Max–Planck–Institut für biologische Kybernetik
Max Planck Institute for Biological Cybernetics

Technical Report No. 136

Approximate Inference for Robust Gaussian Process Regression

Malte Kuss[1], Tobias Pfingsten[1,2], Lehel Csató[1], Carl E. Rasmussen[1]

March 10, 2005

[1] Department Schölkopf {kuss,tpfingst,csatol,carl}@tuebingen.mpg.de
[2] Robert Bosch GmbH, Corporate Sector Research and Advance Engineering

This report is available in PDF–format via anonymous ftp at ftp://ftp.kyb.tuebingen.mpg.de/pub/mpi-memos/pdf/robust.pdf. The complete series of Technical Reports is documented at: http://www.kyb.tuebingen.mpg.de/techreports.html

Approximate Inference for Robust Gaussian Process Regression

Malte Kuss, Tobias Pfingsten, Lehel Csató, Carl E. Rasmussen

Abstract. Gaussian process (GP) priors have been successfully used in non-parametric Bayesian regression and classification models. Inference can be performed analytically only for the regression model with Gaussian noise. For all other likelihood models inference is intractable and various approximation techniques have been proposed. In recent years *expectation-propagation* (EP) has been developed as a general method for approximate inference. This article provides a general summary of how expectation-propagation can be used for approximate inference in Gaussian process models. Furthermore we present a case study describing its implementation for a new robust variant of Gaussian process regression. To gain further insights into the quality of the EP approximation we present experiments in which we compare to results obtained by *Markov chain Monte Carlo* (MCMC) sampling.

---

1 Introduction – Robustness & Bayesian Regression

To solve a real-world regression problem the analyst should carefully screen the data and use all prior information at hand in order to choose an appropriate regression model. The model is selected so as to approximate the beliefs about the data generating process. A mismatch seems unavoidable in practice. Robust regression methods can be understood as attempts to limit undesired distractions and distortions that result from this mismatch.

Robust regression is often associated with the notion of *outliers*, which refers to observations that are in some sense structurally conspicuous. Often the presence of such outliers is attributed to observational errors, e.g. data processing errors or failures of measuring instruments. Commonly a statistical model is called robust if it leads to conclusions which are insensitive to the occurrence of such outlier observations. Note that this implies that an observation can only be called an outlier relative to a given model. As Jaynes (2003, ch. 21) phrases it: "One seeks data analysis methods that are *robust*, which means insensitive to the exact sampling distribution of errors, as it is often stated, insensitive to the model, or are, *resistant*, meaning that large errors in small proportion of the data do not greatly affect the conclusions."

The Bayesian answer to robust regression, i.e. handling outliers, results automatically from the common statement that a model should be chosen so as to reflect all the analyst's beliefs and uncertainties. So a Bayesian regression model can be considered robust if it explicitly accounts for the potential existence of outliers. Therefore, unless the analyst has absolutely no doubt that the model he has accounts for all possible observations—in other words, unless he is certain that there *are* no outliers relative to that model—he should adjust the model to account explicitly for the potential occurrence of outliers. A convenient way to reflect this belief is a mixture model. Jaynes (2003, ch. 21) calls it a "two-model model" being a mixture of a model which accounts for the *regular* observations and a second model for explaining *outliers*. The "two-model model" will be the line of thought in the remainder of this paper.

Before we go on, we briefly describe inference in the framework of non-parametric Bayesian regression. By *inference* we refer to the process of updating our beliefs according to Bayes' rule, i.e. computing the posterior from likelihood and prior, integrating the information contained in observed data. In regression analysis the objective is to make inference related to a latent real-valued function $f(\boldsymbol{x})$ where $\boldsymbol{x} \in \mathbb{R}^D$. The non-parametric approach is to put a prior $p_0(f|\boldsymbol{\theta}_1)$ directly on the space of functions and to

do inference on f. The simplest and most common prior over functions is a Gaussian process, described in Section 2.

Inference about f is based on observed samples $y(\boldsymbol{x}) = f(\boldsymbol{x}) + \varepsilon$ which are corrupted by additive noise. We assume the noise term ε to be independent and identically distributed (iid.), leading to the joint likelihood

$$p(\boldsymbol{y}|f, \boldsymbol{X}, \boldsymbol{\theta}_2) = \prod_{n=1}^{N} p(y_n|f_n, \boldsymbol{x}_n, \boldsymbol{\theta}_2) \tag{1}$$

where $\boldsymbol{y} = [y_1, \ldots, y_N]^\top$ denotes the observed outputs, $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]^\top$ are the corresponding inputs, and $f_n = f(\boldsymbol{x}_n)$ are the latent function values. We introduce a set of parameters $\boldsymbol{\theta}_2$ to parameterise the likelihood $p(y|f, \boldsymbol{x}, \boldsymbol{\theta}_2)$.[1] For non-parametric Bayesian models the posterior over the f is computed according to Bayes' rule

$$p_{\text{post}}(f|\boldsymbol{X}, \boldsymbol{y}, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = \frac{p(\boldsymbol{y}|f, \boldsymbol{X}, \boldsymbol{\theta}_2)\; p_0(f|\boldsymbol{\theta}_1)}{p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2)} \tag{2}$$

where f is a random function and the parameters $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ are considered fixed. The denominator is the *evidence*, or *marginal likelihood* $p(\mathcal{D}|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$, which is the normalising constant of the product of likelihood and prior. Here $\mathcal{D} = \{\boldsymbol{X}, \boldsymbol{y}\}$ denotes the observed data and we use the slight abuse of notation $p(\mathcal{D}|\cdot)$ to mean $p(\boldsymbol{y}|\boldsymbol{X}, \cdot)$.

We now describe how we can construct a mixture likelihood—a two-model model—in order to obtain a robust Bayesian regression model wrt. outliers in y. Let $p_r(y_n|f_n, \boldsymbol{\theta}_2)$ denote a noise model which describes our beliefs about *regular* observations, like the typical error of a measuring instrument. Assume we cannot deny the potential existence of outliers. For these outliers we believe the distribution of errors $p_o(y_n|f_n, \boldsymbol{\theta}_2)$ to be different. If we use π to denote the fraction of outlier observations, we can combine both models

$$p(y_n|f_n, \boldsymbol{\theta}_2) = (1 - \pi)\; p_r(y_n|f_n, \boldsymbol{\theta}_2) + \pi\; p_o(y_n|f_n, \boldsymbol{\theta}_2)\;.$$

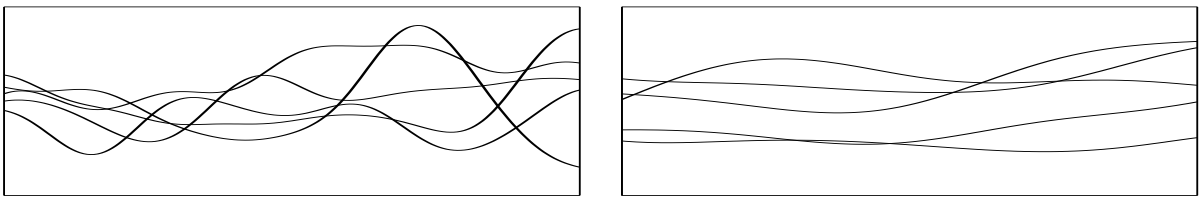
and obtain a mixture likelihood. In the following we consider the mixture of two Gaussian distributions. For *regular* observations we assume a relatively small variance σ_r^2 compared to the variance σ_o^2 of the outlier distribution. Thus the noise model is

$$p(y_n|f_n, \boldsymbol{\theta}_2) = (1 - \pi)\; \mathcal{N}(y_n|f_n, \sigma_r^2) + \pi\; \mathcal{N}(y_n|f_n, \sigma_o^2) \tag{3}$$

where $\boldsymbol{\theta}_2 = [\pi, \sigma_r^2, \sigma_o^2]$ collects the parameters. Assuming p_r to be Gaussian is a common and often plausible hypothesis. It seems more questionable to explain the outliers by Gaussian noise with relatively large variance. If we were certain that this were the case then the Gaussian mixture model would be correct and we would not call it robust. Generally, if we knew the outlier generating process, the notion of robustness would vanish. But the notion of an *outlier* involves a large uncertainty about their origin and distribution. Consequently, using a wide Gaussian distribution for p_o must be interpreted as a *back-up* model explaining observations which are highly unlikely to come from p_r.

In the following section we give an introduction to Gaussian process regression and describe why a direct application of Bayes rule is unfeasible for the proposed mixture noise model. Then we proceed by describing how the posterior process can be approximated using the *expectation propagation* method. For comparison we describe a *Markov chain Monte Carlo* approach to approximate inference in Section 4. Finally we describe experiments on several data sets in Section 5.

[1] We use $\boldsymbol{\theta}_1$ to refer to parameters of prior distributions and $\boldsymbol{\theta}_2$ to denote likelihood model parameters (other than $\boldsymbol{f}$) throughout the paper. For different models the actual parameterisation can differ.



(a) (b)

Figure 1: Samples from a zero-mean *GP* using a squared exponential covariance function (5). Figure (b) was generated with the squared length-scale w^2 six times larger than the one used in generating Figure (a). We observe that w can be interpreted as the characteristic length-scale at which the functions vary. Since the particular coordinates are not important, we omit axis labelling in the figures.

2 Inference in Gaussian Process Models

After specifying a likelihood (1) we have to assign a prior distribution to the latent function values to implement Bayesian inference. In Bayesian non-parametric regression we consider distributions on *any* collection of function values, the family of these distributions constituting a stochastic process.

As prior over functions we use Gaussian process (*GP*) priors below. Each input position $\boldsymbol{x} \in \mathbb{R}^D$ has an associated random variable $f(\boldsymbol{x})$. A Gaussian process over f technically means that the joint distributions of a collection $\boldsymbol{f} = [f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_N)]^\top$ associated to any input set $\boldsymbol{X}$ is multivariate Gaussian

$$p(\boldsymbol{f}|\boldsymbol{X}, \boldsymbol{\theta}_1) = \mathcal{N}(\boldsymbol{f}|\boldsymbol{\mu}, \boldsymbol{K}) \tag{4}$$

with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{K}$. A Gaussian process is specified by a mean function $\mu(\boldsymbol{x})$ and a positive-definite covariance[2] function $k(\boldsymbol{x}_i, \boldsymbol{x}_j|\boldsymbol{\theta}_1)$, such that $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j|\boldsymbol{\theta}_1)$ and $\boldsymbol{\mu} = [\mu(\boldsymbol{x}_1), \ldots, \mu(\boldsymbol{x}_N)]^\top$. By choosing a covariance function we introduce *hyper-parameters* $\boldsymbol{\theta}_1$ to the prior *GP*.

Several families of covariance functions are known in the literature, for example, see Abrahamsen (1997) or Schölkopf and Smola (2002, ch. 2). In the following we use a squared exponential covariance function of the form

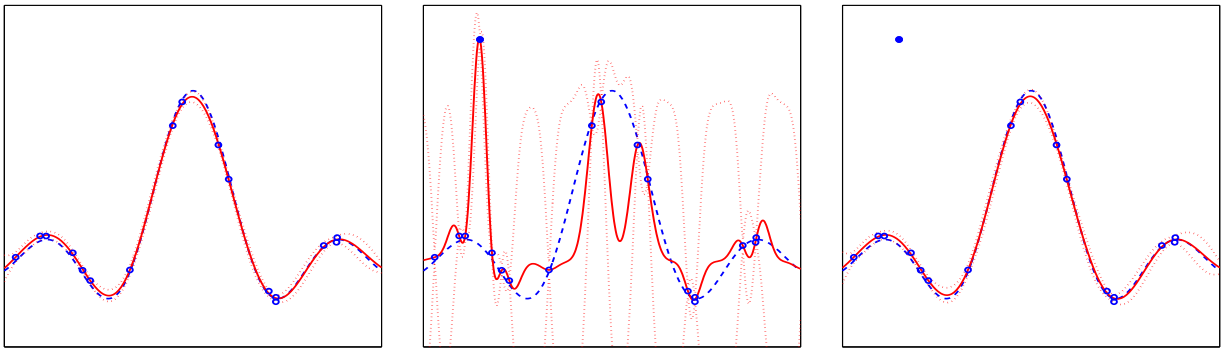
$$k(\boldsymbol{x}, \boldsymbol{x}'|\boldsymbol{\theta}_1) = \sigma_s^2 \exp\left(-\frac{1}{2}\sum_{d=1}^{D} \frac{\left(x^{(d)} - x'^{(d)}\right)^2}{w_d^2}\right) \tag{5}$$

where D is the dimension of inputs $\boldsymbol{x} \in \mathbb{R}^D$, σ_s^2 is the signal variance and $\boldsymbol{w} = [w_1, \ldots, w_D]^\top$ are scaling parameters, such that $\boldsymbol{\theta}_1 = [\sigma_s^2, \boldsymbol{w}]$. The effect of changing the length-scales $\boldsymbol{w}$ on the prior *GP* is illustrated in Figure 1. Note that having a scaling parameter for each input dimension allows the model to adjust the influence of the respective input variables—a concept which Neal (1996, ch. 1) calls *automatic relevance determination* (ARD).

In *GP* regression models inference over f is analytically tractable if the noise is assumed to be Gaussian. We now describe briefly how to obtain the posterior process, which is again a *GP* so this can be considered the conjugate setting. The derivation can be found in many introductory texts, e.g. Williams (1998), MacKay (2003, ch. 45) or O'Hagan (1994, 10.48) to mention only three. Nevertheless we repeat it here to contrast the difficulties that occur for other likelihoods, i.e. more complex noise models like the Gaussian mixture noise described in the previous section.

The model is $y(\boldsymbol{x}) = f(\boldsymbol{x}) + \varepsilon$ where $\varepsilon \sim \mathcal{N}(0, \sigma_r^2)$ and on the latent function f we put a *GP* prior with zero-mean and given covariance function (e.g. (5)). Given the observed $\boldsymbol{X}$ we write the likelihood

[2] We use the terms covariance function and kernel function interchangeably.



(a) (b) (c)

Figure 2: Illustration of *GP* regression and the effect of outliers. The dashed line shows the sinc function $f(x) = \sin(x)/x$ and the circles mark noisy samples thereof. Figure (a) shows the fit of a *GP* model with Gaussian noise. The posterior process is represented by its mean (solid line) and four standard deviations (dotted lines). Adding a single outlier (solid blue circle) affects highly the posterior *GP* with Gaussian noise – results shown in Figure (b). As before the mean function roughly interpolates the samples while the uncertainty about the function is increased dramatically. This can be explained as an effect of the inferred shorter length scale w and larger signal variance σ_s^2. Figure (c) shows the posterior *GP* obtained when the noise is modelled as a mixture of Gaussians. The methods used to generate the Figures are snMLII and mnMCMC as described in Section 5.

of $\boldsymbol{f}$

$$p(\boldsymbol{y}|\boldsymbol{f}, \boldsymbol{X}, \boldsymbol{\theta}_2) = \prod_{n=1}^{N} p(y_n|f(\boldsymbol{x}_n), \boldsymbol{\theta}_2) = \mathcal{N}(\boldsymbol{y}|\boldsymbol{f}, \boldsymbol{\Pi}) \tag{6}$$

where $\boldsymbol{\Pi} = \sigma_r^2 \boldsymbol{I}$ is a diagonal matrix with σ_r^2 on its diagonal entries and $\boldsymbol{\theta}_2 = \sigma_r^2$. According to the model conditioning the likelihood on $\boldsymbol{f}$ is equivalent to conditioning on the full function f.

The *posterior predictive distribution* of the latent function value f_* for an arbitrary test location $\boldsymbol{x}_*$ can be computed using standard results for multivariate normal distributions (Mardia et al., 1979, ch. 3). First we write the joint distribution under the *GP* prior $p_0(\boldsymbol{f}, f_*)$, compute the joint posterior distribution $p_{\text{post}}(\boldsymbol{f}, f_*|\mathcal{D}, \boldsymbol{x}_*)$ and marginalise $\boldsymbol{f}$ out to obtain $p_{\text{post}}(f_*|\mathcal{D}, \boldsymbol{x}_*)$. The posterior predictive distribution of f_* is again Gaussian $\mathcal{N}(f_*|\mu_{\text{post}}(\boldsymbol{x}_*), \sigma^2_{\text{post}}(\boldsymbol{x}_*))$ with the following mean and variance

$$\mu_{\text{post}}(\boldsymbol{x}_*) = \boldsymbol{k}(\boldsymbol{x}_*)^\top (\boldsymbol{K} + \boldsymbol{\Pi})^{-1} \boldsymbol{y} \tag{7a}$$

$$\sigma^2_{\text{post}}(\boldsymbol{x}_*) = k(\boldsymbol{x}_*, \boldsymbol{x}_*) - \boldsymbol{k}(\boldsymbol{x}_*)^\top (\boldsymbol{K} + \boldsymbol{\Pi})^{-1} \boldsymbol{k}(\boldsymbol{x}_*) \tag{7b}$$

where $\boldsymbol{k}(\boldsymbol{x}_*) = [k(\boldsymbol{x}_1, \boldsymbol{x}_*), \ldots, k(\boldsymbol{x}_N, \boldsymbol{x}_*)]^\top$. The posterior predictive distribution over f_* provides us a notion of the *uncertainty* of the model about the prediction, as illustrated in Figure 2. The above argumentation generalises to an arbitrary set of input locations, meaning that the posterior process on f is again a *GP* with mean function (7a) and posterior covariance function

$$k_{\text{post}}(\boldsymbol{x}, \boldsymbol{x}') = k(\boldsymbol{x}, \boldsymbol{x}') - \boldsymbol{k}(\boldsymbol{x})^\top (\boldsymbol{K} + \boldsymbol{\Pi})^{-1} \boldsymbol{k}(\boldsymbol{x}') \; . \tag{8}$$

So for any set of input locations $\boldsymbol{X}_*$ we can compute the posterior predictive distribution of the corresponding function values $\boldsymbol{f}_*$ which is multivariate normal $\mathcal{N}(\boldsymbol{f}_*|\boldsymbol{\mu}^*_{\text{post}}, \boldsymbol{K}^*_{\text{post}})$.

So far we have described inference over latent function values $f(\boldsymbol{x})$. We introduced the parameters $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ which were considered fixed. In a full Bayesian setting one should also perform inference over these parameters. Therefore we have to assign some prior distributions $p_0(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$ and write the posterior distribution of the parameters as

$$p_{\text{post}}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2|\mathcal{D}) \propto p(\mathcal{D}|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)\; p_0(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) \tag{9}$$

where $p(\mathcal{D}|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$ is the marginal likelihood, as it appeared in the denominator in equation (2). Usually we are not primarily interested in the posterior distribution of the parameters (9) and therefore could integrate them out from the joint posterior $p_{\text{post}}(f_*, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2|\mathcal{D}, \boldsymbol{x}_*)$ to obtain the predictive distribution over the function value $p_{\text{post}}(f_*|\mathcal{D}, \boldsymbol{x}_*)$. However, the calculations are analytically intractable. As will be shown in Section 4, this step can be approximated using sampling techniques.

Instead of doing inference, a computationally more attractive procedure is to find maximum-likelihood estimates for $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$. Following the *maximum likelihood II* (ML-II) scheme, values for the parameters are found by maximising the evidence $p(\mathcal{D}|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$, see Williams and Rasmussen (1996) or MacKay (1992) for details. Within the *GP* framework the logarithm of the evidence is

$$\ln p(\mathcal{D}|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = \ln \int d\boldsymbol{f}\; p(\boldsymbol{y}|\boldsymbol{f}, \boldsymbol{\theta}_2)\; p_0(\boldsymbol{f}|\boldsymbol{\theta}_1) \tag{10}$$

which in the case of Gaussian noise can be computed analytically:

$$\ln p(\mathcal{D}|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = -\frac{1}{2}\ln|\boldsymbol{K}(\boldsymbol{\theta}_1) + \boldsymbol{\Pi}(\boldsymbol{\theta}_2)| - \frac{1}{2}\boldsymbol{y}^\top \left(\boldsymbol{K}(\boldsymbol{\theta}_1) + \boldsymbol{\Pi}(\boldsymbol{\theta}_2)\right)^{-1}\boldsymbol{y} - \frac{N}{2}\ln(2\pi) \tag{11}$$

where the dependencies on the parameters have been made explicit. One can now maximise the log-evidence in $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ to find adequate values for the parameters given the observed data. The optimal parameters are not analytically computable but standard optimisation techniques, e.g. conjugate gradient, can be used to find a local maximum.

Gaussian process regression with Gaussian noise combined with ML-II parameter estimation has found many successful applications. Computationally the algorithm scales as $\mathcal{O}(n^3)$ and so several thousand observations can be handled without using further approximations. But changing the likelihood, e.g. for classification or by assuming a different noise model for regression, leads to analytically or computationally intractable inference problems. In those cases methods for approximate inference have to be applied.

In the case we consider in this paper, the noise is modelled as a mixture of two Gaussian distributions (3). The posterior becomes

$$p_{\text{post}}(\boldsymbol{f}|\mathcal{D}, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = \frac{p_0(\boldsymbol{f}|\boldsymbol{\theta}_1)}{p(\mathcal{D}|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)} \prod_{n=1}^{N} \left[(1-\pi)\,\mathcal{N}(y_n|f_n, \sigma_r^2) + \pi\,\mathcal{N}(y_n|f_n, \sigma_o^2)\right] \tag{12a}$$

where the evidence is

$$p(\mathcal{D}|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = \int d\boldsymbol{f}\; p_0(\boldsymbol{f}|\boldsymbol{\theta}_1) \prod_{n=1}^{N} \left[(1-\pi)\,\mathcal{N}(y_n|f_n, \sigma_r^2) + \pi\,\mathcal{N}(y_n|f_n, \sigma_o^2)\right]\ . \tag{12b}$$

This integral is analytically solvable but rewriting it in terms of Gaussian integrals involves a change in the order of summation and the product. This leads to a combinatorial explosion in the number of terms and the resulting posterior comes in the form of a mixture of 2^N normal distributions. Therefore, for real problems the large number of components makes it computationally intractable and we have to resort to approximations. Note that the posterior process for the mixture noise model is not a *GP* anymore. In fact it becomes a mixture of Gaussian processes, i.e. $p_{\text{post}}(f_*|\mathcal{D}, \boldsymbol{x}_*)$ becomes a mixture of Gaussian distributions which can be multi-modal as illustrated in Figure 3.

Various approximation techniques have been proposed that facilitate the implementation of *GP* models for inference tasks in which the posterior cannot be computed analytically. For example in the case of *GP* classification Williams and Barber (1998) propose a Laplace approximation, Gibbs and MacKay (2000) use variational techniques, and Opper and Winther (2000) apply mean field methods. For *GP* regression

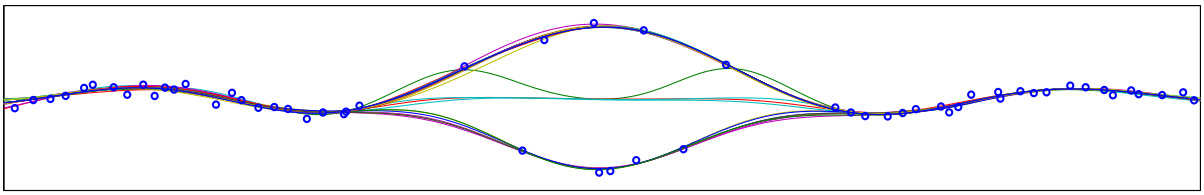


Figure 3: Sampled functions from the posterior process of a *GP* model with Gaussian mixture noise. The data (circles) have been designed in order to offer multiple alternative hypotheses to explain the data. Accordingly, the model shows uncertainty about whether the observations in the upper- and/or lower arc should be considered *outliers*. Therefore several hypotheses are mixed which leads to a multimodalilty in the conflicting region. The samples have been generated using MCMC as described in Section 4.

Neal (1997) describes an MCMC scheme for implementing robust regression using t-distributed noise. In the context of the *Relevance Vector Machine* framework a variational approach has been proposed for mixture noise models and t-distributed noise by Faul and Tipping (2001) and Lawrence and Tipping (2003) respectively.

In this article we implement and compare two methods for approximate inference for the *GP* regression model with mixture noise: an analytic *expectation-propagation* approximation (Section 3) and a sampling based approximation using *Markov chain Monte Carlo* techniques (Section 4).

3 Expectation Propagation for Gaussian Process Models

As described above, for non-Gaussian likelihoods computing the posterior in *GP* models becomes intractable. In this section we describe *expectation propagation* (EP) as a method to approximate the posterior by a Gaussian distribution. We start with a general description of the method and later exemplify its use for the mixture noise *GP* regression model in combination with ML-II parameter estimation. In describing the algorithm we follow Opper and Winther (2000) and Minka (2001).
Expectation propagation aims at minimising the Kullback-Leibler (KL) divergence

$$\mathrm{KL}\left[p_{\mathrm{post}}(\boldsymbol{f}) \| \mathcal{N}(\boldsymbol{f}|\boldsymbol{m}, \boldsymbol{A})\right] = \int d\boldsymbol{f}\; p_{\mathrm{post}}(\boldsymbol{f}) \ln\left(\frac{p_{\mathrm{post}}(\boldsymbol{f})}{\mathcal{N}(\boldsymbol{f}|\boldsymbol{m}, \boldsymbol{A})}\right) \tag{13}$$

between the posterior distribution $p_{\mathrm{post}}(\boldsymbol{f})$ and its Gaussian approximation with mean $\boldsymbol{m}$ and covariance $\boldsymbol{A}$. The minimum of (13) is taken for a normal distribution that matches the posteriors moments, $\boldsymbol{m} = \boldsymbol{\mu}^{\mathrm{post}}$ and $\boldsymbol{A} = \boldsymbol{\Sigma}^{\mathrm{post}}$,where $\boldsymbol{\mu}^{\mathrm{post}}$ and $\boldsymbol{\Sigma}^{\mathrm{post}}$ denote mean and covariance of the posterior distribution[3]. For non-Gaussian likelihoods—just as the mixture model—calculation of the moments is not possible. Expectation propagation approximates those moments.

Starting point of EP is to impose a factorising structure

$$\mathcal{N}(\boldsymbol{f}|\boldsymbol{m}, \boldsymbol{A}) \propto p_0(\boldsymbol{f}) \prod_{n=1}^{N} t_n(f_n) \tag{14}$$

on the approximation which resembles the structure of a factorising likelihood times prior. Since the terms $t_n(f_n)$ depend only on a single f_n Seeger (2003) calls them *site* functions. As the approximated posterior is to be Gaussian, the site functions t_n have to be quadratic exponentials

$$t_n(f_n|\mu_n, \sigma_n^2, C_n) \stackrel{\text{def}}{=} C_n \exp\left(-\frac{(f_n - \mu_n)^2}{2\sigma_n^2}\right) . \tag{15}$$

[3]This result can be obtained by writing the multivariate normal as $p_{\text{appr}}(\boldsymbol{f}|\alpha, \boldsymbol{\beta}, \boldsymbol{\Lambda}) = \exp\left(\alpha + \boldsymbol{\beta}^T\boldsymbol{f} + \boldsymbol{f}^T\boldsymbol{\Lambda}\boldsymbol{f}\right)$ and using a Lagrange parameter on the normalisation constraint.

There is no need for the individual sites $t_n(f_n)$ to be normalizable, so we do not restrict them to be distributions. It is sufficient to constrain the resulting covariance matrix $\boldsymbol{A}$ to be positive-definite, i.e. $\mathcal{N}(\boldsymbol{f}|\boldsymbol{m}, \boldsymbol{A})$ to be a proper distribution.

For the Gaussian approximation constrained to the form of (14) the mean and covariance are

$$\boldsymbol{m} \stackrel{\text{def}}{=} \boldsymbol{A}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} \quad , \quad \boldsymbol{A} \stackrel{\text{def}}{=} \left(\boldsymbol{K}^{-1} + \boldsymbol{\Sigma}^{-1}\right)^{-1} \tag{16a}$$

where $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_N)^\top$ and $\boldsymbol{\Sigma} = \operatorname{diag}(\sigma_1^2, \ldots, \sigma_N^2)$. The objective is to minimise the KL-divergence (13)

$$\{\boldsymbol{\mu},\, \boldsymbol{\Sigma}\} = \operatorname*{argmin}_{\boldsymbol{\mu},\boldsymbol{\Sigma}} \operatorname{KL}\left[p_{\text{post}}(\boldsymbol{f}) \| \mathcal{N}(\boldsymbol{f}|\boldsymbol{m}, \boldsymbol{A})\right] ; \text{ where } \boldsymbol{m}, \boldsymbol{A} \text{ are given by (16a)} . \tag{16b}$$

Note that since $\boldsymbol{A}$ is constrained, it might not be possible to match all elements of the covariance of the posterior $\boldsymbol{\Sigma}^{\text{post}}$ exactly.

Before we give a formal derivation of the EP algorithm, we provide an intuitive description of how the approximation $\mathcal{N}(\boldsymbol{f}|\boldsymbol{m}, \boldsymbol{A})$ is found. Assume we have given all but the nth site function such that the following approximation holds:

$$p_0(\boldsymbol{f}) \prod_{j \neq n} p(y_j|f_j, \boldsymbol{\theta}_2) \quad \approx \quad p_0(\boldsymbol{f}) \prod_{j \neq n} t(f_j) \, . \tag{17}$$

We aim at finding parameters of the remaining t_n such that the resulting approximation (14) is as close to the posterior as possible. Trying to match all m_i and A_{ij} to the moments of the posterior is too ambitious as we have only two parameters μ_n and σ_n. Therefore we restrict ourselves to matching the moments m_n and A_{nn}. This corresponds to finding the parameters μ_n and σ_n such that $\langle f_n^k \rangle_{\mathcal{N}(\boldsymbol{f}|\boldsymbol{m},\boldsymbol{A})} = \langle f_n^k \rangle_{\text{post}}$ for $k = 1, 2$. Assuming (17) holds we approximate the posterior moments:

$$\langle f_n^k \rangle_{\text{post}} = \frac{1}{p(\mathcal{D}|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)} \int d\boldsymbol{f} \; f_n^k \; p(y_n|f_n, \boldsymbol{\theta}_2) \; p_0(\boldsymbol{f}) \prod_{j \neq n} p(y_j|f_j, \boldsymbol{\theta}_2) \tag{18a}$$

$$\approx \frac{1}{p(\mathcal{D}|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)} \int d\boldsymbol{f} \; f_n^k \; p(y_n|f_n, \boldsymbol{\theta}_2) \; p_0(\boldsymbol{f}) \prod_{j \neq n} t(f_j) \, . \tag{18b}$$

After calculating the integrals we can match the moments and find the corresponding parameter values of μ_n and σ_n. The EP scheme iteratively updates the site functions in random order until the system converges. Changing t_n affects all elements of $\boldsymbol{m}$ and $\boldsymbol{A}$ globally through (16a). At convergence the approximate posterior $\mathcal{N}(\boldsymbol{f}|\boldsymbol{m}, \boldsymbol{A})$ is guaranteed only to match posterior mean and diagonal entries of the covariance matrix exactly. However, as we imposed the posteriors factorising structure in (14), we can expect that this leads to a good global approximation in the sense of (13).

3.1 Derivation of Expectation Propagation

While the above description is an attempt to provide an intuition on how EP finds an approximation to the posterior, in this section we reformulate the derivation given by Opper and Winther (2000). The derivation of EP comes as a sequence of approximation steps which lead to a set of nonlinear equations that characterise the optimal parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ for the site terms. If we could find a Gaussian approximation $\mathcal{N}(\boldsymbol{f}|\boldsymbol{m}, \boldsymbol{A})$ that matched the first two moments of the posterior distribution exactly we would have the solution to (16), but due to the structural constraints in (16a) an exact match between moments might not be possible. Therefore instead of matching the mean and the complete covariance matrix, we only match means $\mu_n^{\text{post}} = \langle f_n \rangle_{\text{post}}$ and diagonal elements $\Sigma_{nn}^{\text{post}} = \operatorname{var}(f_n)_{\text{post}}$ of the covariance matrix.

So only the first and second moments $\langle f_n^k \rangle_{\text{post}}$, $k = 1, 2$ have to be computed. The kth moment of f_n under the posterior distribution is

$$\begin{aligned}
\langle f_n^k \rangle_{\text{post}} &= \frac{1}{p(\mathcal{D}|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)} \int d\boldsymbol{f}\ f_n^k\ p_0(\boldsymbol{f}) \prod_{j=1}^{N} p(y_j|f_j, \boldsymbol{\theta}_2) && \text{(19a)} \\
&= \frac{1}{p(\mathcal{D}|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)} \int df_n\ f_n^k\ p(y_n|f_n, \boldsymbol{\theta}_2) \int d\boldsymbol{f}^{\setminus n}\ p_0(\boldsymbol{f}) \prod_{j \neq n} p(y_j|f_j, \boldsymbol{\theta}_2) && \text{(19b)} \\
&\stackrel{\text{def}}{=} \frac{1}{p(\mathcal{D}|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)} \int df_n\ f_n^k\ p(y_n|f_n, \boldsymbol{\theta}_2)\ q^{\setminus n}(f_n) && \text{(19c)}
\end{aligned}$$

where in (19c) we have grouped and integrated out the $\boldsymbol{f}^{\setminus n}$ which denotes $\boldsymbol{f}$ *except* for the site variable f_n. We defined the *cavity* function $q^{\setminus n}(f_n)$ according to Opper and Winther (2000). The cavity function is proportional to the predictive distribution of f_n given all but the nth sample.

Since the definition of $q^{\setminus n}(f_n)$ in (19c) includes all likelihood terms except $p(y_n|f_n, \boldsymbol{\theta}_2)$, computing the integral is still unfeasible. Therefore we need an approximation to the cavity function. As we approximate the posterior by a Gaussian distribution we approximate the cavity function $q^{\setminus n}(f_n)$ by an unnormalised Gaussian distribution $\tilde{q}^{\setminus n}(f_n) \propto \mathcal{N}(f_n|\mu_{\setminus n}, \sigma^2_{\setminus n})$ (for details see Opper and Winther (2000)). Consequently we can identify

$$\tilde{q}^{\setminus n}(f_n) = \int d\boldsymbol{f}^{\setminus n}\ p_0(\boldsymbol{f}) \prod_{j \neq n} t_j(f_j). \tag{20}$$

The integral only involves quadratic exponentials and we can find analytic expressions for the parameters of $\tilde{q}^{\setminus n}(f_n)$. We have

$$\begin{aligned}
\tilde{q}^{\setminus n}(f_n)\ t(f_n) &= \int d\mathbf{f}^{\setminus n}\ \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K}) \prod_j t(f_j) && \text{(21a)} \\
&\propto \int d\mathbf{f}^{\setminus n}\ \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{K})\, \mathcal{N}(\mathbf{f}|\boldsymbol{\mu}, \boldsymbol{\Sigma}) \quad = \mathcal{N}(f_n|m_n, \mathbf{A}_{nn}) && \text{(21b)}
\end{aligned}$$

and therefore $\mathcal{N}(f_n|\mu_{\setminus n}, \sigma^2_{\setminus n}) t(f_n) \propto \mathcal{N}(f_n|m_n, \mathbf{A}_{nn})$ which we can solve for the parameters of the approximate cavity function:

$$\sigma^2_{\setminus n} = \left(\frac{1}{A_{nn}} - \frac{1}{\sigma_n^2} \right)^{-1} \qquad \text{and} \qquad \mu_{\setminus n} = \sigma^2_{\setminus n} \left(\frac{m_n}{A_{nn}} - \frac{\mu_n}{\sigma_n^2} \right) . \tag{22}$$

In (22) one identifies the covariance matrix $\boldsymbol{A} = (\boldsymbol{K}^{-1} + \boldsymbol{\Sigma}^{-1})^{-1}$ as a link between the equations for all sites.

The next step in the derivation of EP leads to a set of nonlinear equations that characterise the solution. Using approximation (20) we state expressions for moment matching $\langle f_n^k \rangle_{\mathcal{N}(\boldsymbol{f}|\boldsymbol{m}, \boldsymbol{A})} = \langle f_n^k \rangle_{\text{post}}$ for $k = 0, 1, 2$. The moments $\langle f_n^k \rangle_{\text{post}}$ of the approximate marginal posterior, including the true likelihood term

$$\begin{aligned}
Z_n^* &\stackrel{\text{def}}{=} \int df_n \quad \tilde{q}^{\setminus n}(f_n)\ p(y_n|f_n, \boldsymbol{\theta}_2) && \text{(23a)} \\
F_{\mu_n} &\stackrel{\text{def}}{=} \frac{1}{Z_n^*} \int df_n\ f_n\ \tilde{q}^{\setminus n}(f_n)\ p(y_n|f_n, \boldsymbol{\theta}_2) && \text{(23b)} \\
F_{\sigma_n^2} + F_{\mu_n}^2 &\stackrel{\text{def}}{=} \frac{1}{Z_n^*} \int df_n\ f_n^2\ \tilde{q}^{\setminus n}(f_n)\ p(y_n|f_n, \boldsymbol{\theta}_2)\ . && \text{(23c)}
\end{aligned}$$

must be calculated by solving the one-dimensional integrals. Being able to solve the integrals is crucial for applying EP. Since $\tilde{q}^{\backslash n}$ is Gaussian this can be done analytically for various functional forms of likelihoods $p(y_n|f_n, \boldsymbol{\theta}_2)$. For cases where there is no analytical solution, Seeger (2003, app. C) proposes calculating the one-dimensional integrals numerically using Gauss-Hermite quadrature.

We now equate these moments with the according moments $\langle f_n^k \rangle_{\mathcal{N}(\boldsymbol{f}|\boldsymbol{m},\boldsymbol{A})}$ of the approximation in which the site function replaces the likelihood term:

$$Z_n^* \stackrel{!}{=} \int df_n \quad \tilde{q}^{\backslash n}(f_n)\, t_n(f_n) \tag{24a}$$

$$F_{\mu_n} \stackrel{!}{=} \frac{1}{Z_n^*} \int df_n\, f_n\, \tilde{q}^{\backslash n}(f_n)\, t_n(f_n) \tag{24b}$$

$$F_{\sigma_n^2} + F_{\mu_n}^2 \stackrel{!}{=} \frac{1}{Z_n^*} \int df_n\, f_n^2\, \tilde{q}^{\backslash n}(f_n)\, t_n(f_n) \tag{24c}$$

Using basic Gaussian identities we obtain the following set of coupled nonlinear equations for the parameters C_n, μ_n and σ_n of the site function:

$$C_n = \frac{Z_n^*}{Z_n} \quad \text{with} \quad Z_n \stackrel{\text{def}}{=} \int df_n\, \tilde{q}^{\backslash n}(f_n)\, \exp\left(-\frac{(f_n - \mu_n)^2}{2\sigma_n^2}\right) \tag{25a}$$

$$\sigma_n^2 = \left(F_{\sigma_n^2}^{-1} - \sigma_{\backslash n}^{-2}\right)^{-1} \tag{25b}$$

$$\mu_n = \sigma_n^2 \left(F_{\sigma_n^2}^{-1} F_{\mu_n} - \sigma_{\backslash n}^{-2}\, \mu_{\backslash n}\right) . \tag{25c}$$

The solutions $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ form the fixed point of the above set of equations. They can be found by iteratively updating the parameters of the individual sites as shown in Algorithm 1. However, convergence cannot be guaranteed.

Algorithm 1 Expectation Propagation Scheme for Gaussian Process Models

Given: $\boldsymbol{K}$, $\boldsymbol{y}$, $p(y|f, \boldsymbol{\theta}_2)$, convergence tolerance ϵ
Initialise: $\mathbf{A} \leftarrow \mathbf{K}$ and site function parameters σ_n^2, μ_n
repeat
  for all site functions n in random order **do**
    1. Compute cavity distribution $\tilde{q}^{\backslash n}(f_n) = \mathcal{N}(f_n|\mu_{\backslash n}, \sigma_{\backslash n}^2)$ using equations (22).
    2. Compute moments $F_{\sigma_n^2}$, F_{μ_n}, Z_n and Z_n^* analytically or by numerical integration.
    3. Update the parameters of $t_n(f_n)$ according to (25)(a–c).
    4. Update $\mathbf{m}$ and $\mathbf{A}$ using (16a).
  end for
until absolute change in Z_n^*, σ_n^2 and μ_n smaller than ϵ

In the description of EP we have not restricted ourselves to a particular likelihood function. The algorithm was formulated using the expectations in (23) and update equations (25), where the $p(y_n|f_n, \boldsymbol{\theta}_2)$ denote the likelihood terms, and the cavity functions are approximated by Gaussians. For many likelihood models these integrals for Z_n^*, F_{μ_n} and $F_{\sigma_n^2}$ can be calculated analytically. Thus in order to use EP for a particular model one has to compute the moments (23) and plug the corresponding terms into Algorithm 1. In the following we describe how EP can be implemented for the *GP* regression model with mixture noise in conjunction with an ML-II parameter estimation.

3.2 Implementing EP for the Gaussian Process Mixture Model

While in the previous section the EP approximation scheme has been described for Gaussian process models from a rather abstract viewpoint we now concentrate on the implementation. For the mixture

noise model we work out all necessary steps in detail and state the according results for other prominent likelihood functions.

In the above sections we used mean μ_n and variance σ_n^2 to parametrize the site functions $t_n(f_n)$. This way, however, not all relevant cases are captured, for example the case of a constant $t_n(f_n) = C_n e^0$ cannot be represented. The natural parametrisation of the exponential family in contrast to moments not only encloses the whole set of possible functions but also leads to a very convenient algebra when handling these function. Seeger (2003) describes the necessary background in detail. In numerical implementation we therefore switch to these natural parameters $\bar{r}_n = \sigma_n^{-2}$ and $\bar{\bar{r}}_n = \sigma_n^{-2}\mu_n$ so that $t_n(f_n) = C_n \exp[-\frac{1}{2}(\bar{r}_n f_n^2 - \bar{\bar{r}}_n)]$.

Note that in the update equations (25) of the site function parameters we ignored the possibility that updates lead to an invalid, non-positive definite covariance matrix $\boldsymbol{A}$. In a numerical implementation in those cases one can soften the update according to $\sigma_n^{-2} \leftarrow \gamma\sigma_n^{-2} + (1-\gamma)\sigma_{n\,\text{old}}^{-2}$ choosing γ to be just small enough to obtain a positive definite $\boldsymbol{A}$. Each inner loop in Algorithm 1 gives new values to the parameters of one site term and the covariance matrix can efficiently be updated using rank one updates on its Cholesky decomposition, see Seeger (2003, app. A) for details.

For notational convenience in the remainder of this section we reparameterise the Gaussian mixture noise model (3) by setting $\pi_1 = (1-\pi)$, $\pi_2 = \pi$, $\sigma_1 = \sigma_r$ and $\sigma_2 = \sigma_o$ so that we can write everything in terms of sums. While in the following only a mixture of $k = 2$ Gaussian distributions will be considered, note that the equations below also generalise to mixtures of any number of Gaussian distributions.

3.2.1 Computing the Moments

In equations (23) we have left the moments Z_n^*, F_{μ_i} and $F_{\sigma_i^2}$ in their integral form. For the mixture of Gaussians they can be calculated analytically using the moment generating function (DeGroot and Schervish, 2002, ch. 4.4), the same procedure applies for many other models. We have

$$M_n(\lambda) \stackrel{\text{def}}{=} \int df_n \; e^{\lambda f_n} \; \tilde{q}^{\backslash n}(f_n) \; p(y_n|f_n, \boldsymbol{\theta}_2) = N^{\backslash n} \sum_j \pi_j \frac{e^{g_{n,j}(\lambda)}}{\sqrt{2\pi}\sqrt{\sigma_j^2 + \sigma_{\backslash n}^2}} \tag{26a}$$

where $N^{\backslash n}$, $\mu_{\backslash n}$ and $\sigma_{\backslash n}^2$ are, respectively, the normalising constant, mean and variance of the cavity function $\tilde{q}^{\backslash n}(f_n)$ and

$$g_{n,j}(\lambda) = -\frac{1}{2}\left[\left(\frac{\mu_{\backslash n}}{\sigma_{\backslash n}}\right)^2 + \left(\frac{y_n}{\sigma_j}\right)^2 - \left(\frac{1}{\sigma_j^2} + \frac{1}{\sigma_{\backslash n}^2}\right)^{-1}\left(\frac{\mu_{\backslash n}}{\sigma_{\backslash n}^2} + \frac{y_n}{\sigma_j^2} + \lambda\right)^2\right] . \tag{26b}$$

We obtain the moments as derivatives of the generating function:

$$Z_n^* = M_n(0) \tag{27a}$$

$$F_{\mu_n} = \frac{M_n'(0)}{Z_n^*} = \frac{N^{\backslash n}}{Z_n^*} \sum_j \pi_j \frac{g_{n,j}'(0) \; e^{g_{n,j}(0)}}{\sqrt{2\pi}\sqrt{\sigma_j^2 + \sigma_{\backslash n}^2}} \tag{27b}$$

$$F_{\sigma_n^2} + F_{\mu_n}^2 = \frac{M_n''(0)}{Z_n^*} = \frac{N^{\backslash n}}{Z_n^*} \sum_j \pi_j \frac{(g_{n,j}'^2(0) + g_{n,j}''(0)) \; e^{g_{n,j}(0)}}{\sqrt{2\pi}\sqrt{\sigma_j^2 + \sigma_{\backslash n}^2}} . \tag{27c}$$

These equations constitute all we need to implement EP. For each update step in Algorithm 1 we pick a site, use equations (27) to compute the moments and solve for the parameters of the site terms according to equations (25).

3.2.2 ML-II Parameter Estimation

So far we have only described how to find the approximate posterior for given $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$. Having found it, we can also calculate an approximation to the evidence $p(\mathcal{D}|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$ which allows us to implement ML-II parameter estimation. The evidence can be approximated using (14) and the terms in (25a). It has a form corresponding to the one in (11) with likelihoods replaced by site functions, and factors including Z_n and Z_n^* from the EP approximation:

$$\begin{aligned}
\ln p(\mathcal{D}|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) &= \ln \int d\boldsymbol{f}\; p_0(\boldsymbol{f}|\boldsymbol{\theta}_1) \prod_{n=1}^{N} p(y_n|f_n, \boldsymbol{\theta}_2) && \text{(28a)} \\
&\approx \ln \int d\boldsymbol{f}\; p_0(\boldsymbol{f}|\boldsymbol{\theta}_1) \prod_{n=1}^{N} t_n(f_n) && \text{(28b)} \\
&= \frac{1}{2}\sum_{n=1}^{N} \ln \sigma_n^2 + \sum_{n=1}^{N} \ln C_n - \frac{1}{2}\ln|\boldsymbol{\Sigma} + \boldsymbol{K}| - \frac{1}{2}\boldsymbol{\mu}^\top (\boldsymbol{\Sigma} + \boldsymbol{K})^{-1}\boldsymbol{\mu}. && \text{(28c)}
\end{aligned}$$

The evidence depends on both the hyper-parameters of the *GP* prior $\boldsymbol{\theta}_1$ and of the model parameters $\boldsymbol{\theta}_2$. In our EP implementation for the *GP* mixture model we optimise the approximate evidence (28b) wrt. $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ using a conjugate gradient scheme. In this optimisation EP is used to compute the approximate evidence and its gradients for given parameter values.

The gradients of the evidence wrt. the parameters can be calculated analytically as follows. It can be shown[4] that at the fixed point approached by EP the derivatives with respect to the site parameters vanish:

$$\frac{\partial \ln p(\mathcal{D}|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)}{\partial(\mu_n, \sigma_n)} = 0\ . \tag{29}$$

This means that when differentiating wrt. $\boldsymbol{\theta}_{1,2}$ we only have to take explicit dependencies into account. We can therefore neglect changes induced by changing $\mu_n(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$ and $\sigma_n(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$, and give analytic expressions for the gradient:

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{\theta}_1} \ln p(\mathcal{D}|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) &= -\frac{1}{2}\frac{\partial}{\partial \boldsymbol{\theta}_1}\left(\ln|\boldsymbol{\Sigma} + \boldsymbol{K}(\boldsymbol{\theta}_1)| + \boldsymbol{\mu}^\top (\boldsymbol{\Sigma} + \boldsymbol{K}(\boldsymbol{\theta}_1))^{-1}\boldsymbol{\mu}\right) && \text{(30a)} \\
\frac{\partial}{\partial \boldsymbol{\theta}_2} \ln p(\mathcal{D}|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) &= \frac{\partial}{\partial \boldsymbol{\theta}_2}\sum_{n=1}^{N} \ln Z_n^*(\boldsymbol{\theta}_2) && \text{(30b)}
\end{aligned}$$

Terms appearing in (30a) depend on $\boldsymbol{\theta}_2$ only via site parameters $\boldsymbol{\Sigma}$ and $\boldsymbol{\mu}$, and likewise the term in (30b) is independent of $\boldsymbol{\theta}_1$. For the *GP* mixture model the gradients (30) result to be

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{\theta}_1} \ln p(\mathcal{D}|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) &= -\frac{1}{2}\mathrm{tr}\left(\boldsymbol{Q}^{-1}\frac{\partial \boldsymbol{K}}{\partial \boldsymbol{\theta}_1}\right) + \frac{1}{2}\boldsymbol{\mu}^\top\left(\boldsymbol{Q}^{-1}\frac{\partial \boldsymbol{K}}{\partial \boldsymbol{\theta}_1}\boldsymbol{Q}^{-1}\right)\boldsymbol{\mu} && \text{(31a)} \\
\frac{\partial}{\partial \pi_j} \ln p(\mathcal{D}|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) &= \sum_n \frac{N^{\backslash n}}{Z_n^*}\frac{e^{g_{n,j}(0)}}{\sqrt{2\pi}\sqrt{\sigma_j^2 + \sigma_{\backslash n}^2}} && \text{(31b)} \\
\frac{\partial}{\partial \sigma_j} \ln p(\mathcal{D}|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) &= \sum_n \frac{N^{\backslash n}}{Z_n^*}\pi_j\,\sigma_j\, e^{g_{n,j}(0)}\left[\left(\frac{\mu_{\backslash n} - y_n}{\sigma_j^2 + \sigma_{\backslash n}^2}\right)^2 - \frac{1}{\left(\sigma_j^2 + \sigma_{\backslash n}^2\right)^{\frac{3}{2}}}\right] && \text{(31c)}
\end{aligned}$$

where $\boldsymbol{Q} = \boldsymbol{K} + \boldsymbol{\Sigma}$. For the *GP* mixture model EP does not converge for all values of the parameters. In those cases no evidence can be calculated and we have to resort to a workaround to make the conjugate

[4] See M. Seeger's note *Expectation Propagation for Exponential Families* from the author's web page.

ascent work. When EP fails to converge after a given number of iterations, a low evidence is returned which makes the optimiser search in other regions of the parameter space.

The presented scheme to link EP and gradient based ML-II optimisation can be conveniently generalised to other likelihood models. Just as the moments (27) can be adapted to new likelihood models, so can be the gradients (31).

3.2.3 Prediction

Once the ML-II parameters and the normal approximation to the posterior are found, the predictive distribution of f_* corresponding to input location $\boldsymbol{x}_*$ is again a normal distribution. Its moments can be calculated analogously to the case of Gaussian noise (7) where $\boldsymbol{y}$ is replaced by $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ replaces $\boldsymbol{\Pi}$.

4 Approximate Inference by Markov Chain Monte Carlo Sampling

In later experiments we are interested in how well the EP technique works for the Gaussian mixture noise regression model. We therefore describe a Markov chain Monte Carlo implementation which we use for comparison. We start this section with a short introduction of the general idea of using Markov chain Monte Carlo (MCMC) methods for approximate Bayesian inference (for details the reader is referred to Gilks and Richardson (1996) and Neal (1993)). At first, we will use a simplified notation to describe the basic concepts and later describe our implementation of an MCMC scheme for the *GP* regression model with mixture noise.

In a nutshell, assume a model $p(\mathcal{D}|\boldsymbol{\theta})$ where $\boldsymbol{\theta} \in \Theta$ denotes the parameters[5], Θ the parameter space, and $\mathcal{D}$ the observed data. In an inference step we update our beliefs about $\boldsymbol{\theta}$ in the light of observed data $\mathcal{D}$ according to Bayes rule

$$p_{\text{post}}(\boldsymbol{\theta}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{\theta})\; p_0(\boldsymbol{\theta})}{\int d\boldsymbol{\theta}\; p(\mathcal{D}|\boldsymbol{\theta})\; p_0(\boldsymbol{\theta})}\;. \tag{32}$$

Problems arise for most nontrivial models because we are unable to solve the integral in the denominator and so to obtain the posterior $p_{\text{post}}(\boldsymbol{\theta}|\mathcal{D})$ analytically.

Now the task is to find a method of approximate inference which is computationally feasible yet adequately accurate. As seen in the previous sections, one approach is to approximate the posterior by another distribution. Instead, in situations where $p(\mathcal{D}|\boldsymbol{\theta})p_0(\boldsymbol{\theta})$ can be evaluated we can use MCMC methods to generate samples $\boldsymbol{\theta}^{(i)}$ from the posterior distribution $p_{\text{post}}(\boldsymbol{\theta}|\mathcal{D})$ of the parameters. These samples can be used for inspection or for approximating expectations of a given function $h(\boldsymbol{\theta})$ wrt. the posterior distribution according to

$$\int d\boldsymbol{\theta}\; h(\boldsymbol{\theta})\; p_{\text{post}}(\boldsymbol{\theta}|\mathcal{D}) \approx \frac{1}{T}\sum_{i=1}^{T} h(\boldsymbol{\theta}^{(i)}) \tag{33}$$

where $\boldsymbol{\theta}^{(i)}$ are approximately independent samples from the posterior. In order to generate these samples a Markov Chain in the parameter space is constructed such that the distribution of the state $\mathbf{s} \in \Theta$ is asymptotically identical to the posterior distribution of the parameters $\boldsymbol{\theta}$. Then the Markov chain is simulated and its states are interpreted as samples from $p_{\text{post}}(\boldsymbol{\theta}|\mathcal{D})$. The challenge is to construct a Markov chain properly such that it explores the whole posterior distribution efficiently, in order to obtain a number of approximately independent samples $\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(T)}$ in reasonable time.

The basic technique to construct such a chain is the *Metropolis-Hastings* algorithm and practically all MCMC methods are refined versions thereof. Let $\mathbf{s}_t$ denote the state of the chain at time t. In order to

[5] Although we aim at describing the concepts independently of the later application, for the proposed *GP* regression model one can think of $\boldsymbol{\theta} = [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2]$.

find the next state $\mathbf{s}_{t+1}$ a candidate $\tilde{\mathbf{s}}_{t+1}$ is proposed as a sample from a *proposal distribution* $q(\tilde{\mathbf{s}}_{t+1}|\mathbf{s}_t)$. The proposal is accepted as the consecutive state of the Markov chain ($\mathbf{s}_{t+1} \leftarrow \tilde{\mathbf{s}}_{t+1}$) if

$$\frac{p(\mathcal{D}|\tilde{\mathbf{s}}_{t+1})\; p_0(\tilde{\mathbf{s}}_{t+1})\; q(\mathbf{s}_t|\tilde{\mathbf{s}}_{t+1})}{p(\mathcal{D}|\mathbf{s}_t)\; p_0(\mathbf{s}_t)\; q(\tilde{\mathbf{s}}_{t+1}|\mathbf{s}_t)} > a \tag{34}$$

where a is a sample from an uniform distribution on $[0, 1]$. Ignoring the role of q for a moment, the above algorithm has an intuitive interpretation. The decision whether $\tilde{\mathbf{s}}_{t+1}$ is accepted as next state depends on the ratio of the target distribution evaluated at $\tilde{\mathbf{s}}_{t+1}$ and $\mathbf{s}_t$. If this ratio is larger than one, i.e. $\tilde{\mathbf{s}}_{t+1}$ yields a higher value, the proposal is always accepted. Otherwise the probability of acceptance is equivalent to the ratio of values under the posterior.

While simulating the Markov chain, states that occur close-by in the chain are highly dependent. We therefore subsample $\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(T)}$ from the observed sequence $\mathbf{s}_1, \mathbf{s}_2, \ldots$ to obtain samples which are approximately independent and distributed according to the posterior.

Most refinements of this scheme are directed towards clever proposal distributions q so that the probability of accepting a proposed state is increased—so that the procedure is computationally more efficient—while the chain is moving around quickly in the support of the target distribution.

4.1 Sampling Scheme for the Gaussian Process Mixture Model

In the remainder of this section we describe our implementation of a MCMC scheme for the mixture noise *GP* regression model. Again, let $\boldsymbol{\theta}_1$ denote the parameters of the covariance function and $\boldsymbol{\theta}_2$ collect the parameters of the noise model. We presume $y_i = f(\boldsymbol{x}_i) + \varepsilon_i$ where ε_i is iid. according to a mixture of Gaussians. In order to model this we introduce a vector of binary indicator variables $\mathbf{c}$ such that

$$\varepsilon_i|c_i, \sigma_r^2, \sigma_o^2 \sim (1 - c_i)\mathcal{N}(0, \sigma_r^2) + c_i\mathcal{N}(0, \sigma_o^2) \tag{35}$$

and $\boldsymbol{\theta}_2 = [\mathbf{c}, \sigma_r^2, \sigma_o^2]$. So c_i indicates whether ε_i is attributed to the noise component with (larger) variance σ_o^2. This corresponds to the likelihood

$$p(\boldsymbol{y}|\boldsymbol{f}, \boldsymbol{\theta}_2) = \prod_{i=1}^{N} \left[(1 - c_i)\mathcal{N}(y_i|f_i, \sigma_r^2) + c_i\mathcal{N}(y_i|f_i, \sigma_o^2)\right] \; . \tag{36}$$

Note that in principle there are two possible implementations of MCMC for the Gaussian mixture model. Either one introduces the indicator variables $\mathbf{c}$ or represents the latent function values $\boldsymbol{f}$ explicitly as proposed by Neal (1997).

Again we use a Gaussian process prior $p_0(f|\boldsymbol{\theta}_1)$ with zero-mean and squared exponential covariance function (5) so that $\boldsymbol{\theta}_1 = [\sigma_s^2, \mathbf{w}]$. Note that we will also make inference over the elements of $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$.

We have to specify prior distributions for the parameters of interest—namely the elements of $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$. The c_i are Bernoulli variables $p(c_i|\pi) = \text{Bernoulli}(\pi)$ where π is the fraction of samples attributed to noise variance σ_o^2. On π we put a beta prior $p_0(\pi|\alpha, \beta) = \text{Beta}(\alpha, \beta)$ introducing two more hyper-parameters. Furthermore we use a log-normal prior $p_0(\ln \mathbf{w}|\sigma_w^2) = \mathcal{N}(\mathbf{0}, \boldsymbol{I}\sigma_w^2)$ on the ARD weights of the covariance function. For the signal variance σ_s^2 as well as for the noise variances σ_r^2 and σ_o^2 we use flat (constant, degenerate) priors. Let $\boldsymbol{\psi} = [\alpha, \beta, \sigma_w^2]$ denote the hyper-parameters. The inference step is

$$p_{\text{post}}(\boldsymbol{f}, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2|\mathcal{D}, \boldsymbol{\psi}) \propto p(\boldsymbol{y}|\boldsymbol{f}, \boldsymbol{\theta}_2)\; p_0(\boldsymbol{f}|\boldsymbol{X}, \boldsymbol{\theta}_1)\; p_0(\boldsymbol{\theta}_1|\boldsymbol{\psi})\; p_0(\boldsymbol{\theta}_2|\boldsymbol{\psi}) \tag{37}$$

and we can approximate the marginal distribution over function values by

$$p(\boldsymbol{f}|\mathcal{D}, \boldsymbol{\psi}) = \int p(\boldsymbol{f}, \boldsymbol{\theta}_1, \boldsymbol{\theta}_2|\mathcal{D}, \boldsymbol{\psi})\; d\boldsymbol{\theta}_1 d\boldsymbol{\theta}_2 \tag{38a}$$

$$\approx \frac{1}{T}\sum_{t=1}^{T} p(\boldsymbol{f}|\boldsymbol{\theta}_1^{(t)}, \boldsymbol{\theta}_2^{(t)}, \mathcal{D}, \boldsymbol{\psi}) \tag{38b}$$

where $\boldsymbol{\theta}_1^{(t)}$ and $\boldsymbol{\theta}_2^{(t)}$ are MCMC samples from the posterior $p_{\text{post}}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2|\mathcal{D}, \boldsymbol{\psi})$.

To generate these samples we have to construct a Markov chain whose states vector $\mathbf{s}_t = [\boldsymbol{\theta}_1^{(t)}, \boldsymbol{\theta}_2^{(t)}]$ corresponds to the parameters we want to sum over in equation (38b). The Markov chain is constructed according to the *Metropolis-Hastings* procedure. We now describe how proposal states $\tilde{\mathbf{s}}_{t+1}$ are generated. The implemented sampling scheme iterates between *Gibbs* updates for the indicator variables $\mathbf{c}$ and π and *Hamiltonian* (also known as *hybrid*) Monte Carlo updates for $\mathbf{w}$, σ_s^2, σ_r^2 and σ_o^2. We employ different sampling techniques to exploit efficiently the structure of the model. Algorithm 2 provides a schematic overview of the sampling scheme and each step will be described in detail below.

Algorithm 2 MCMC sampling scheme for *GP* regression with mixture noise

Given: $\mathcal{D}, \alpha, \beta, \sigma_w^2$, Number τ and size ϵ_τ of leapfrog steps for Hamiltonian updates
Initialisation
Sample π from Beta(α, β)
Sample $\mathbf{c}$ element wise from Bernoulli(π)
Find initial values for $\mathbf{w}$, σ_s^2, σ_r^2 and σ_o^2 (e.g. by maximising the evidence of a model with simple Gaussian noise and setting $\sigma_o^2 \leftarrow 2\sigma_r^2$)
$t \leftarrow 0$
for each step of the Markov chain we simulate **do**
  $t \leftarrow t + 1$
  Gibbs sampling of indicator variables
  for all c_i **do**
    Compute

$$\tilde{\pi}_i = \frac{p(\mathcal{D}|c_i = 1, \mathbf{c}^{\setminus i}, \sigma_r^2, \sigma_o^2, \boldsymbol{\theta}_1)\ \pi}{p(\mathcal{D}|c_i = 0, \mathbf{c}^{\setminus i}, \sigma_r^2, \sigma_o^2, \boldsymbol{\theta}_1)\ (1 - \pi) + p(\mathcal{D}|c_i = 1, \mathbf{c}^{\setminus i}, \sigma_r^2, \sigma_o^2, \boldsymbol{\theta}_1)\ \pi}$$

    Update c_i by a sample from Bernoulli$(\tilde{\pi}_i)$
  end for
  Sample π from Beta$(\alpha + |\mathbf{c}|, \beta + N - |\mathbf{c}|)$
  Hamiltonian updates
  Update $\boldsymbol{\theta}_1$, σ_r^2, and σ_o^2 using Hamiltonian MCMC (see code in MacKay (2003, p. 388))
  Save state $\mathbf{s}_t = [\pi^{(t)}, \boldsymbol{\theta}_1^{(t)}, \boldsymbol{\theta}_2^{(t)}]$
end for

First, however, we have to describe how the state $\mathbf{s}_1$ is initialised. We initialise π by a sample from its prior distribution Beta(α, β) and consecutively sample $\mathbf{c}$ element-wise from a Bernoulli(π). Since we did not specify proper prior distributions for $\mathbf{w}$, σ_r^2, σ_o^2 and σ_s^2 we could find initial values by random samples from a log-normal distribution. Alternatively we can use ML-II estimates for σ_r^2, σ_s^2 and $\mathbf{w}$ from a model with simple Gaussian noise. The initial value of σ_o^2 is simply set to $2\sigma_r^2$ afterwards. In the following we describe how we update the elements of the state—the value of the parameters—in the Markov chain.

4.1.1 The Gibbs Updates

Gibbs sampling is a common MCMC technique in which the state is updated dimension-wise by sampling from the conditional distributions (in the above notation this would be $p(\theta_i|\boldsymbol{\theta}^{\setminus i}, \mathcal{D})$). The method is very appealing since the proposed updates are always accepted and no further parameters are introduced (see again Gilks and Richardson (1996), Neal (1993, ch. 4) or MacKay (2003, ch. 29)). We can use this method to sample the fraction of outliers π and indicator variables $\mathbf{c}$. Therefore we have to sample from

the conditional distribution of c_i given all values of the other variables $p(c_i|\mathbf{c}^{\setminus i}, \sigma_r^2, \sigma_o^2, \boldsymbol{\theta}_1, \mathcal{D}, \boldsymbol{\psi})$ where $\mathbf{c}^{\setminus i}$ denotes all elements of $\mathbf{c}$ except for the ith. We can decompose this probability

$$p(c_i|\mathbf{c}^{\setminus i}, \sigma_r^2, \sigma_o^2, \boldsymbol{\theta}_1, \mathcal{D}, \boldsymbol{\psi}) = \frac{p(\mathcal{D}|\mathbf{c}, \sigma_r^2, \sigma_o^2, \boldsymbol{\theta}_1, \boldsymbol{\psi}) p(\mathbf{c}|\boldsymbol{\psi})}{p(\mathcal{D}|\sigma_r^2, \sigma_o^2, \boldsymbol{\theta}_1, \boldsymbol{\psi}) p(\mathbf{c}^{\setminus i}|\sigma_r^2, \sigma_o^2, \boldsymbol{\theta}_1, \mathcal{D}, \boldsymbol{\psi})} \tag{39}$$

and observe that $p(c_i|\mathbf{c}^{\setminus i}, \boldsymbol{\theta}_1, \sigma_r^2, \sigma_o^2, \mathcal{D}, \boldsymbol{\psi}) \propto p(\mathcal{D}|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\psi}) p(\mathbf{c}|\boldsymbol{\psi})$. Since c_i is a binary indicator it is Bernoulli distributed. The probability of success $\tilde{\pi}_i$ of this Bernoulli distribution can be computed by comparing $p(\mathcal{D}|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \boldsymbol{\psi}) p(\mathbf{c}|\boldsymbol{\psi})$ evaluated for $c_i = 1$ and $c_i = 0$. Terms independent of c_i cancel and we find $\tilde{\pi}_i$ by looking at the ratio

$$\tilde{\pi}_i = \frac{p(\mathcal{D}|c_i = 1, \mathbf{c}^{\setminus i}, \sigma_r^2, \sigma_o^2, \boldsymbol{\theta}_1)\ \pi}{p(\mathcal{D}|c_i = 0, \mathbf{c}^{\setminus i}, \sigma_r^2, \sigma_o^2, \boldsymbol{\theta}_1)\ (1 - \pi) + p(\mathcal{D}|c_i = 1, \mathbf{c}^{\setminus i}, \sigma_r^2, \sigma_o^2, \boldsymbol{\theta}_1)\ \pi} \tag{40}$$

which can be interpreted as the relative plausibility of the ith sample being an *outlier* given the current values of all other variables. Technically equation (40) compares the marginal likelihood evaluated for both values of c_i weighted by the current value of π.

The marginal likelihood $p(\mathcal{D}|\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$ can be computed according to (11) where the noise term $\boldsymbol{\Pi}(\boldsymbol{\theta}_2)$ becomes a diagonal matrix with entries

$$\Pi_{jj} = (1 - c_j)\sigma_r^2 + c_j \sigma_o^2 \tag{41}$$

reflecting the currently assumed noise on the observations respectively. So we can update c_i easily by a sample from Bernoulli$(\tilde{\pi}_i)$.

As will be discussed later, a drawback of Gibbs sampling is that variables cannot change in a coordinated way. We use *ordered overrelaxation* as described by Neal (1998) to improve the mixing behaviour.

The next step in the sampling scheme is a Gibbs update of the mixing proportion π by a sample from $p(\pi|\mathbf{c}, \alpha, \beta) = \text{Beta}(\alpha + |\mathbf{c}|, \beta + N - |\mathbf{c}|)$ where $|\mathbf{c}|$ is the sum over elements of $\mathbf{c}$. This is a standard result, since the beta distribution is conjugate to the binomial (see for example, O'Hagan (1994, ch. 1)).

4.1.2 The Hamiltonian Updates

For updating the part of the state corresponding to $\mathbf{w}$, σ_s^2, σ_r^2 and σ_o^2 we use Hamiltonian updates which utilise gradient information of the posterior distribution to propose samples which are more *likely* to be accepted. Figuratively speaking, the gradient of the unnormalised (log) posterior distribution shows the way to high density regions and Hamiltonian MCMC can be understood as a gradient ascent with added noise (MacKay, 2003, ch. 30).

All we have to compute is the value of the log-evidence (11), where $\boldsymbol{\Pi}$ is as described in equation (41) and the value of the log-prior. We also have to provide derivatives of these quantities wrt. the parameters of interest $\mathbf{w}$, σ_s^2, σ_r^2 and σ_o^2.

Hamiltonian MCMC needs (at least) two additional parameters: the number of *leapfrog* steps and the step size(s) (for details on the method see Neal (1993) or MacKay (2003, ch. 30)). Both parameters determine the speed at which the chain mixes, i.e. the speed at which the chain moves in the support of the posterior. In the experiments presented in Section 5 we first set the value of the step size. As a rough rule of thumb: since having a large step size is computationally cheaper than increasing the number of steps, we first increase the step size until the acceptance rate is down to $50\% - 70\%$ before increasing the number of steps to values such that the expected runtime remains bearable.

4.2 Prediction

Assume we have simulated the chain as described above and observed a sequence $\mathbf{s}_1, \mathbf{s}_2, \ldots$ of states. We inspect the convergence and mixing of the chain by plotting the parameters over time and computing the

autocorrelation (*acf*) of parameters (see Gilks and Richardson (1996, ch. 8) or Cowles and Carlin (1996) for practical aspects of monitoring convergence). We also discard the first part of the chain as "burn-in period" in which the parameters show a clear trend leading them from their initial values into their high posterior density region. As mentioned above, states close in the chain are likely to be highly dependent. Therefore we subsample the chain and obtain parameter configurations $[\boldsymbol{\theta}_1^{(i)}, \boldsymbol{\theta}_2^{(i)}]$ $i = 1, \ldots, T$ which we hope to be a representative sample from their posterior distribution. In order to make predictions for a test case $\boldsymbol{x}_*$ we simply have to average (33) the predictive distributions based on each of the T parameter configurations

$$p(f_*|\boldsymbol{x}_*, \mathcal{D}) \approx \frac{1}{T} \sum_{i=1}^{T} p(f_*|\boldsymbol{x}_*, \mathcal{D}, \boldsymbol{\theta}_1^{(i)}, \boldsymbol{\theta}_2^{(i)}) \; . \tag{42}$$

Since for each parameter set the predictive distribution is Gaussian we obtain a mixture of T Gaussian distributions. For parameters $\boldsymbol{\theta}_1^{(i)}$ and $\boldsymbol{\theta}_2^{(i)}$ the moments of $p(f_*|\boldsymbol{x}_*, \mathcal{D}, \boldsymbol{\theta}_1^{(i)}, \boldsymbol{\theta}_2^{(i)})$ can be calculated from (7) where we have to plug in the mixture noise term (41). The mean of the predictive distribution $p(f_*|\boldsymbol{x}_*, \mathcal{D})$ is equal to the mean of the T predictive means. The variance of the predictive distribution can be obtained as the variance of the mean predictions plus the mean of the predictive variances.

5 Experiments

In this section we report experiments in order to compare and discuss several regression models and their performance. We are interested in whether the proposed robust *GP* model leads to improved predictive performance. We also describe problems which have become apparent in practical implementations of the algorithms.

Another aspect of interest is to analyse empirically the quality of the EP approximation relative to approximate inference using MCMC. For *GP* regression with Gaussian mixture noise we compare the predictive performance of the EP approximation to the MCMC predictions. Theoretically the MCMC approach leads to asymptotically correct results as the number of posterior samples increases. But practically it is difficult to ascertain that the Markov chains converge during simulation and that the obtained samples are approximately iid. samples from the posterior.

In order to get a better absolute impression of the performance, we will compare the *GP* model with mixture noise to one with simple Gaussian noise. For this model we report results obtained by ML-II parameter estimation (see Section 2 or Williams and Rasmussen (1996)) and a MCMC treatment as described by Neal (1997).

Furthermore we report the performance of ϵ-*support vector regression* (SVR) using an RBF kernel (Schölkopf and Smola, 2002, ch. 9). This variant of support vector regression is based on the ϵ-*insensitive* loss function

$$|y_i - f(\boldsymbol{x}_i)|_\epsilon = \max\{0, |y_i - f(\boldsymbol{x}_i)| - \epsilon\} \tag{43}$$

which is summed over all $i = 1, \ldots, N$ training cases. This loss function is zero for residuals smaller than ϵ and linear in the absolute value of the residual otherwise. In SVR the sum of the ϵ-*insensitive* loss and a regularisation term is minimised. The ϵ-insensitive loss function is robust—in the sense of Huber (1981, ch. 7) or Rousseeuw and Leroy (1987, ch. 1)—similar to the L_1 loss. For details on the connection between SVR and frequentist robust estimators the reader is referred to Schölkopf and Smola (2002, ch. 9). The RBF kernel used in the experiments is similar to the squared exponential (5) where $\sigma_s^2 = 1$ and all elements of $\mathbf{w} = \mathbf{1}w$ have the same value, so that all input dimensions are weighted equally. This is a clear disadvantage compared to the ARD parameterisation we implemented in the *GP* models because the scaling of input variables becomes a sensitive issue. The algorithm has three parameters, i.e. the insensitivity parameter ϵ, a regularisation parameter C, and the width w of the RBF kernel. In the experiments we find values for all three parameters by 5-fold cross-validations on the

training data. We manually refine the parameter grids and repeat the cross-validation procedure until the performance on the training data stabilises. The performance of the estimated model is reported for a separate test set.

We will use the following abbreviations to refer to the different models in the comparison:

- OLS – ordinary least squares linear regression (Mardia et al., 1979, ch. 6).
- SVR – support vector regression with an ϵ-insensitive loss function (Schölkopf and Smola, 2002, ch. 9). We use the implementation provided by Chang and Lin (2001).
- snMLII – *GP* regression model with simple Gaussian noise where values for $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ are found by ML-II estimation using conjugate gradient optimisation (Williams and Rasmussen, 1996).
- snMCMC – *GP* regression model with simple Gaussian noise where we sample $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ from their respective posterior using MCMC (Neal, 1997). We use a wide log-normal prior for the elements of $\mathbf{w}$ while we use constant priors on all other parameters.
- mnEP – expectation propagation approximation of the posterior in the *GP* regression model with mixture noise (Section 3).
- mnMCMC – approximate inference in the *GP* regression model with mixture noise using MCMC (Section 4).

In the sampling based *GP* methods snMCMC and mnMCMC we perform approximate inference over the elements of $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ and so we have to specify prior distributions over their elements respectively. In order to be as fair as possible we used constant, uniform (e.g. $p_0(\pi) = \text{Beta}(1, 1)$) or very broad priors and we believe their influence on the posterior to be negligible relative to the influence of the observed samples.

For comparing the predictive performance of the various models we report the *root mean square error* (RMSE) and the *mean absolute error* (MAE). In case full predictive distribution is provided we state the *negative log predictive probability* (NLP) of the test cases. For artificial data sets these measures will be given for separate test sets, while for real-world datasets a 10-fold cross-testing will be used. Let $\boldsymbol{X}_*$ denote test inputs and $\boldsymbol{t}_*$ the corresponding test targets. The *root mean square error* is defined as

$$\text{RMSE}(\boldsymbol{t}_*, \boldsymbol{f}_*) = \sqrt{\frac{1}{N_*}\sum_{i=1}^{N^*}(t_i^* - \langle f_i^* \rangle)^2} \tag{44}$$

where N^* denotes the number of test cases. The RMSE can be highly dominated by a few large residuals, so we also report the *mean absolute error*

$$\text{MAE}(\boldsymbol{t}_*, \boldsymbol{f}_*) = \frac{1}{N_*}\sum_{i=1}^{N^*}|t_i^* - \langle f_i^* \rangle| \tag{45}$$

in which the influence of a single observation is linear. Gaussian process models provide predictive distributions for the latent function values $p(\boldsymbol{f}_*|\mathcal{D}, \boldsymbol{X}_*)$ and including the inferred noise we can compute $p(\boldsymbol{y}_*|\mathcal{D}, \boldsymbol{X}_*)$. By *negative log predictive probability* we refer to the average negative logarithmic value of the predictive distribution

$$\text{NLP}(\boldsymbol{t}_*, \boldsymbol{X}_*, \mathcal{M}) = -\frac{1}{N_*}\sum_{i=1}^{N_*}\ln p(t_i^*|\mathcal{M}, \boldsymbol{X}_*) \tag{46}$$

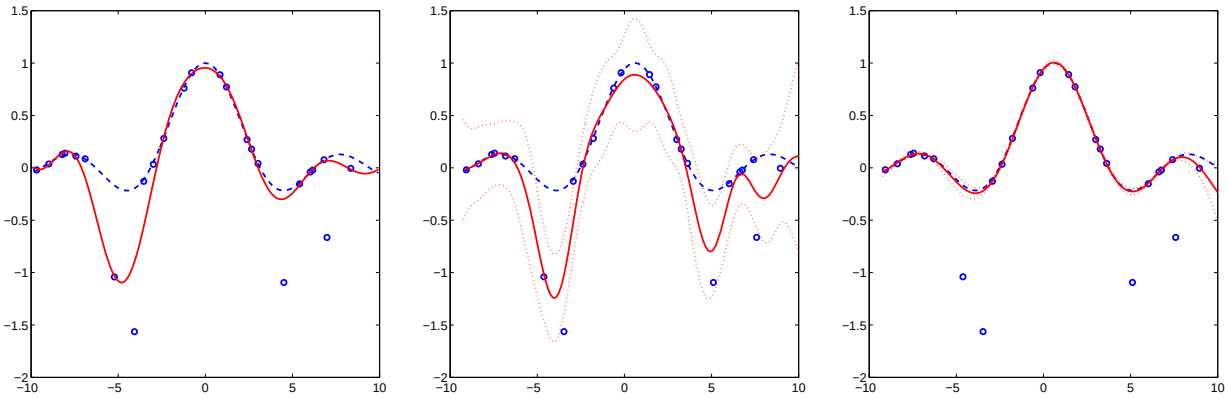


Figure 4: Three model fits to one of the generated sinc data sets. The dashed line describes the underlying sinc function and circles mark the training examples. Figure (a) shows the SVR fit. In Figure (b) the *GP* fit with a single Gaussian noise model (snMLII) is plotted. The solid line links the predictive means of the test cases and dotted lines are two standard deviations away. Figure (c) shows the MCMC fit of the mixture noise model (mnMCMC).

evaluated at the test samples where $\mathcal{M}$ denotes the model. The NLP is a measure of accuracy of the predictive distribution. For artificially generated data the test targets $\boldsymbol{t}_*$ are noise-free function values so we use the predictive distribution $p(\boldsymbol{f}_*|\mathcal{M}, \boldsymbol{X}_*)$, while for real world data sets only noisy test targets $\boldsymbol{y}_*$ are given and we use $p(\boldsymbol{y}_*|\mathcal{M}, \boldsymbol{X}_*)$.

5.1 One-dimensional Toy Problem

For illustration purposes, the first set of experiments we report are on artificially generated samples from the sinc function $y = \sin(x)/x + \varepsilon$ where ε is distributed according to our model assumptions. We generate 10 training sets of each $N = 25$ examples. The inputs x are uniformly sampled from the interval $[-10, 10]$. We compute the function values and subsequently pick five *outlier* samples randomly per set to which we add Gaussian noise with variance $\sigma_o^2 = 1$. To the remaining 20 samples we add Gaussian noise with variance $\sigma_r^2 = 10^{-4}$. This exactly corresponds to the noise mixture model described in Section 1. The test set consists of $N_* = 500$ noise free samples of the sinc function where the test inputs are uniformly sampled from the $[-10, 10]$ interval.

Running the algorithms on the 10 training sets it becomes apparent that the sets broadly vary in *difficulty*, i.e. we see a large variance in performance over the training sets. Three model fits to one of the training sets are illustrated in Figure 4. Summarising the performance on the 10 training sets Figure 5 shows box and whisker plots of the RMSE and MAE measures on the test set. The performances vary widely for different training sets. Screening the model fits for the individual training sets we observe that even the mixture noise model has large posterior uncertainty about the underlying function in one case. This one training set is difficult to fit for all the methods in the comparison and is responsible for the large span of the performance measures. Nevertheless, because we have generated the data accordingly, it comes with no surprise that the noise mixture model outperforms the other models in all three measures. Comparing mnMCMC and mnEP we obtain slightly better results on average for mnMCMC. Also the variance of the measure for mnMCMC appears to be smaller.

The simple Gaussian noise *GP* models (snMLII and snMCMC) shows serious difficulties in explaining the training data. The snMLII optimisations often produces solutions which have large predictive uncertainty while the mean function interpolates the training examples. Note that the optimisation problem is non-convex and therefore the occurrence of local maxima are a serious problem. For some training sets the optimisation even leads to estimates of $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$ such that all observations are explained as noise and the mean function of the posterior *GP* remains zero. Inspecting the parameter sets sampled using

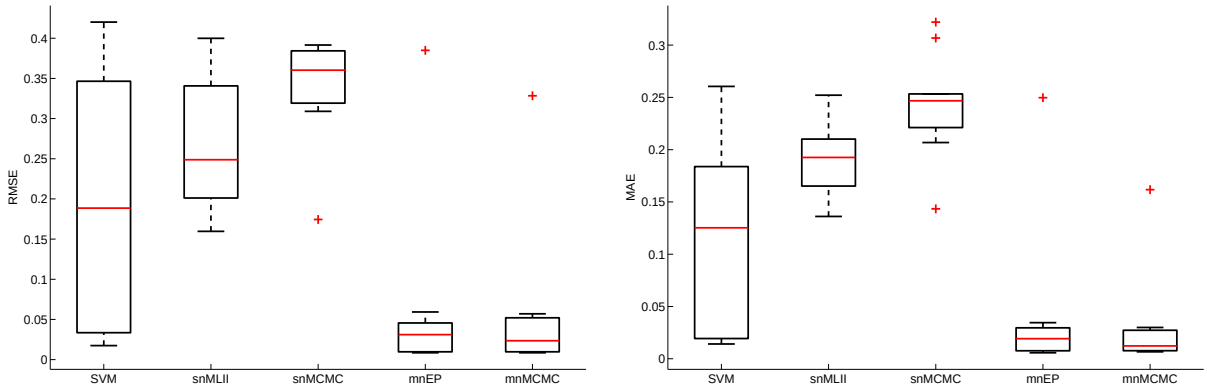


Figure 5: RMSE and MAE model comparison based on 10 artificially generated data sets from the sinc function. Figure (a) shows box and whisker plots of the root mean square errors obtained by the respective methods. The box and whisker plots illustrate the span, lower quartile, median, and upper quartile. Figure (b) shows box and whisker plots of the mean absolute errors. Both measures of fit show the same structure. Although the ranges are large it can be said that the noise mixture *GP* model clearly outperforms SVR and simple Gaussian noise *GP* models. OLS results are omitted (average RMSE $= 0.39$).

MCMC (snMCMC) it becomes apparent that the posterior uncertainty is large, and averaged over this uncertainty, the models shows poor predictive performance. In comparison, support vector regression performs relatively well on the data sets. Note that for one-dimensional inputs the covariance function of the *GP* model and the kernel in SVR only differ by the signal variance parameter σ_s^2 and the *GP* models do not profit from their ARD capability.

The mnEP implementation also relies on ML-II parameter estimation. Due to the non-convexity of the problem the conjugate gradient based optimisation can get stuck in local maxima, so that the parameter estimates can depend on the starting point. Performing several runs of mnEP from different starting points we have observed that the algorithm converges to different parameter configurations in some cases. A practical way of dealing with this problem is to run mnEP several times initialised with different parameter configurations and to pick the solution showing highest evidence (28b). All results on mnEP presented in this paper have been obtained by picking the solution that showed highest evidence from several, i.e. 3 to 5, runs of the algorithm. As discussed in Section 3.2.2, convergence of the EP algorithm is not guaranteed. In some cases, especially for difficult folds, we observe this problem.

The NLP measures for the mixture noise *GP* models give consistently better values than for simple Gaussian noise. Among the mixture noise models mnMCMC gives NLP $= -2.11$ averaged over the training sets which is slightly better than mnEP with NLP $= -2.04$. The NLP measures for simple Gaussian noise models are orders of magnitude larger which again indicates their inability to explain the data.

For model comparison we relate the average marginal likelihoods obtained by snMLII and mnEP on the training data. Let $p(\mathcal{D}|\mathcal{M})$ denote the marginal likelihood of model $\mathcal{M}$ on training set $\mathcal{D}$. We compute the log of the *marginal likelihood ratio* (MLR)

$$\text{logMLR} = \ln\left(\frac{p(\mathcal{D}|\text{mnEP})}{p(\mathcal{D}|\text{snMLII})}\right) = \ln p(\mathcal{D}|\text{mnEP}) - \ln p(\mathcal{D}|\text{snMLII}) \tag{47}$$

which averaged over the 10 training sets in our experiments gives a value of 19.7. Computing the average MLR per training example $\sqrt[N]{\text{MLR}} = 2.19$ we see that the MLR clearly favours the mnEP model. The value of the average MLR per training example can be interpreted in a sense that on average each training example is explained twice as well by the mnEP model than by snMLII. Note that the logMLR value is equivalent to the log posterior odds ratio when our prior belief in the models is equal (Jaynes, 2003, ch. 20).

5.2 Friedman Data

In this section we report experiments on artificially generated data which are derived from a problem introduced by Friedman (1991). Given 10-dimensional input vectors $\boldsymbol{x}$ the function value f depends on the first five input dimensions only

$$f(\boldsymbol{x}) = 10 \sin(\pi\, x_1\, x_2) + 20\,(x_3 - 0.5)^2 + 10 x_4 + 5 x_5\ , \tag{48}$$

while the purpose of the remaining input dimensions $x_6, \ldots, x_{10}$ is only to complicate the problem. We generate 10 training sets of $N = 100$ examples respectively. The inputs $\boldsymbol{x}$ are randomly sampled from the uniform distribution on the unit hyper-cube $[0, 1]^{10}$. We then compute the corresponding function values and add Gaussian noise with zero mean and unit variance. In each training set we replace 10 outputs by samples drawn from a normal distribution with mean $\mu = 15$ and variance $\sigma_o^2 = 9$. So we generate *outliers* which are unrelated to the function (48) but are likely to lie in the same range as the function values. For testing we generate a data set of 1000 noise-free samples.

In the experiments none of the implementations showed obvious difficulties. As described above, for mnEP we let the algorithm start with different initial values and picked the solutions with highest marginal likelihood. The results are presented in Figure 6. The RMSE and MAE measures show that the *GP* models with noise mixture perform consistently better than the ones with simple Gaussian noise. The performance of mnEP and mnMCMC appears to be very similar with a small advantage of the latter on average. The variances of RMSE and MAE over the training sets are small compared to the $sinc$-problem. This may be explained as a consequence of the larger signal-to-noise ratio and the increased homogeneity of the different training sets due to larger sample size. The *GP* models clearly benefit from the ARD capability, which allows them to ignore the input dimensions which do not prove to be informative about the output. The absence of a similar mechanism may explain the rather poor performance of support vector regression. The NLP measures in Figure 6(c) show that mnMCMC and mnEP provide similarly accurate predictive distributions. As one would expect, the *GP* models with simple Gaussian noise also exhibit worse NLP values.

Computing the average logMLR (47) over training sets between mnEP and snMLII we obtain a value of 16.64 which corresponds to an average MLR per training example of 1.18. This affirms that the mixture noise model explains the data better. The average MLR per example is lower than for the $sinc$ problem, which can be explained by the smaller fraction of outliers in the training data.

5.3 Boston Housing Data

We now report experiments carried out on the *Boston housing* data set. These data have been analysed by Harrison and Rubinfeld (1978) and since then the data-set has become a popular reference problem in nonlinear regression. The task is to predict the median price of houses in different parts of the Boston metropolitan area based on 13 input variables. The target variable appears to be truncated at \$50,000. For a more detailed description, the reader is referred to Neal (1996, ch. 4.4.2). The data set consists of $N = 506$ observations which we normalise to zero mean and unit variance. We then split the data into 10 folds. Since we want to compare to the results given by Neal (1996) we use exactly the same split. We use a 10 fold cross-testing procedure which means that each of the folds is left out once as a test set, while the remaining nine folds constitute the training data. The experimental results are illustrated in Figure 7. We state the RMSE and MAE values reported by Neal (1996, ch. 4.4.2) for a two hidden layer neural network with Gaussian priors on the network weights and t-distributed additive noise. In the cited study approximate Bayesian inference is performed over the weights in the neural network using the Hamiltonian MCMC method.

At first sight the *GP* models and the Bayesian neural network show similar performance wrt. average RMSE and MAE. The Bayesian neural network has a slightly lower average $(\text{RMSE} = 2.49)$ and less

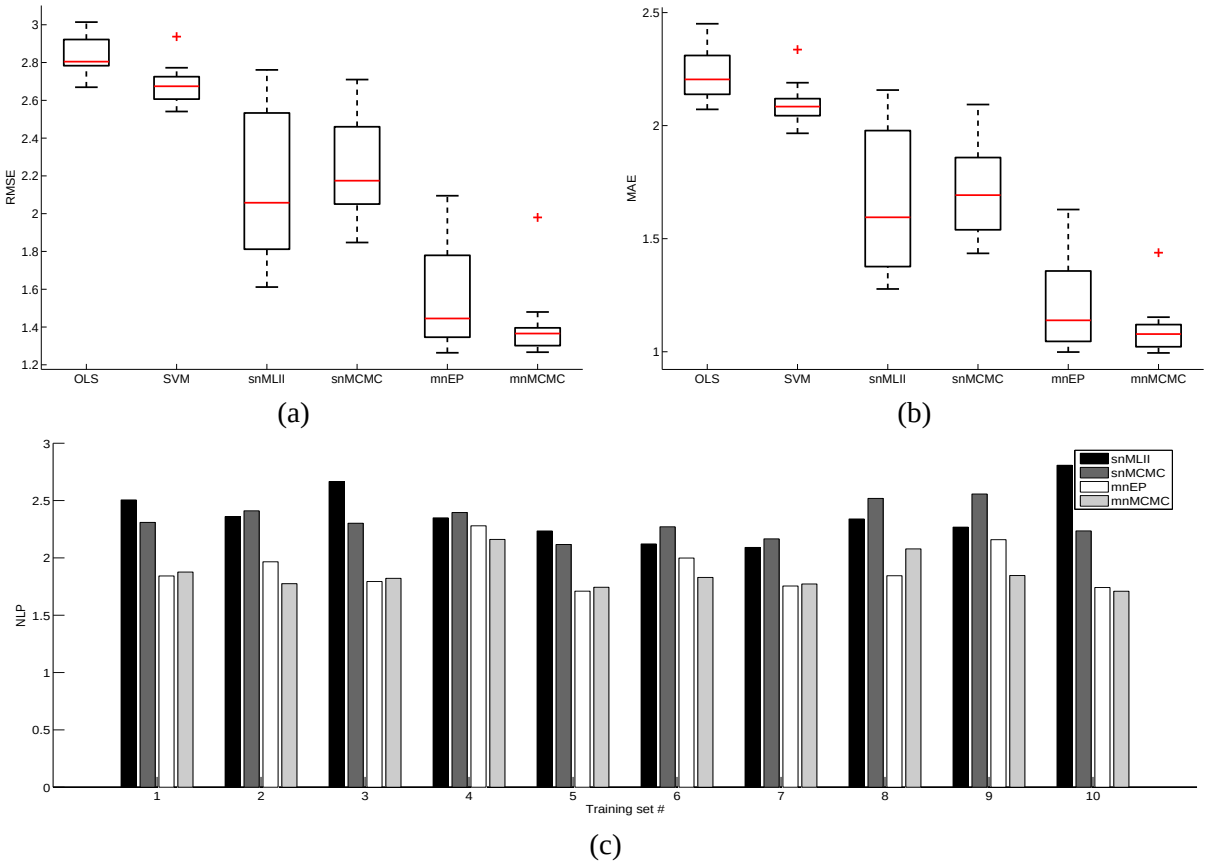


Figure 6: Results for RMSE, MAE and NLP on the Friedman data sets. Figure (a) shows box and whisker plots of the root mean square errors obtained by the respective methods. Figure (b) shows box and whisker plots of the mean absolute errors. In (c) we show the NLP measures of the test set for the models obtained from the individual training sets.

variance over the folds compared with the best *GP* model mnEP (RMSE $= 2.55$). Nevertheless, the *GP* models and the Bayesian neural network perform very similarly on average. Support vector regression gives worse results on average and larger variance over the folds. Other experiments using SVR on the Boston housing data—in a different experimental setting—can be found in Schölkopf and Smola (2002, ch. 9.6) and Stitson et al. (1999). Breaking the results down to the individual folds in Figure 7(c) we cannot observe a regular pattern anymore.

Inspecting the Markov chains of several mnMCMC simulations we observed that—especially for folds #5 and #7—the chains had not mixed properly, i.e. the state of the chain did not travel the support of the posterior evenly but rather infrequently switched between discrete areas. This behaviour indicates the presence of local modes of the posterior. The Markov chain should switch between the modes and sample from them proportionally. The indicator variables $\mathbf{c}$ appear to be the crucial factor. Recall that for each state of the chain the indicator variables mark observations which are considered *outliers*. Whether an observation is likely to be considered an outlier depends on the configuration of the other outliers. One can think of several configurations of outliers which are plausible under the posterior and so form local modes of the distribution. Intuitively these local modes can be understood as alternative hypotheses about which subset of samples are *outliers* and which *regular* samples have to be explained by the model (see again Figure 3). In order to switch between these hypotheses several indicator variables have to change their values in a coordinated manner. But the Gibbs updates we use in the mnMCMC sampling scheme allow the indicators to change one at a time given all the other indicators. This makes a switch between hypotheses very difficult. Dealing with this problem is difficult and outside the scope of this paper. We

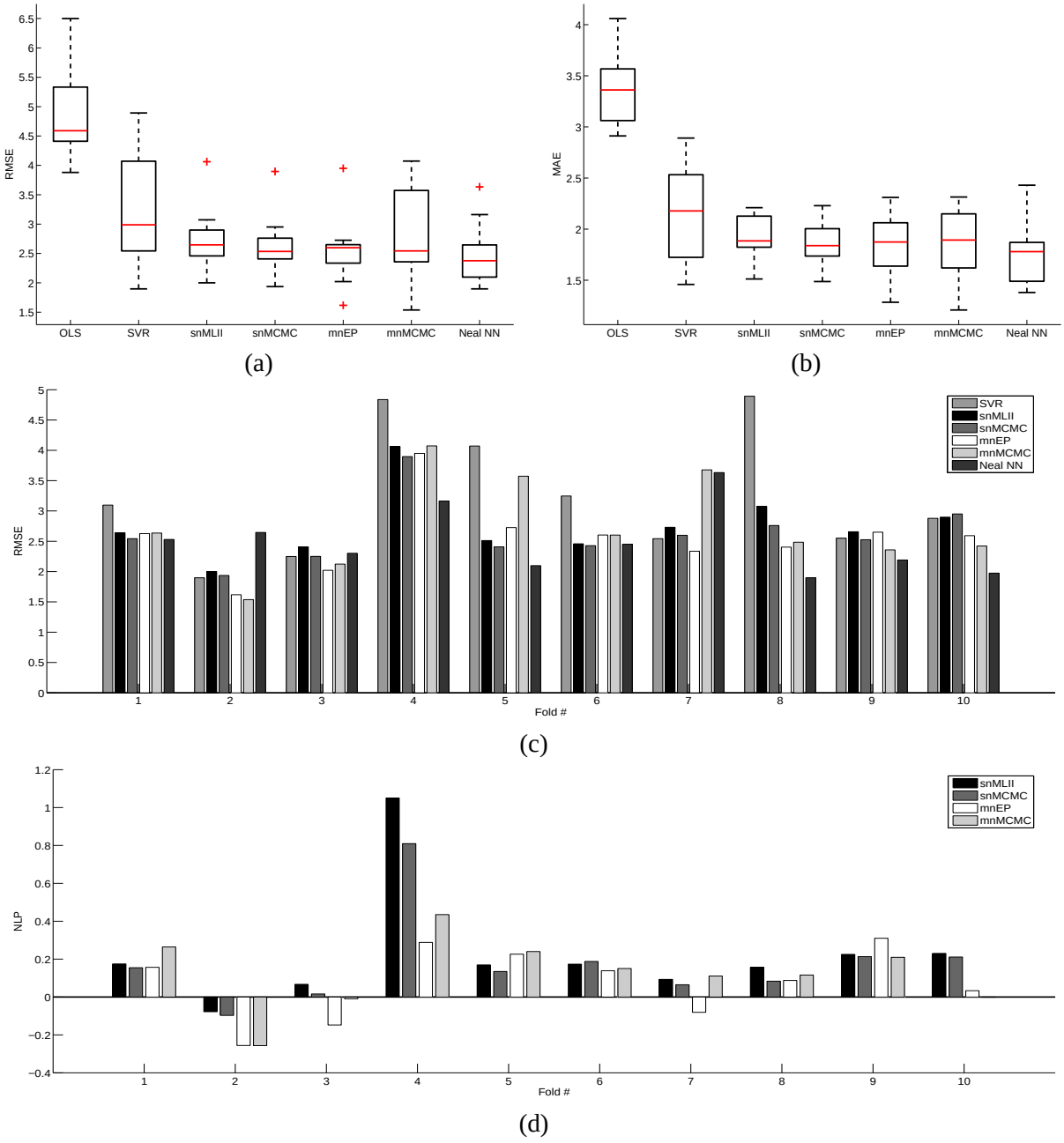


Figure 7: RMSE, MAE and NLP obtained by 10 fold cross-testing on the Boston housing data set. Figures (a) and (b) give on overview of the RMSE and MAE measures using box and whisker plots. The values for "Neal NN" are taken from Neal (1996, p. 134). Figure (c) shows the RMSE for the individual fold. For the *GP* models the NLP measures are given in Figure (d).

believe that this behaviour did not occur in the previous examples because the problems were such that one hypothesis clearly displaced the alternatives.

For the folds #5 and #7 the model shows uncertainty whether a few training samples with maximum (truncated) target value should be fitted as regular observations or should be ignored as outliers. Therefore the model is cautious to predict large y-values for test data which leads to a tendency to underestimate the value for test cases which indeed have maximum target value. This explains the large RMSE and MAE values for mnMCMC.

The simulation time per training set for mnMCMC was in the order of one night. For mnEP the ML-II parameter optimisations converged slowly and the runtime per fold was similar to mnMCMC, although each evaluation of the approximate marginal likelihood using EP took several minutes only.

The fraction of outliers π inferred by mnEP and mnMCMC lies in the range of $2\% - 4\%$ for all folds and for σ_o^2 we obtain values that are an order of magnitude larger than σ_r^2. For model comparison we compute the logMLR (47) between mnEP and snMLII which results in a value of 17.72. The corresponding average MLR per training example is $\sqrt[N]{\mathrm{MLR}} = 1.04$. It should be mentioned again that the value of the marginal likelihood for mnEP (28b) is an approximation. The average ratio of predictive densities evaluated at the test data

$$\frac{p(\mathcal{D}_*|\mathrm{mnEP})}{p(\mathcal{D}_*|\mathrm{snML})} = 1.16 \tag{49}$$

which can be calculated from the NLP values illustrated in Figure 7(d) also favours mnEP. The average RMSE for snMLII is 2.74 compared to 2.55 for mnEP, which is an improvement of 7.5%. Note that the average RMSE for snMCMC is 2.62 which advises caution when attributing the improvement to the noise model only. Thus we can conclude that the experiments provide some evidence that the mixture noise *GP* model is better suited to explaining the data.

6 Summary & Conclusions

In applied regression analysis the potential existence of *outliers* in the data can rarely be ruled out with certainty. In this situation the analyst should choose a model which takes this belief explicitly into account. As we argued above, one way of doing this is to use a mixture of models which in the simplest form leads to a "two-model model" approach, combining a model for *regular* observations and one for *outliers*.

In this work we addressed robustness in the context of *GP* regression. We proposed the use of a mixture of Gaussian noise model and described why analytic inference in this case becomes intractable. We then presented and compared two schemes for approximate inference. First expectation propagation approximation was described in general form for *GP* models and for the mixture noise model in particular. Second, for comparison we described how Markov chain Monte Carlo sampling can be implemented. We then compared the performance of the mixture noise *GP* model—or rather the two approximations thereof—and several other regression techniques on three data sets. In the description of experiments, some problems of the respective methods were already mentioned. In the remainder we summarise our conclusions:

- Experiments on artificially generated data show that the mixture noise *GP* model outperforms the other models in this comparison when outliers in y are present in the training data. The predictive performance of mnMCMC and mnEP was very similar in our experiments, indicating that the EP approximation works satisfyingly.

- In terms of RMSE the performance of *GP* models on the Boston housing data set could not be improved significantly. Since the target variable is the *median* of house prices in a given area the presence of outliers also seems unlikely. Nevertheless a model comparison using marginal likelihood ratios indicates that the experiments provide some evidence in favour of the mixture model

compared to simple Gaussian noise. There are arguments suggesting that the noise is not Gaussian, but whether outliers in y are present is unclear. Note that using a *local* covariance function, for example the squared exponential (5), *GP* regression methods are inherently robust wrt. outliers in $\boldsymbol{x}$. Unfortunately we could not find reference problems for *non-linear* regression with outliers in the literature.

- In the proposed mixture noise *GP* model the marginal posterior distribution $p_{\text{post}}(\boldsymbol{f}|\mathcal{D})$ comes in the form of a mixture of Gaussians. The approximation by a single Gaussian as in EP can be poor if the posterior is highly multimodal, in which case MCMC sampling is also difficult. Intuitively, the posterior is highly multimodal if the model can explain the data in various distinct but equally plausible ways. For simpler outlier-structures this problem might be negligible, since the posterior can be expected to have a strongly dominant mode.

- The proposed mnEP scheme iterates between approximate inference over the latent function and ML-II estimation of the remaining parameters. Our mnEP implementation suffers from convergence problems in two ways. The first is that for given parameters EP might not converge. We observed that its convergence behaviour highly depends on the values of the parameters. Thus in case EP does not converge we make the gradient ascent search in other regions of the parameter space. The second problem is inherent to the ML-II parameter estimation, where our gradient ascent method can get caught in local maxima. Doing multiple runs of the algorithm on the same data sets the respective approximated marginal likelihood values, however, provide a reliable indication as to which solution to choose. Note that For large data sets $(N > 1000)$ one could explore sparse EP approximations to the posterior process following the lines of Csató and Opper (2002).

- MCMC sampling in the mixture noise *GP* model (mnMCMC) was the computationally most demanding method in the comparison. A conceptual advantage is that inference is performed over the function and the parameters $\boldsymbol{\theta}_{1,2}$ jointly. In mnMCMC problems related to multimodal posterior distributions can be alleviated by running several shorter chains from different initial states in favour of a single long chain, which in finite simulation can get stuck in a local mode. However, setting the parameters in the mnMCMC sampling scheme and inspecting the chains requires some experience.

In summary, the proposed noise mixture model is a practical way of applying *GP* regression in situations in which the potential existence of outliers in the data cannot be ruled out. We exemplified the use of the EP approximation and compared to MCMC sampling in *GP* regression models for non-standard likelihoods. This approach should encourage researchers to choose a noise model not for analytical convenience but to represent prior beliefs. In a practical problem the prior beliefs might be better reflected using different noise models, e.g. t-distributions (Lawrence and Tipping, 2003). Performing approximate inference follows similar schemes to those presented here.

Acknowledgements

MK, LC and CER acknowledge support for this project by the German Research Council (DFG) through grant RA 1030/1. We thank Dilan Görür and Jeremy Hill for helpful comments on the manuscript.

References

P. Abrahamsen. A review of Gaussian random fields and correlation functions. Technical Report 917, Norwegian Computing Center, Oslo, 1997.

C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

M. K. Cowles and B. P. Carlin. Markov chain Monte Carlo convergence diagnostics: A comparative review. *Journal of the American Statistical Association*, 91(434):883–904, 1996.

L. Csató and M. Opper. Sparse online Gaussian processes. *Neural Computation*, 14(2):641–669, 2002.

M. H. DeGroot and M. J. Schervish. *Probability and Statistics*. Addison-Wesley, 2002, third edition, 2002.

A. C. Faul and M. E. Tipping. A variational approach to robust regression. In *Proceedings of the International Conference on Artificial Neural Networks*, pages 95–102. Springer, 2001.

J. H. Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, 19(1):1–67, 1991.

M. N. Gibbs and D. J. C. MacKay. Variational Gaussian process classifiers. *IEEE Transactions on Neural Networks*, 11(6):1458–1464, 2000.

W. R. Gilks and S. Richardson, editors. *Markov Chain Monte Carlo in Practice*. Chapman & Hall, London, 1996.

D. Harrison and D. L. Rubinfeld. Hedonic housing prices and the demand for clean air. *Journal of Environmental Economics and Management*, 5(1):81–102, 1978.

P. J. Huber. *Robust Statistics*. John Wiley & Sons, New York, 1981.

E. T. Jaynes. *Probability Theory*. Cambridge University Press, Cambridge, 2003.

N. D. Lawrence and M. E. Tipping. A variational approach to robust Bayesian interpolation. In *Neural Networks for Signal Processing*, pages 229–238. IEEE, 2003.

D. J. C. MacKay. Bayesian interpolation. *Neural Computation*, 4(3):415–447, 1992.

D. J. C. MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003.

K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate Analysis*. Academic Press, London, 1979.

T. P. Minka. *Expectation Propagation for approximate Bayesian inference*. PhD thesis, Massachusetts Institute of Technology, 2001.

R. M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, Department of Computer Science, University of Toronto, 1993.

R. M. Neal. *Bayesian Learning for Neural Networks*. Springer, New York, 1996.

R. M. Neal. Monte Carlo implementation of Gaussian process models for Bayesian regression and classification. Technical Report 9702, Department of Statistics, University of Toronto, 1997.

R. M. Neal. Suppressing random walks in Markov chain Monte Carlo using ordered overrelaxation. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 205–230. Kluwer Academic Publishers, Dordecht, 1998.

A. O'Hagan. *Bayesian Inference*, volume 2B of *Kendall's Advanced Theory of Statistics*. Arnold, London, 1994.

M. Opper and O. Winther. Gaussian processes for classification: Mean-field algorithms. *Neural Computation*, 12(11):2655–2684, 2000.

P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection*. John Wiley & Sons, New York, 1987.

M. J. Schervish. *Theory of Statistics*. Springer, New York, 1997.

B. Schölkopf and A. J. Smola. *Learning with Kernels*. MIT Press, Cambridge, Massachusetts, 2002.

M. Seeger. *Bayesian Gaussian Process Models: PAC-Bayesian Generalisation Error Bounds and Sparse Approximations*. PhD thesis, University of Edinburgh, 2003.

M. O. Stitson, A. Gammerman, V. Vapnik, V. Vovk, C. Watkins, and J. Weston. Support vector regression with ANOVA decomposition kernels. In B. Schölkopf, J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*, chapter 17, pages 286–291. MIT Press, Cambridge, Massachusetts, 1999.

C. K. I. Williams. Prediction with Gaussian processes: From linear regression to linear prediction and beyond. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 599–621. Kluwer Academic Publishers, Dordecht, 1998.

C. K. I. Williams and D. Barber. Bayesian classification with Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1342–1351, 1998.

C. K. I. Williams and C. E. Rasmussen. Gaussian processes for regression. In David S. Touretzky, Michael C. Mozer, and Michael E. Hasselmo, editors, *Advances in Neural Information Processing Systems 8*, pages 514–520. MIT Press, 1996.