frac{\partial}{\partial\hat{f}_i}\left(\boldsymbol{I}+\boldsymbol{K}\boldsymbol{W}\right)\right)\\&=-\frac{1}{2}\operatorname{tr}\left(\left(\boldsymbol{I}+\boldsymbol{K}\boldsymbol{W}\right)^{-1}\boldsymbol{K}\frac{\partial\boldsymbol{W}}{\partial\hat{f}_i}\right)=-\frac{1}{2}\operatorname{tr}\left(\left(\boldsymbol{K}(\boldsymbol{K}^{-1}+\boldsymbol{W})\right)^{-1}\boldsymbol{K}\frac{\partial\boldsymbol{W}}{\partial\hat{f}_i}\right)\\&=-\frac{1}{2}\operatorname{tr}\left(\left(\boldsymbol{K}^{-1}+\boldsymbol{W}\right)^{-1}\underbrace{\boldsymbol{K}^{-1}\boldsymbol{K}}_{=\boldsymbol{I}}\frac{\partial\boldsymbol{W}}{\partial\hat{f}_i}\right)=-\frac{1}{2}\operatorname{tr}\left(\left(\boldsymbol{K}^{-1}+\boldsymbol{W}\right)^{-1}\frac{\partial\boldsymbol{W}}{\partial\hat{f}_i}\right)\\&\overset{(27)}{=}-\frac{1}{2}\operatorname{tr}\left(\left(\boldsymbol{K}-\boldsymbol{J}^\top\boldsymbol{J}\right)\frac{\partial\boldsymbol{W}}{\partial\hat{f}_i}\right)=-\frac{1}{2}\operatorname{diag}\left(\operatorname{diag}(\boldsymbol{K})-\operatorname{diag}(\boldsymbol{J}^\top\boldsymbol{J})\right)\cdot\frac{\partial\boldsymbol{W}}{\partial\hat{f}_i}.\end{aligned}$$

We still need to take care of $\frac{\partial \hat{\boldsymbol{f}}}{\partial \theta_j}$. By using the definition of $\hat{\boldsymbol{f}}$ from Equation (11) as well as the multidimensional chain rule we obtain

$$\begin{aligned}\frac{\partial \hat{\boldsymbol{f}}}{\partial \theta_j} &= \frac{\partial \boldsymbol{K} \nabla \ln p(\boldsymbol{y}|\hat{\boldsymbol{f}})}{\partial \theta_j} + \frac{\partial \boldsymbol{K} \nabla \ln p(\boldsymbol{y}|\hat{\boldsymbol{f}})}{\partial \hat{\boldsymbol{f}}} \frac{\hat{\boldsymbol{f}}}{\partial \theta_j} \\ &= \frac{\partial \boldsymbol{K}}{\partial \theta_j} \nabla \ln p(\boldsymbol{y}|\hat{\boldsymbol{f}}) + \boldsymbol{K} \underbrace{\frac{\partial \nabla \ln p(\boldsymbol{y}|\hat{\boldsymbol{f}})}{\partial \hat{\boldsymbol{f}}}}_{=-\boldsymbol{W}} \frac{\partial \hat{\boldsymbol{f}}}{\partial \theta_j} \\ &= \frac{\partial \boldsymbol{K}}{\partial \theta_j} \nabla \ln p(\boldsymbol{y}|\hat{\boldsymbol{f}}) - \boldsymbol{K}\boldsymbol{W} \frac{\partial \hat{\boldsymbol{f}}}{\partial \theta_j}\end{aligned}$$

which is equivalent to

$$\begin{aligned}&\frac{\partial \hat{\boldsymbol{f}}}{\partial \theta_j} + \boldsymbol{K}\boldsymbol{W} \frac{\partial \hat{\boldsymbol{f}}}{\partial \theta_j} = (\boldsymbol{I} + \boldsymbol{K}\boldsymbol{W}) \frac{\partial \hat{\boldsymbol{f}}}{\partial \theta_j} = \frac{\partial \boldsymbol{K}}{\partial \theta_j} \nabla \ln p(\boldsymbol{y}|\hat{\boldsymbol{f}}) \\ \iff \quad &\frac{\partial \hat{\boldsymbol{f}}}{\partial \theta_j} = (\boldsymbol{I} + \boldsymbol{K}\boldsymbol{W})^{-1} \frac{\partial \boldsymbol{K}}{\partial \theta_j} \nabla \ln p(\boldsymbol{y}|\hat{\boldsymbol{f}}) \\ &= \Big(\boldsymbol{I} - \boldsymbol{K} \underbrace{(\boldsymbol{W}^{-1} + \boldsymbol{K})^{-1}}_{=\boldsymbol{R}}\Big) \frac{\partial \boldsymbol{K}}{\partial \theta_j} \nabla \ln p(\boldsymbol{y}|\hat{\boldsymbol{f}}) \\ &= \Big(\boldsymbol{I} - \boldsymbol{K}\boldsymbol{R}\Big) \underbrace{\frac{\partial \boldsymbol{K}}{\partial \theta_j} \nabla \ln p(\boldsymbol{y}|\hat{\boldsymbol{f}})}_{=\boldsymbol{b}} = \boldsymbol{b} - \boldsymbol{K}\boldsymbol{R}\boldsymbol{b}.\end{aligned}$$

Finally, the partial derivative of the approximate log marginal likelihood with respect to the kernel hyperparameters is given by

$$\begin{aligned}\frac{\partial \ln q(\boldsymbol{y}|\boldsymbol{X})}{\partial \theta_j} = &\frac{1}{2} \boldsymbol{a}^\top \frac{\partial \boldsymbol{K}}{\partial \theta_j} \boldsymbol{a} - \frac{1}{2} \operatorname{tr}\left(\boldsymbol{R} \frac{\partial \boldsymbol{K}}{\partial \theta_j}\right) \\ &+ \left(-\frac{1}{2} \operatorname{diag}\Big(\operatorname{diag}(\boldsymbol{K}) - \operatorname{diag}(\boldsymbol{J}^\top \boldsymbol{J})\Big) \cdot \frac{\partial \boldsymbol{W}}{\partial \hat{f}_i}\right)^\top \Big(\boldsymbol{b} - \boldsymbol{K}\boldsymbol{R}\boldsymbol{b}\Big).\end{aligned}$$

Partial Derivatives with respect to the likelihood hyperparameters

We now consider the partial derivatives with respect to the likelihood hyperparameters. Like in Equation (28), it splits up into an explicit and implicit term. For the explicit part we utilize the factorization of the likelihood $p(\boldsymbol{y}|\hat{\boldsymbol{f}})$ to obtain

$$\frac{\partial}{\partial \theta_j} \ln p(\boldsymbol{y}|\hat{\boldsymbol{f}}) = \frac{\partial}{\partial \theta_j} \ln \prod_{i=1}^{n} p(y_i|\hat{f}_i) = \frac{\partial}{\partial \theta_j} \sum_{i=1}^{n} \ln p(y_i|\hat{f}_i) = \sum_{i=1}^{n} \frac{\partial}{\partial \theta_j} \ln p(y_i|\hat{f}_i).$$

The second term in the explicit part is equal to zero since no variable directly depends on a likelihood hyperparameter,

$$\frac{\partial}{\partial \theta_j}\left(-\frac{1}{2} \hat{\boldsymbol{f}}^\top \boldsymbol{K}^{-1} \hat{\boldsymbol{f}}\right) = 0.$$

For the third term, we have

$$\begin{aligned}
\frac{\partial}{\partial\theta_j}\left(-\frac{1}{2}\ln|\boldsymbol{B}|\right) &= -\frac{1}{2}\operatorname{tr}\left(\boldsymbol{B}^{-1}\frac{\boldsymbol{B}}{\partial\theta_j}\right) = -\frac{1}{2}\operatorname{tr}\left(\left(\boldsymbol{I}+\boldsymbol{K}\boldsymbol{W}\right)^{-1}\frac{\partial}{\partial\theta_j}\left(\boldsymbol{I}+\boldsymbol{K}\boldsymbol{W}\right)\right)\\
&= -\frac{1}{2}\operatorname{tr}\left(\left(\boldsymbol{I}+\boldsymbol{K}\boldsymbol{W}\right)^{-1}\boldsymbol{K}\frac{\partial\boldsymbol{W}}{\partial\theta_j}\right)\right)\\
&= -\frac{1}{2}\operatorname{tr}\left(\left(\boldsymbol{K}(\boldsymbol{K}^{-1}+\boldsymbol{W})\right)^{-1}\boldsymbol{K}\frac{\partial\boldsymbol{W}}{\partial\theta_j}\right)\right)\\
&= -\frac{1}{2}\operatorname{tr}\left(\underbrace{\left(\boldsymbol{K}^{-1}+\boldsymbol{W}\right)^{-1}}_{=\boldsymbol{K}-\boldsymbol{J}^\top\boldsymbol{J}}\underbrace{\boldsymbol{K}^{-1}\boldsymbol{K}}_{=\boldsymbol{I}}\frac{\partial\boldsymbol{W}}{\partial\theta_j}\right)\right)\\
&\overset{(27)}{=} -\frac{1}{2}\operatorname{tr}\left(\left(\boldsymbol{K}-\boldsymbol{J}^\top\boldsymbol{J}\right)\frac{\partial\boldsymbol{W}}{\partial\theta_j}\right)\right)\\
&= -\frac{1}{2}\operatorname{diag}\Big(\operatorname{diag}(\boldsymbol{K})-\operatorname{diag}(\boldsymbol{J}^\top\boldsymbol{J})\Big)\cdot\frac{\partial\boldsymbol{W}}{\partial\theta_j},
\end{aligned}$$

which yields the final expression for the explicit part, given by

$$\left.\frac{\partial\ln q(\boldsymbol{y}|\boldsymbol{X})}{\partial\theta_j}\right|_{\text{explicit}} = \sum_{i=1}^{n}\frac{\partial}{\partial\theta_j}\ln p(y_i|\hat{f}_i) - \frac{1}{2}\operatorname{diag}\Big(\operatorname{diag}(\boldsymbol{K})-\operatorname{diag}(\boldsymbol{J}^\top\boldsymbol{J})\Big)\cdot\frac{\partial\boldsymbol{W}}{\partial\theta_j}.$$

The implicit part is rather similar to the other implicit part but with marginal modifications.

$$\begin{aligned}
\frac{\partial\hat{\boldsymbol{f}}}{\partial\theta_j} &= \frac{\partial\boldsymbol{K}\nabla\ln p(\boldsymbol{y}|\hat{\boldsymbol{f}})}{\partial\theta_j} + \frac{\partial\boldsymbol{K}\nabla\ln p(\boldsymbol{y}|\hat{\boldsymbol{f}})}{\partial\hat{\boldsymbol{f}}}\frac{\hat{\boldsymbol{f}}}{\partial\theta_j}\\
&= \boldsymbol{K}\frac{\partial\nabla\ln p(\boldsymbol{y}|\hat{\boldsymbol{f}})}{\partial\theta_j} + \boldsymbol{K}\underbrace{\frac{\partial\nabla\ln p(\boldsymbol{y}|\hat{\boldsymbol{f}})}{\partial\hat{\boldsymbol{f}}}}_{=-\boldsymbol{W}}\frac{\partial\hat{\boldsymbol{f}}}{\partial\theta_j}\\
&= \boldsymbol{K}\frac{\partial\nabla\ln p(\boldsymbol{y}|\hat{\boldsymbol{f}})}{\partial\theta_j} - \boldsymbol{K}\boldsymbol{W}\frac{\partial\hat{\boldsymbol{f}}}{\partial\theta_j}.
\end{aligned}$$

This is equivalent to

$$\begin{aligned}
& \frac{\partial \hat{\boldsymbol{f}}}{\partial \theta_j} + \boldsymbol{K}\boldsymbol{W}\frac{\partial \hat{\boldsymbol{f}}}{\partial \theta_j} = (\boldsymbol{I} + \boldsymbol{K}\boldsymbol{W})\frac{\partial \hat{\boldsymbol{f}}}{\partial \theta_j} = \boldsymbol{K}\frac{\partial \nabla \ln p(\boldsymbol{y}|\hat{\boldsymbol{f}})}{\partial \theta_j} \\
\iff \quad & \frac{\partial \hat{\boldsymbol{f}}}{\partial \theta_j} = (\boldsymbol{I} + \boldsymbol{K}\boldsymbol{W})^{-1}\boldsymbol{K}\frac{\partial \nabla \ln p(\boldsymbol{y}|\hat{\boldsymbol{f}})}{\partial \theta_j} \\
& = \Big(\boldsymbol{I} - \boldsymbol{K}\underbrace{(\boldsymbol{W}^{-1} + \boldsymbol{K})^{-1}}_{=\boldsymbol{R}}\Big)\boldsymbol{K}\frac{\partial \nabla \ln p(\boldsymbol{y}|\hat{\boldsymbol{f}})}{\partial \theta_j} \\
& = \Big(\boldsymbol{I} - \boldsymbol{K}\boldsymbol{R}\Big)\underbrace{\boldsymbol{K}\frac{\partial \nabla \ln p(\boldsymbol{y}|\hat{\boldsymbol{f}})}{\partial \theta_j}}_{=\boldsymbol{d}} = \boldsymbol{d} - \boldsymbol{K}\boldsymbol{R}\boldsymbol{d}.
\end{aligned}$$

Sebastian Mair received a Master of Science in Computer Science as well as a Bachelor of Science in Mathematics from Technische Universität Darmstadt and a Bachelor of Science in Computer Science from Hochschule Darmstadt University of Applied Sciences in 2015, 2016 and 2013, respectively. He is currently a research assistant and PhD student at Leuphana Universität Lüneburg.

Ulf Brefeld is professor for Machine Learning at Leuphana Universität Lüneburg. Prior to joining Leuphana, he was joint professor for Knowledge Mining & Assessment at TU Darmstadt and at the German Institute for Educational Research (DIPF), Frankfurt am Main. Before, Ulf led the Recommender Systems group at Zalando SE and worked on machine learning at Universität Bonn, Yahoo! Research Barcelona, Technische Universität Berlin, Max Planck Institute for Computer Science in Saarbrücken, and at Humboldt-Universität zu Berlin. Ulf Brefeld received a Diploma in Computer Science in 2003 from Technische Universität Berlin and a Ph.D. (Dr. rer. nat.) in 2008 from Humboldt-Universität zu Berlin.