Robust Gaussian Process Regression with the Trimmed Marginal Likelihood

Abstract

Accurate outlier detection is not only a necessary preprocessing step, but can itself give important insights into the data. However, especially, for non-linear regression the detection of outliers is non-trivial, and actually ambiguous. We propose a new method that identifies outliers by finding a subset of data points T such that the marginal likelihood of all remaining data points S is maximized. Though the idea is more general, it is particular appealing for Gaussian processes regression, where the marginal likelihood has an analytic solution. While maximizing the marginal likelihood for hyper-parameter optimization is a well established non-convex optimization problem, optimizing the set of data points S is not. Indeed, even a greedy approximation for optimizing the set of data points S is computationally challenging due to the high cost of evaluating the marginal likelihood. As a remedy, we propose an efficient projected gradient descent method with provable convergence guarantees. Moreover, we also establish the breakdown point when jointly optimizing hyper-parameters and S. For various datasets and types of outliers, our experiments demonstrate that the proposed method can improve outlier detection and robustness when compared with several popular alternatives, like the student-t likelihood.

1 INTRODUCTION

Many real world data sets contain outliers, i.e. data points that are not representative of the majority of samples. For example, the output of a broken sensor might lead to an outlier observation. It is well known that estimating the parameters of a statistical model from data which contains outliers, can often lead to arbitrarily bad estimates, and therefore various robust learning techniques have been proposed [Rousseeuw and Leroy, 2005, Basu et al., 1998, Fujisawa and Eguchi, 2008].

Once the model has been robustly trained, we can detect outliers by ranking them according to the absolute value of the residuals, or remove some of the outliers in order to improve predictive performance. However, the success hinges on choosing the correct hyper-parameters for the robust training procedure.

Here in this work, we address the issue by proposing the use of the trimmed marginal likelihood. Let M be some probabilistic model, and denote by $p(\mathbf{y}_S|M)$ the marginal likelihood of data samples index by S. Let $\Omega = \{1, 2, \ldots, n\}$ denote the index set of all training samples. Given some trimming factor ν, we propose to find the set T, such that $p(\mathbf{y}_S|M)$ is maximized, with $S = \Omega \setminus T$, and subject to $|T| = \lfloor \nu n \rfloor$.

The trimmed marginal likelihood is particularly attractive for Gaussian process (GP) regression where the marginal likelihood has an analytic solution. In particular, we focus here on non-parametric regression model:

$$y = f(\mathbf{x}) + \epsilon\,,$$

where y and $\mathbf{x}$ are the response and covariates, respectively; f is sampled from a GP, and ϵ is some random noise, for example, $\epsilon \sim N(0, \sigma^2)$.

For GP regression, ν can be easily specified, since, as we prove in Section 3.1, ν corresponds to the breakdown point of our proposed method. In case where knowledge about the upper bound on the ratio of outliers is not available, we propose an iterative procedure for estimating ν (see Section 5).

However, the optimization over the set of data points S is NP hard and even a greedy approximation is computationally challenging. As a remedy, we propose an efficient projected gradient descent method with provable convergence guarantees (see Section 4.1).

Submitted to the 39th Conference on Uncertainty in Artificial Intelligence (UAI 2023). **To be used for reviewing only**.

Our experiments on various datasets and types of outliers demonstrate that the proposed method improves outlier detection and robustness when compared to several popular alternatives. Building on GPyTorch [Gardner et al., 2018], we also provide a computationally efficient implementation of our proposed method. (Source code will be made available after acceptance.)

2 RELATED WORK

Our proposed method is related to the trimmed likelihood approach for linear regression [Rousseeuw and Leroy, 2005, Rousseeuw and Van Driessen, 2006] (also known as trimmed least squares). Extending the trimmed likelihood approach beyond linear regression, was explored in Müller and Neykov [2003], though, they do not consider non-parametric models.

It is well known that the trimmed least squares method tends to underestimate the true variance, and therefore asymptotic correction factors [Rousseeuw and Leroy, 2005] and correction factors based on simulations [Pison et al., 2002] were previously proposed.

Another general approach for robust parameter estimation is to replace the Kullback-Leibler-divergence, underlying the maximum likelihood estimate, by the β or γ-distribution [Basu et al., 1998, Fujisawa and Eguchi, 2008]. This approach has also been extended to Bayesian inference in general [Nakagawa and Hashimoto, 2020, Futami et al., 2018], and Gaussian processes [Knoblauch et al., 2019] in particular. However, how to specify the hyper-parameters of these methods is less clear [Nakagawa and Hashimoto, 2020].

The most popular method for robust GP regression is to replace the Gaussian likelihood function by a student-t distribution [Jylänki et al., 2011]. However, the student-t distribution assumes that outliers are symmetric, i.e. an approximate even number of unusual large and small values. Furthermore, when combined with a GP prior, the marginal likelihood is not analytically tractable anymore.

Recently, also several other methods for robust GP regression have been proposed, which can roughly be categorized into likelihood robustification methods and residual-based methods.

Likelihood Robustification Methods The methods in [Daemi et al., 2019a,b] propose to use a mixture of two normal distributions for noise: one for modeling inliers and one for modeling outliers. [Lindfors et al., 2020] proposes to use a G-confluent distribution which generalizes the t-distribution, but still assumes symmetric outliers. In contrast, the work in [Alodat and Shakhatreh, 2020] and [Benavoli et al., 2021] propose to use the skew-normal distribution instead of the normal likelihood. However, all of the above methods make a particular assumption on the type of noise through the choice of the likelihood function.

Residual-based Methods The method in [Li et al., 2021] proposes to first train an ordinary GP regression model and then remove the data points with the largest residuals. Afterwards the GP regression model is trained again on the smaller set of data points, and the procedure of removing and retraining is repeated after a pre-defined number of steps. However, it is not difficult to see that their proposed method has a break down point of 1, meaning that one data can have an arbitrarily large impact on the posterior distribution. Consider one outlier with $y_{i_*} \to \infty$, then the residual to the outlier i_* will always be smaller than the residual of all other data points, which will lead to i_* being never removed. Similarly, [Ramirez-Padron et al., 2021] proposes to assign weights to each observation, based on the distance of the response to other neighboring data points. However, the method is sensitive to the choice of the neighborhood.
The method in [Park et al., 2021] introduces a bias vector $\boldsymbol{\delta} \in \mathbb{R}^n$, where n is the number of samples. If and only if $\delta_i \neq 0$, then sample i is considered an outlier. They propose to learn $\boldsymbol{\delta}$ using the ℓ_1-penalty. However, it can be shown that if there is only one outlier with $y_{i_*} \to \infty$, then $\forall i : \delta_i \neq 0$, meaning all samples are considered as outliers.

3 PROPOSED METHOD

Let $\Omega := \{1, ..., n\}$ denote the indices of all observations. Let M denote some probabilistic model (likelihood + prior), and $\log p(\mathbf{y}_S|M)$ the log-marginal likelihood of a given subset $S \subseteq \Omega$ of observations. For detecting a set of outliers $T \subseteq \Omega$, with $|T| = \lfloor \nu n \rfloor$, we propose to use the ν-trimmed marginal likelihood given as follows

$$\hat{S} := \underset{S \subseteq \Omega}{\arg\max} \log p(\mathbf{y}_S|M), \quad \text{subject to } |S| = \lceil (1-\nu)n \rceil ,$$

where $\hat{T} := \Omega \setminus \hat{S}$ is the set of potential outliers. This is a natural way to define the set of outliers and inliers, since the set $\hat{S}$ contains the samples that are best explained given model M.

In particular, for our model, we assume a zero mean GP process prior with covariance function k, and a Gaussian likelihood, that is

$$f \sim GP(0, k) ,$$
$$y \sim N(f(\mathbf{x}), \sigma^2) .$$

For our analysis and experiments we consider the scaled squared exponential covariance function, i.e.

$$k_{\eta,\mathbf{l}}(\mathbf{x}_{i_1}, \mathbf{x}_{i_2}) = \eta e^{-\sum_{j=1}^{d} \frac{1}{2l_j}(\mathbf{x}_{i_1}(j) - \mathbf{x}_{i_2}(j))^2} , \quad (1)$$

where η is the variance of the signal, and l_j are the length-scale parameters which control the change in correlation

when the data points differ in dimension j. We assume that $\mathbf{l} = (l_1, \ldots, l_d) \in \mathbb{D}$, where $\mathbb{D}$ is a compact subset of $\mathbb{R}_+^d$.[1] Furthermore, we assume $\sigma^2 \in \mathbb{R}_+$ and $\eta \in \mathbb{R}_+$.

Let $K_{\eta,\mathbf{l}} \in \mathbb{R}^{n \times n}$ denote the covariance matrix of all training data points, when using the covariance function from Equation (1). We assume that $K_{\eta,\mathbf{l}}$ is a positive definite matrix for all $\mathbf{l} \in \mathbb{D}$.

The log marginal likelihood $\log p(\mathbf{y}|X, \eta, \mathbf{l}, \sigma^2)$ is therefore given by

$$-\frac{1}{2}\mathbf{y}^T(K_{\eta,\mathbf{l}} + \sigma^2 I)^{-1}\mathbf{y} - \frac{1}{2}\log|K_{\eta,\mathbf{l}} + \sigma^2 I| - \frac{n}{2}\log 2\pi\,.$$

Since a fully Bayesian approach, i.e. integrating out the hyper-parameters $\boldsymbol{\theta} := (\eta, \mathbf{l}, \sigma^2)$, is computationally too expensive, we use empirical Bayes. For $S \subseteq \Omega$, let $(\mathbf{y}_S, X_S)$ denote the corresponding subset of the data. We define the v-trimmed marginal likelihood GP by

$$\underset{S,\boldsymbol{\theta}}{\text{maximize}}\ \log p(\mathbf{y}_S|X_S, \boldsymbol{\theta})\ \text{, subject to } |S| = \lceil(1-\nu)n\rceil\,. \tag{2}$$

3.1 ASYMPTOTICALLY CORRECT OUTLIER REJECTION

Similar in spirit to the definition of an outlier-prone model [O'Hagan, 1979], we define an outlier rejection method as asymptotically correct, if the set of observations with $y_i \to \infty$, or $y_i \to -\infty$ are detected as outliers.[2] The following proposition ensures asymptotically correctness.

Proposition 1. *Assume the covariance function from Equation* (1)*. Let V denote the true set of outliers, with $y_i \to \infty$, or $y_i \to -\infty$, for $i \in V$. Let U denote the true set of inliers, with y_i being bounded, for $i \in U$. Then, eventually (i.e. for $i \in V$, $|y_i|$ being large enough), we have*

$$S \subseteq U\,,$$

where S is the set of observations selected by the ν-trimmed marginal likelihood GP, with $\lfloor \nu n \rfloor \geq |V|$.

We defer the proof to the supplement material. Note that the proof were trivial, if the hyper-parameters $\boldsymbol{\theta}$ were fixed. However, since S and $\boldsymbol{\theta}$ are jointly optimized, a careful, non-trivial, proof is required.

Also note that Proposition 1 expresses that the ν-trimmed marginal likelihood GP has a breakdown point of ν, in the sense that ν is the minimal ratio of data points that need to be contaminated in order to lead to an arbitrary bad posterior.[3]

[1] We denote by $\mathbb{R}_+$ the set of positive reals which excludes 0 and ∞.

[2] The original definition of outlier-prone is only applicable to parametric models.

[3] For a more formal definition of the classical concept of breakdown point see [Rousseeuw and Leroy, 2005], which should be read by replacing "parameters" with "hyper-parameters".

4 OPTIMIZATION

Though conceptually easy, the ν-trimmed marginal likelihood GP, as given in Equation (2) is a computationally difficult optimization problem. Even if the hyper-parameters $\boldsymbol{\theta}$ were fixed, the remaining discrete optimization problem over $S \subseteq \Omega$ is still NP-hard.

In the following let $m := \lceil(1-\nu)n\rceil$. After initializing all hyper-parameters $\boldsymbol{\theta}$, we iterate between the optimization of $\boldsymbol{\theta}$ and S, as follows:

1. For fixed $\boldsymbol{\theta}$, find the set S that approximately maximizes the marginal likelihood, subject to the constraint $|S| = m$.
2. For fixed S, optimize $\boldsymbol{\theta}$ using one gradient descent step.

We repeat Step 1 and Step 2 till the marginal likelihood is not improved anymore. Step 2 is equal to the typical hyper-parameter optimization for GPs.

The complete algorithm is shown in Algorithm 1. When the step size $\xi^{(t)}$ is set small enough to ensure that $\ell^{(t)}$ decreases, Algorithm 1 is guaranteed to converge. In our implementation, we set step size $\xi^{(t)}$ and search direction $\Delta\boldsymbol{\theta}^{(t)}$ using Adam [Kingma and Ba, 2015] as Optimizer $\mathcal{O}$.

Algorithm 1: Trimmed-GP (Joint Optimization)

Input: $X, \mathbf{y}, \nu$
Output: set of inliers $S^{(t)}$, hyperparameters $\boldsymbol{\theta}^{(t)}$

$m := \lceil(1-\nu)n\rceil$
$t := 1;\ \ell^{(t)} := \infty$
initialize $\boldsymbol{\theta}^{(t)}$.
initialize optimizer $\mathcal{O}$ with global learning rate ξ_0.
repeat
  `// Step 1: Optimize S with PGD or Greedy`
  $S' := \underset{S\subseteq\Omega,\,|S|\,=\,m.}{\arg\min} \log p(\mathbf{y}_S|X_S, \boldsymbol{\theta}^{(t)})$
  if $\log p(\mathbf{y}_{S'}|X_{S'}, \boldsymbol{\theta}^{(t)}) > \log p(\mathbf{y}_{S^{(t)}}|X_{S^{(t)}}, \boldsymbol{\theta}^{(t)})$ **then**
    $S^{(t+1)} := S'$
    reset history of optimizer $\mathcal{O}$.
  else
    $S^{(t+1)} := S^{(t)}$
  end
  `// Step 2: increase` $\log p(\mathbf{y}_{S^{(t+1)}}|X_{S^{(t+1)}}, \boldsymbol{\theta})$ `by updating` $\boldsymbol{\theta}$
  find step size $\xi^{(t)}$ and direction $\Delta\boldsymbol{\theta}^{(t)}$ with $\mathcal{O}$.
  $\boldsymbol{\theta}^{(t+1)} := \boldsymbol{\theta}^{(t)} + \xi^{(t)}\Delta\boldsymbol{\theta}^{(t)}$
  $\ell^{(t+1)} := -\frac{1}{m}\log p(\mathbf{y}_{S^{(t+1)}}|X_{S^{(t+1)}}, \boldsymbol{\theta}^{(t+1)})$
  $t := t + 1$
until $\ell^{(t-1)} < \ell^{(t)}$

The optimization problem in Step 1 can be expressed as

follows. Find the set of samples $S \subseteq \{1, 2, \ldots, n\}$, with $|S| = m$, that maximize the marginal likelihood

$$-\frac{1}{2}\mathbf{y}_S^T(K_S + \sigma^2 I)^{-1}\mathbf{y}_S - \frac{1}{2}\log|K_S + \sigma^2 I| - \frac{m}{2}\log 2\pi\,, \tag{3}$$

where $K_S \in \mathbb{R}^{m\times m}$ is a sub-matrix of the positive-definite matrix $K \in \mathbb{R}^{n\times n}$, such that K_S contains the rows and columns of K indexed by S. Step 1 is challenging, since even a greedy search algorithm is computationally expensive due to the need for the repeated evaluation of the marginal likelihood.

4.1 PROJECTED GRADIENT DESCENT (PGD)

For finding a computationally feasible solution to Step 1, we proceed as follows. Assuming that the outliers are in the responses $\mathbf{y}$, and not in the covariates, we can ignore the term $\log|K_S + \sigma^2 I|$ in Equation (3). This reduces the problem to the maximization of

$$-\frac{1}{2}\mathbf{y}_S^T(K_S + \sigma^2 I)^{-1}\mathbf{y}_S\,.$$

Further, assuming that

$$(K_S + \sigma^2 I)^{-1} \approx [(K + \sigma^2 I)^{-1}]_S\,, \tag{4}$$

we arrive at the following optimization problem

$$\underset{S\subseteq\{1,2,\ldots,n\}}{\text{minimize}} \quad \mathbf{y}_S^T[(K + \sigma^2 I)^{-1}]_S\mathbf{y}_S\,, \qquad \text{subject to} \quad |S| = m\,. \tag{P1}$$

The resulting problem is still NP-hard (see e.g. [Chen et al., 2019]), but can be solved approximately using the following projected gradient method.

We can express (P1) as follows

$$\underset{\mathbf{b}}{\text{minimize}} \quad f(\mathbf{b})\,, \quad \text{subject to} \quad \|\mathbf{b}\|_0 = n - m\,, \tag{P2}$$

where we defined $f(\mathbf{b}) := (\mathbf{y} + \mathbf{b})^T(K + \sigma^2 I)^{-1}(\mathbf{y} + \mathbf{b})$, and $\|\cdot\|_0$ counts the number of non-zero entries. Denote by c a Lipschitz constant of $\nabla f(\mathbf{b})$, i.e.

$$\forall \mathbf{b}_1, \mathbf{b}_2 : \|\nabla f(\mathbf{b}_1) - \nabla f(\mathbf{b}_2)\|_2 \le c\|\mathbf{b}_1 - \mathbf{b}_2\|_2\,.$$

Here, the smallest Lipschitz constant of f is given by

$$\max_{\mathbf{x},\|\mathbf{x}\|_2=1} \|2(K + \sigma^2 I)^{-1}\mathbf{x}\|_2 = 2\frac{1}{\lambda_{\min}(K + \sigma^2 I)}\,.$$

A local minima can then be found by iterating

$$\mathbf{b}_{k+1} = \text{proj}_C\Big[\mathbf{b}_k - \frac{1}{c}\nabla f(\mathbf{b}_k)\Big]\,,$$

where

$$\nabla f(\mathbf{b}) = 2(K + \sigma^2 I)^{-1}(\mathbf{y} + \mathbf{b})\,,$$

and proj_C denotes the projection onto the set $C := \{\mathbf{x} \in \mathbb{R}^n \mid \|\mathbf{x}\|_0 \le n - m\}$, which is given by

$$\text{proj}_C\big[\mathbf{b}\big] = \underset{\mathbf{x},\|\mathbf{x}\|_0\le n-m}{\arg\min} \|\mathbf{b} - \mathbf{x}\|_2^2\,.$$

Note that, though the constraint $||\mathbf{b}||_0 = n - m$ is not convex, we can prove that the proposed projected gradient algorithm is guaranteed to converge to a stationary point:

Theorem 1. *Any sequence $\{\mathbf{b}_k\}$ generated by the projected gradient descent algorithm for Problem* (P2) *globally converges to a stationary point with locally linear convergence rate.*

The proof is in the supplement material. We note that since $(K + \sigma^2 I)^{-1}$ is fixed, each iteration involves only one matrix-vector multiplication which is in $O(m^2)$ and can be efficiently computed with GPUs.

4.2 GREEDY METHODS

Recall that our goal is to maximize the marginal likelihood (Equation (3)). However, the projected gradient descent method described in the previous section optimizes the simplified objective in (P1). Therefore, we also compare to a greedy method that directly optimizes Equation (3).

The greedy method starts with the index set of all data points $S := \{1, 2, \ldots, n\}$, and then removes the data point i_* which leads to the largest marginal likelihood, i.e.

$$i_* := \underset{i\in S}{\arg\max}\Big(\log p(\mathbf{y}_{S\setminus\{i\}}|X_{S\setminus\{i\}}, \boldsymbol{\theta})\Big)\,. \tag{5}$$

This is repeated until $|S| = \lceil(1 - \nu)n\rceil$. Naively solving the optimization in Equation (5) is in $O(n^4)$, since we need to repeat n-times the calculation of the determinant and inverse of $K_{S\setminus\{i\}}$, where $K_{S\setminus\{i\}}$ denotes the covariance matrix (plus $\sigma^2 I$) of the data points in $S \setminus \{i\}$. However, using the block matrix inversion lemma (together with the Woodbury formula) and the cofactor representation of the determinant, we can solve it in $O(n^3)$ (details in supplement material). Since the computation needs to be repeated $\lfloor\nu n\rfloor$ times, the greedy algorithm can still be too computationally expensive. Therefore, we also propose a batched version: first evaluate the leave-one-out (loo) estimate $\log p(\mathbf{y}_{S\setminus\{i\}}|X_{S\setminus\{i\}}, \boldsymbol{\theta})$ for all $i \in \{1, 2, \ldots, n\}$, and, second, remove at once the $\lfloor\nu n\rfloor$ samples with the highest loo estimate. We call the original greedy method Greedy (1-by-1), and the batched version Greedy (batch).

5 IMPROVED ν ESTIMATE

The upper bound ν on the ratio of the number of outliers might be too conservative, and as a consequence can lead to statistical inefficiency. Therefore, we propose the following procedure to improve upon the initial upper bound ν:

1. Using k-fold cross-validation, we estimate the residuals $\mathbf{r}$ of all data points.
2. Based on the residuals $\mathbf{r}$, we calculate a robust estimate of the noise variance σ^2.
3. We count the number of data points which residuals $\mathbf{r}$ are within two standard deviations σ, and use this number to get a new estimate for ν.

In Step 3, if the new estimate is smaller than the original ν, we repeat the above procedure. The details of the algorithm are show in Algorithm 2, where k denotes the number of folds, and $(train, test)$ denotes the training and test indices of one fold. For our experiments we use $k = 10$, i.e. 10-fold cross-validation. Furthermore, note that within cross-validation, we use ν_* (defined in line 5) instead of $\nu^{(t)}$ due to a possibly uneven split of outliers in $(train, test)$. In line 7, $\mathbb{E}[\hat{\mathbf{y}}_{test}|X_{test}, X_S, \mathbf{y}_S, \boldsymbol{\theta}]$ denotes the predicted mean response at data points X_{test} using the GP with training data points $(X_S, \mathbf{y}_S)$ and covariance function hyperparameters $\boldsymbol{\theta}$. Note that in line 9, $r^2_{(w)}$ denotes the w-th smallest squared residual, and $Q_{\chi^2(1)}$ denotes the quantile function for the χ^2 distribution with 1 degree of freedom. The robust variance estimator, in line 9, is a generalization of the estimator proposed in [Rousseeuw, 1984] and is explained in more detail in the supplement material.

Algorithm 2: Improved ν estimate

Input: $X, \mathbf{y}, \nu$
Output: new upper bound on outlier ratio $\nu^{(t)}$

1 $t := 1$
2 $\nu^{(t)} := \nu$ `// set to initial estimate of number of outliers`
3 **repeat**
4 **for** *(train, test)* **in** k-*Fold*(n) **do**
 `// use` ν_* `instead of` $\nu^{(t)}$ `due to possibly uneven split of outliers`
5 $\nu_* := \nu^{(t)}/(1 - \frac{1}{k})$
6 $S, \boldsymbol{\theta}$ = Trimmed-GP$(X_{train}, \mathbf{y}_{train}, \nu_*)$
 `// residuals at test points`
7 $\mathbf{r}_{test} := \mathbf{y}_{test} - \mathbb{E}[\hat{\mathbf{y}}_{test}|X_{test}, X_S, \mathbf{y}_S, \boldsymbol{\theta}]$
8 **end**
9 $\sigma^2 := r^2_{(\lfloor(1-\nu^{(t)})n\rfloor)}/Q_{\chi^2(1)}(1 - \nu^{(t)})$
10 $\nu^{(t+1)} := \#(|\mathbf{r}| > 2\sigma)/n$ `// count samples not within two std`
11 $t := t + 1$
12 **until** $\nu^{(t)} \geq \nu^{(t-1)}$

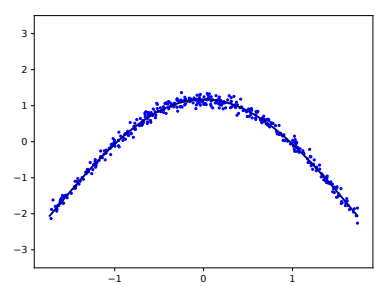

Figure 1: Blue dots show the samples from the synthetic bow-shaped data ($\sigma^2 = 0.01$). Black line shows true function.

6 EXPERIMENTS

In this section, we evaluate the proposed method and several baselines on the task of correctly identifying outliers and predictive performance.

Baselines and Implementations We compare to a GP with student-t likelihood for estimating $\mathbb{E}[y|\mathbf{x}]$, denoted as t-GP. Note that the student-t likelihood does not explicitly distinguish between inliers and outliers. Therefore it is essentially a noise model for both the inliers and outliers. We also compare our method to a standard GP trained by minimizing the KL-divergence (GP), and one trained by minimizing the γ-divergence (γ-GP). All hyper-parameters are estimated with empirical Bayes using the complete data set $(X, \mathbf{y})$.

All methods were implemented using GPyTorch [Gardner et al., 2018], and the full dataset (no inducing points) was used.[4] For the proposed method (ν-GP) we set $\nu = 0.5$ and use Algorithm 2. Note that 0.5 is also the breakdown point of the student-t distribution. In Algorithm 1 (line 6) we use the proposed projected gradient descent (PGD) method (if not mentioned otherwise).

Synthetic Datasets For illustration of the differences between each method, we created a simple one-dimensional bow-shaped data, shown in Figure 1 with $n = 400$ (bow). Furthermore, we use the Friedman data set as in [Friedman, 1991, Naish-Guzman and Holden, 2007] with $d = 10$, and $n = 100$ (F100), and $n = 400$ (F400).

Real Datasets We also evaluated all methods on three commonly used regression datasets: bodyfat ($d = 14, n = 252$), housing ($d = 13, n = 506$) and spacega ($d = 6, n =$

[4] We will make our implementation available.

3107) that are available from the LIBSVM archive.[5]

Outlier Types A random subset of data points is replaced by the following three types of outliers.

- **uniform** The position (=covariates) of the outliers is unchanged, but the response is changed by randomly adding or subtracting a value which is uniformly drawn between 3 and 9 standard deviations of the original response.
- **focused** The position of the outliers is the median of each dimension plus some jitter. The response is set to the original response minus 3 times the standard deviation of the original response plus some jitter.
- **asymmetric** Same as *uniform*, but the responses, corresponding to the outliers, are changed by either always adding or always subtracting a uniformly drawn positive number.

In all cases, we change 10% of the existing data points to outliers. For all experiments we report the average over 10 times randomly adding outliers (and standard deviation in brackets).

6.1 RESULTS

The results for the bow-shaped data are shown in Figure 2. First, looking at the results for uniform outliers, we observe that all methods approximately infer the true underlying function, while only the standard GP shows a few deviations. As a consequence, all methods correctly identify all outliers. However, for the focused outliers, the situation is quite different: t-GP and γ-GP assume that the focused outliers are part of the true function, and the top of the bow are the outliers, while our proposed method ν-GP infers the opposite. Both results are plausible, and show that ν-GP can detect different types of outliers than the popular t-GP. Finally, for the asymmetric outliers all robust GP methods are able to infer the correct function, while only the standard GP is influenced by the outliers.

While we used the bow-shaped data to show the qualitative differences between the GP methods, we next evaluate all methods also quantitatively on the more challenging datasets F100, F400, and the three real datasets (bodyfat, housing, spacega). We investigate each method's performance in terms of ranking the set of outliers correctly using the residuals. Since we know the total number of outliers, we use R-precision for evaluation.[6] Let r be the total number of outliers, then R-precision is defined as the number of true outliers within the top-r largest residuals divided by r.

Table 1: Evaluation of all methods in terms of outlier ranking performance (R-precision).

	GP	**γ-GP**	**t-GP**	**ν-GP**
uniform outliers				
bow	**1.0** (0.0)	**1.0** (0.0)	**1.0** (0.0)	**1.0** (0.0)
F100	0.92 (0.17)	0.86 (0.22)	**1.0** (0.0)	**1.0** (0.0)
F400	**1.0** (0.0)	0.97 (0.02)	0.98 (0.02)	**1.0** (0.0)
body	0.84 (0.05)	**0.86** (0.05)	**0.86** (0.05)	**0.86** (0.06)
house	**0.85** (0.06)	0.84 (0.04)	0.84 (0.04)	**0.85** (0.06)
spacega	**0.99** (0.0)	0.87 (0.07)	0.95 (0.01)	0.98 (0.0)
focused outliers				
bow	0.57 (0.06)	0.18 (0.1)	0.2 (0.06)	**0.97** (0.1)
F100	0.59 (0.16)	0.44 (0.21)	0.39 (0.14)	**0.72** (0.43)
F400	0.41 (0.08)	0.47 (0.17)	0.64 (0.34)	**1.0** (0.0)
body	0.54 (0.13)	0.54 (0.09)	0.56 (0.18)	**0.78** (0.24)
house	0.34 (0.26)	0.46 (0.12)	0.46 (0.13)	**0.64** (0.28)
spacega	0.23 (0.02)	0.18 (0.02)	0.17 (0.01)	**0.97** (0.01)
asymmetric outliers				
bow	**1.0** (0.0)	**1.0** (0.0)	**1.0** (0.0)	**1.0** (0.0)
F100	0.8 (0.17)	0.75 (0.21)	**1.0** (0.0)	**1.0** (0.0)
F400	0.96 (0.03)	0.95 (0.03)	0.95 (0.02)	**1.0** (0.0)
body	0.81 (0.07)	**0.86** (0.05)	**0.86** (0.05)	**0.86** (0.06)
house	0.82 (0.05)	0.85 (0.04)	0.85 (0.03)	**0.87** (0.05)
spacega	0.96 (0.01)	0.85 (0.02)	0.94 (0.01)	**0.98** (0.0)

The results, summarized in Table 1, show that the proposed method is better in identifying outliers than other robust GP methods. As can be seen in Table 2 this also leads to better prediction performance at test time. Notably, for all real datasets we achieve considerable improvements in root mean squared error (RMSE) when compared to other robust GP methods. In terms of runtime, our proposed method is slower, but still in the same order as other robust methods for the largest dataset (details in supplement material).

7 ANALYSIS

Here we investigate several aspects of the proposed ν-GP.

7.1 ESTIMATION OF ν

The values of ν estimated with Algorithm 2 were around 2% for the datasets without added outliers, and around 8% for the datasets with added outliers. That means, the estimated ν were considerably smaller than the initial value of 50%, but slightly lower than the true ratio of outliers (which is 10%). This might be because, some of the outliers do not conflict with the smoothness properties of the covariance function and thus cannot be distinguished from inliers.

[5] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/regression.html

[6] At least for the synthetic data, bow, F100 and F400; for the real data the true number of outliers is unknown, but assumed to be at least the number of extra added outliers.

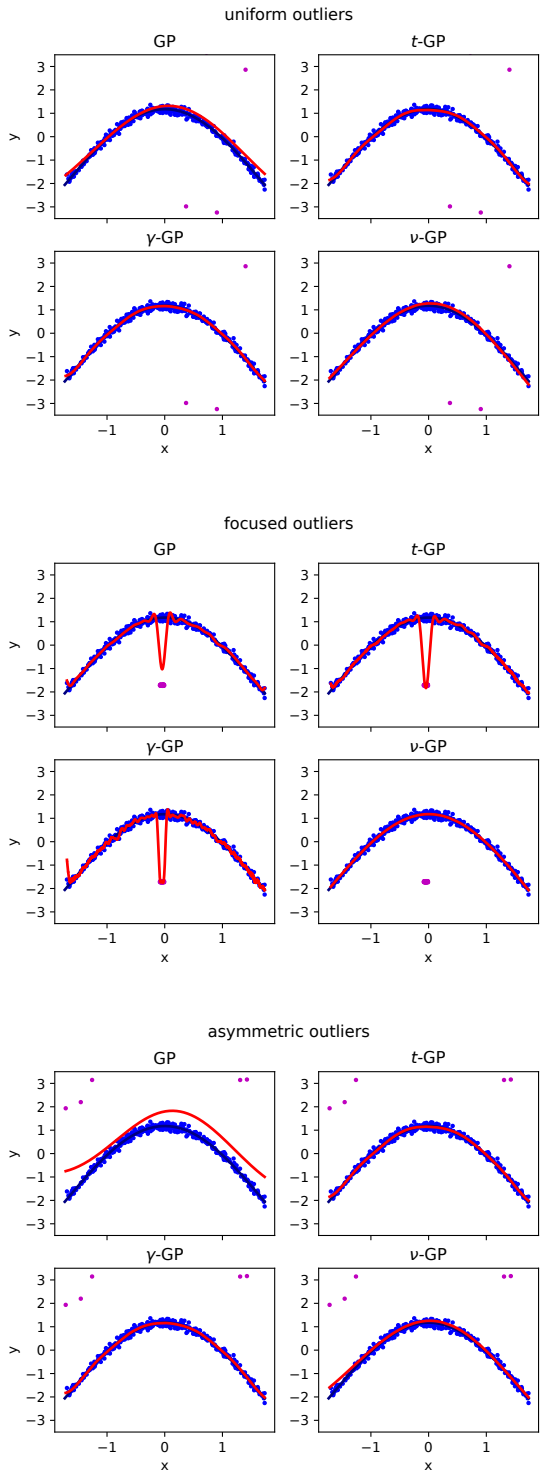

Figure 2: Shows uniform (top), focused (middle) and asymmetric (bottom) outliers for the synthetic bow-shaped data. Note that here focused outliers (middle) are at around position (0, -2). Red shows the predicted function of each method. Pink and blue dots are the true outliers and inliers, respectively. Black line shows true function.

Table 2: Root mean squared error (RMSE) of predictions on test data.

	GP	**γ-GP**	**t-GP**	**ν-GP**
no extra added outliers				
bow	**0.06** (0.0)	**0.06** (0.0)	**0.06** (0.0)	**0.06** (0.0)
F100	0.23 (0.04)	0.25 (0.05)	**0.22** (0.04)	0.31 (0.07)
F400	**0.15** (0.01)	0.61 (0.2)	0.61 (0.19)	0.27 (0.01)
body	0.11 (0.09)	0.22 (0.11)	0.56 (0.23)	**0.06** (0.08)
house	**0.35** (0.07)	0.83 (0.39)	0.99 (0.29)	0.48 (0.13)
spacega	0.41 (0.03)	0.48 (0.04)	0.49 (0.03)	**0.39** (0.02)
uniform outliers				
bow	0.12 (0.04)	**0.06** (0.0)	**0.06** (0.0)	**0.06** (0.0)
F100	0.66 (0.18)	0.47 (0.25)	**0.29** (0.1)	0.32 (0.06)
F400	0.38 (0.05)	0.64 (0.05)	0.64 (0.05)	**0.26** (0.02)
body	0.27 (0.15)	0.57 (0.1)	0.58 (0.08)	**0.1** (0.06)
house	0.65 (0.22)	0.85 (0.15)	0.86 (0.14)	**0.38** (0.11)
spacega	**0.4** (0.02)	0.68 (0.05)	0.53 (0.04)	0.41 (0.02)
focused outliers				
bow	0.2 (0.01)	0.26 (0.03)	0.27 (0.03)	**0.07** (0.07)
F100	0.44 (0.05)	0.46 (0.05)	0.44 (0.05)	**0.28** (0.05)
F400	0.3 (0.04)	0.4 (0.14)	0.46 (0.15)	**0.2** (0.05)
body	0.41 (0.08)	0.5 (0.06)	0.46 (0.08)	**0.1** (0.09)
house	**0.34** (0.05)	0.44 (0.11)	0.51 (0.12)	0.37 (0.12)
spacega	0.44 (0.09)	0.51 (0.09)	0.51 (0.09)	**0.41** (0.06)
asymmetric outliers				
bow	0.34 (0.04)	**0.06** (0.01)	**0.06** (0.0)	0.07 (0.01)
F100	0.74 (0.13)	0.61 (0.2)	**0.23** (0.02)	0.34 (0.02)
F400	0.54 (0.04)	0.57 (0.14)	0.63 (0.04)	**0.26** (0.02)
body	0.42 (0.05)	0.57 (0.15)	0.64 (0.06)	**0.16** (0.08)
house	0.65 (0.07)	0.76 (0.17)	0.81 (0.15)	**0.31** (0.08)
spacega	0.56 (0.02)	0.73 (0.05)	0.55 (0.02)	**0.42** (0.02)

Table 3: Average marginal likelihood of solution found by different optimization methods.

	PGD	Greedy (batch)	Greedy (1-by-1)
bow	**1.73** (0.09)	1.59 (0.13)	**1.73** (0.09)
F100	0.09 (0.15)	-0.09 (0.22)	**0.12** (0.31)
F400	0.21 (0.14)	0.11 (0.22)	**0.27** (0.19)
body	**0.39** (2.56)	0.13 (2.33)	0.37 (2.53)
house	-0.71 (1.2)	-0.77 (1.17)	**-0.67** (1.23)
spacega	-0.27 (0.03)	**0.08** (0.2)	-

Table 4: Average outlier ranking performance (R-precision) of different optimization methods.

	PGD	Greedy (batch)	Greedy (1-by-1)
bow	**1.0** (0.0)	**1.0** (0.0)	**1.0** (0.0)
F100	**1.0** (0.0)	**1.0** (0.02)	**1.0** (0.0)
F400	**1.0** (0.0)	**1.0** (0.0)	**1.0** (0.0)
body	**0.91** (0.08)	0.89 (0.09)	0.9 (0.08)
house	**0.87** (0.11)	0.75 (0.21)	0.81 (0.2)
spacega	**0.97** (0.0)	0.76 (0.36)	-

Table 5: Average root mean squared error (RMSE) on test data of different optimization methods.

	PGD	Greedy (batch)	Greedy (1-by-1)
bow	**0.05** (0.01)	0.06 (0.01)	**0.05** (0.01)
F100	**0.29** (0.06)	0.32 (0.1)	0.32 (0.14)
F400	0.25 (0.03)	**0.24** (0.04)	**0.24** (0.04)
body	**0.08** (0.1)	0.09 (0.1)	**0.08** (0.09)
house	0.42 (0.14)	**0.38** (0.11)	0.42 (0.14)
spacega	0.43 (0.05)	**0.38** (0.04)	-

Table 6: Average runtime in minutes of each optimization method.

	PGD	Greedy (batch)	Greedy (1-by-1)
bow	**0.15** (0.05)	4.61 (4.9)	137.21 (55.65)
F100	**0.13** (0.15)	3.64 (5.38)	6.53 (4.64)
F400	**0.13** (0.04)	4.64 (6.13)	99.66 (64.59)
body	**0.72** (0.67)	3.44 (3.08)	42.06 (42.27)
house	**0.34** (0.65)	4.38 (4.76)	132.08 (138.23)
spacega	**0.71** (0.17)	12.47 (6.59)	-

7.2 MARGINAL LIKELIHOOD OPTIMIZATION

In order to optimize the marginal likelihood in Equation 2, we proposed Algorithm 1 either with a projected gradient descent (PGD) method (Section 4.1) or a greedy method (Section 4.2). Here, we compare the solutions of these different optimization methods with respect to the marginal likelihood, outlier detection, and prediction on test data.[7] For conciseness, we report here the average results over all outlier types, detailed results for each outlier type (no, uniform, focused, asymmetric) can be found in the supplement material. As we can see in Table 3, expect for spacega, the PGD method often provides better solutions to the combinatorial optimization problem than greedy (batch), but worse than greedy (1-by-1). However, as can be seen in Table 6, the runtime of PGD is considerably faster than all greedy methods, and, for the larger dataset spacega, Greedy (1-by-1) was actually infeasible. Comparing Table 3 and 4, we see that in most cases better marginal likelihood translates into better outlier detection. However, the relation between the marginal likelihood and prediction on test data, Table 5, is slightly mixed - a result that is in line with recent discussions about the optimization of the marginal likelihood for improving test performance [Lotfi et al., 2022].

8 CONCLUSIONS

The ν-trimmed marginal likelihood (ν-GP) approach is a natural extension of the empirical Bayes framework to robust Gaussian Process (GP) regression. While for GP regression it is common to optimize the covariance function parameters by maximizing the marginal likelihood, here, we additionally proposed to optimize (= select) the subset of data points that maximize the marginal likelihood. We showed that the trimming ratio ν is an intuitive hyper-parameter since it corresponds to an upper bound on the outlier ratio and has the theoretic guarantee of controlling the breakdown point. Note that this is in contrast to the hyper-parameters of commonly used robust methods like the student-t likelihood and the γ-divergence, which are difficult to interpret. In case where prior knowledge about an upper bound on the outlier ratio is unknown, we proposed to iteratively refine a conservative estimate of $\nu = 0.5$, which is the same break-down point as the student-t likelihood.

In practice, the success of ν-GP hinges on an efficient method for optimizing the subset of inliers. For that purpose, we proposed a projected gradient descent (PGD) method, proved its theoretic convergence guarantees, and showed empirically that the quality of the optimization is at par with greedy methods, while being computationally much more efficient. Finally, the resulting ν-GP with PGD compared favorable against common robust GP methods in terms of outlier detection and test prediction.

[7] Due to the long runtime of the greedy methods, here, we fix ν to 0.2 for all methods.

References

MT Alodat and Mohammed K Shakhatreh. Gaussian process regression with skewed errors. *Journal of Computational and Applied Mathematics*, 370:112665, 2020.

Hedy Attouch, Jérôme Bolte, and Benar Fux Svaiter. Con-

vergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized gauss–seidel methods. *Mathematical Programming*, 137(1):91–129, 2013.

Ayanendranath Basu, Ian R Harris, Nils L Hjort, and MC Jones. Robust and efficient estimation by minimising a density power divergence. *Biometrika*, 85(3):549–559, 1998.

Alessio Benavoli, Dario Azzimonti, and Dario Piga. A unified framework for closed-form nonparametric regression, classification, preference and mixed problems with skew gaussian processes. *Machine Learning*, pages 1–39, 2021.

Yichen Chen, Yinyu Ye, and Mengdi Wang. Approximation hardness for a class of sparse optimization problems. *Journal of Machine Learning Research*, 2019.

Atefeh Daemi, Yousef Alipouri, and Biao Huang. Identification of robust gaussian process regression with noisy input using em algorithm. *Chemometrics and Intelligent Laboratory Systems*, 191:1–11, 2019a.

Atefeh Daemi, Hariprasad Kodamana, and Biao Huang. Gaussian process modelling with gaussian mixture likelihood. *Journal of Process Control*, 81:209–220, 2019b.

Jerome H Friedman. Multivariate adaptive regression splines. *The annals of statistics*, 19(1):1–67, 1991.

Hironori Fujisawa and Shinto Eguchi. Robust parameter estimation with a small bias against heavy contamination. *Journal of Multivariate Analysis*, 99(9):2053–2081, 2008.

Futoshi Futami, Issei Sato, and Masashi Sugiyama. Variational inference based on robust divergences. In *International Conference on Artificial Intelligence and Statistics*, pages 813–822. PMLR, 2018.

Jacob Gardner, Geoff Pleiss, Kilian Q Weinberger, David Bindel, and Andrew G Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. *Advances in neural information processing systems*, 31, 2018.

Pasi Jylänki, Jarno Vanhatalo, and Aki Vehtari. Robust gaussian process regression with a student-t likelihood. *Journal of Machine Learning Research*, 12(11), 2011.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Jeremias Knoblauch, Jack Jewson, and Theodoros Damoulas. Generalized variational inference: Three arguments for deriving new posteriors. *arXiv preprint arXiv:1904.02063*, 2019.

Guoyin Li and Ting Kei Pong. Calculus of the exponent of kurdyka–łojasiewicz inequality and its applications to linear convergence of first-order methods. *Foundations of computational mathematics*, 18(5):1199–1232, 2018.

Z-Z Li, Lu Li, and Zhengyi Shao. Robust gaussian process regression based on iterative trimming. *Astronomy and Computing*, page 100483, 2021.

Martin Lindfors, Tianshi Chen, and Christian A Naesseth. Robust gaussian process regression with g-confluent likelihood. *IFAC-PapersOnLine*, 53(2):394–399, 2020.

Sanae Lotfi, Pavel Izmailov, Gregory Benton, Micah Goldblum, and Andrew Gordon Wilson. Bayesian model selection, the marginal likelihood, and generalization. In *International Conference on Machine Learning*, pages 14223–14247. PMLR, 2022.

Christine H Müller and Neyko Neykov. Breakdown points of trimmed likelihood estimators and related estimators in generalized linear models. *Journal of Statistical Planning and inference*, 116(2):503–519, 2003.

Andrew Naish-Guzman and Sean Holden. Robust regression with twinned gaussian processes. *Advances in neural information processing systems*, 20:1065–1072, 2007.

Tomoyuki Nakagawa and Shintaro Hashimoto. Robust bayesian inference via γ-divergence. *Communications in Statistics-Theory and Methods*, 49(2):343–360, 2020.

Anthony O'Hagan. On outlier rejection phenomena in bayes inference. *Journal of the Royal Statistical Society: Series B (Methodological)*, 41(3):358–367, 1979.

Chiwoo Park, David J Borth, Nicholas S Wilson, Chad N Hunter, and Fritz J Friedersdorf. Robust gaussian process regression with a bias model. *Pattern Recognition*, page 108444, 2021.

Greet Pison, Stefan Van Aelst, and G Willems. Small sample corrections for lts and mcd. *Metrika*, 55(1-2):111–123, 2002.

Ruben Ramirez-Padron, Boris Mederos, and Avelino J Gonzalez. Robust weighted gaussian processes. *Computational Statistics*, 36(1):347–373, 2021.

Kenneth Rice and David Spiegelhalter. A simple diagnostic plot connecting robust estimation, outlier detection, and false discovery rates. *Journal of applied statistics*, 33(10): 1131–1147, 2006.

Peter J Rousseeuw. Least median of squares regression. *Journal of the American Statistical Association*, 79(388): 871–880, 1984.

Peter J Rousseeuw and Annick M Leroy. *Robust regression and outlier detection*, volume 589. John wiley & sons, 2005.

Peter J Rousseeuw and Katrien Van Driessen. Computing lts regression for large data sets. *Data mining and knowledge discovery*, 12:29–45, 2006.