Neurocomputing 452 (2021) 224–238

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Mixture of robust Gaussian processes and its hard-cut EM algorithm with variational bounding approximation

Tao Li [a], Di Wu [b], Jinwen Ma [a,*]

[a] Department of Information Science, School of Mathematical Sciences and LMAM, Peking University, Beijing 100871, China
[b] School of Computer Science, Shanxi Normal University, Xi'an 710100, China

ARTICLE INFO

Article history:
Received 26 April 2020
Revised 16 February 2021
Accepted 22 April 2021
Available online 30 April 2021
Communicated by Zidong Wang

Keywords:
Gaussian processes
EM algorithm
Variational inference
Robust regression
Multi-modal data

ABSTRACT

The Gaussian process is a powerful statistical learning model and has been applied widely in nonlinear regression and classification. However, it fails to model multi-modal data from a non-stationary source since a prior Gaussian process is generally stationary. Based on the idea of the mixture of experts, the mixture of Gaussian processes was established to increase the model flexibility. On the other hand, the Gaussian process is also sensitive to outliers and thus robust Gaussian processes have been suggested to own the heavy-tailed property. In practical applications, the datasets may be multi-modal and contain outliers at the same time. In order to overcome these two difficulties together, we propose a mixture of robust Gaussian processes (MRGP) model and establish a precise hard-cut EM algorithm for learning its parameters. Since the exact solving process is intractable due to the fact that non-Gaussian probability density functions of the noises are adopted into the likelihood of the proposed model on the dataset, we employ a variational bounding method to approximate the marginal likelihood functions so that the hard-cut EM algorithm can be implemented effectively. Moreover, we conduct various experiments on both synthetic and real-world datasets to evaluate and compare our proposed MRGP method with several competitive nonlinear regression methods. The experimental results demonstrate that our MRGP model with the hard-cut EM algorithm is much more effective and robust than the competitive nonlinear regression models.

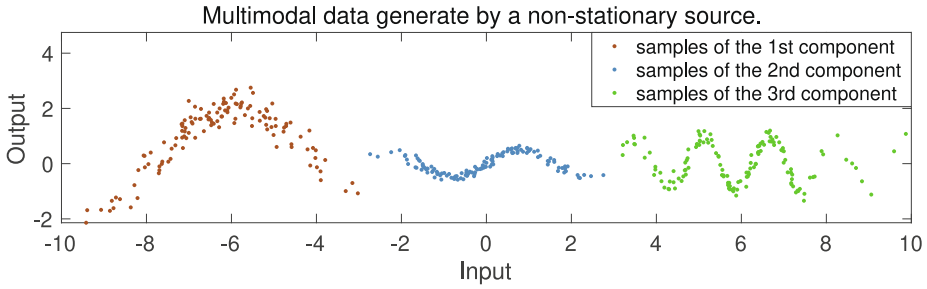
1. Introduction

Gaussian Process (GP) [1] is one of the most popular models in machine learning, pattern recognition, and time series prediction. In fact, it is a non-parametric model and we can use it to infer over uncertain time series or functions. However, the Gaussian processes used for machine learning suffer from two severe problems. Firstly, a prior Gaussian process with zero mean function and commonly used covariance functions is generally stationary [1], which strongly limits the fitting flexibility. So, it is unreasonable to use a single Gaussian process to model the data generated from a non-stationary source, which is very common in practice. For example, in Fig. 1(a) we generate a multimodal dataset and we fit this dataset with a single Gaussian process. The fitted curve is shown in Fig. 1(b). In the first component, the fitted curve oscillates frequently and fails to capture the smooth characteristics of the underlying function. Secondly, Gaussian processes are not robust. In fact, the trained models are sensitive to outliers in the training set, as illustrated in Fig. 2. If there exists no outlier, the trained Gaussian process can fit the data along with the function $\sin(x)$ very well, but if there exits only a single outlier, the fitting of the trained Gaussian process degenerates remarkably.

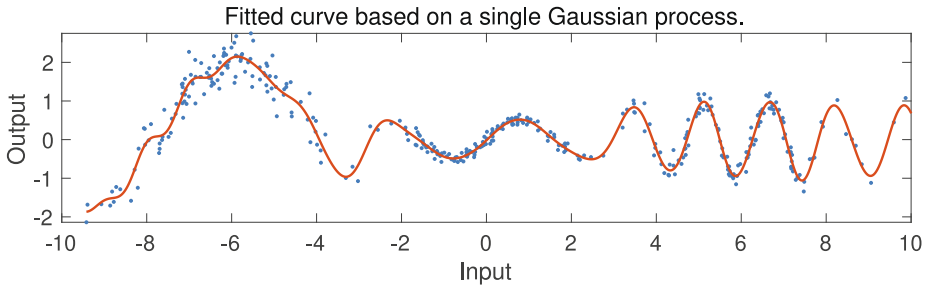
To tackle these two problems, many strategies based on conventional Gaussian processes have been suggested. One effective way to overcome the first problem is to adopt the mixture structure in Gaussian processes. In fact, based on the idea of mixture of experts [2,3], various Mixture of Gaussian Processes (MGP) models [4–6] have been proposed to improve the model flexibility for non-stationary sources. In Fig. 1(c), we also show the fitted curve based on the MGP model. By comparing Fig. 1(b) and (c), we can see that introducing mixture structure into Gaussian process models significantly improve the model flexibility and capacity. However, the parameter learning for an MGP model on a given dataset is very challenging because the samples are not independent, but correlated. To get rid of this difficulty, different kinks of approximate leaning algorithms have been established for the MGP models. We review these MGP models as well as the learning algorithms in Section 2.1.

* Corresponding author.
E-mail addresses: li_tao@pku.edu.cn (T. Li), wudi2@snnu.edu.cn (D. Wu), jwma@math.pku.edu.cn (J. Ma).

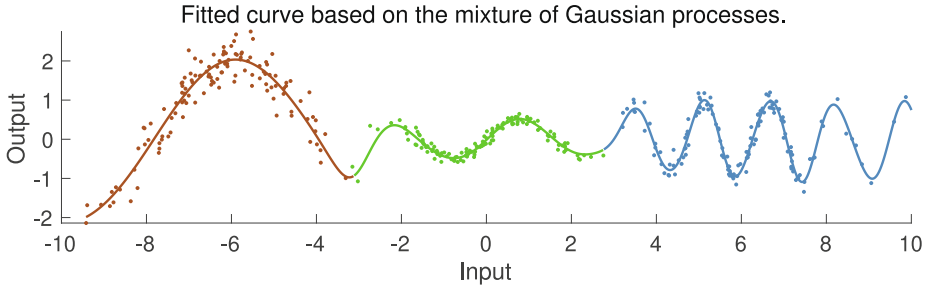
https://doi.org/10.1016/j.neucom.2021.04.085
0925-2312/



(a) A synthetic multimodal dataset.



(b) Fitted curve based on a single Gaussian process.



(c) Fitted curve based on the mixture of Gaussian processes.

Fig. 1. An example of multimodal/non-stationary data fitting task. The data is generated by $f(x) = 2\sin(0.8x)\mathbb{I}(x \leqslant -3) + 0.5\sin(2x)\mathbb{I}(-3 < x < 3) + \sin(4x)\mathbb{I}(x \geqslant 3)$, and we the generate 400 samples. The inputs x obeys a mixture of Gaussian distributions: $\frac{1}{3}\mathcal{N}(-6,\sqrt{3}) + \frac{1}{3}\mathcal{N}(0,\sqrt{3}) + \frac{1}{3}\mathcal{N}(6,\sqrt{3})$.

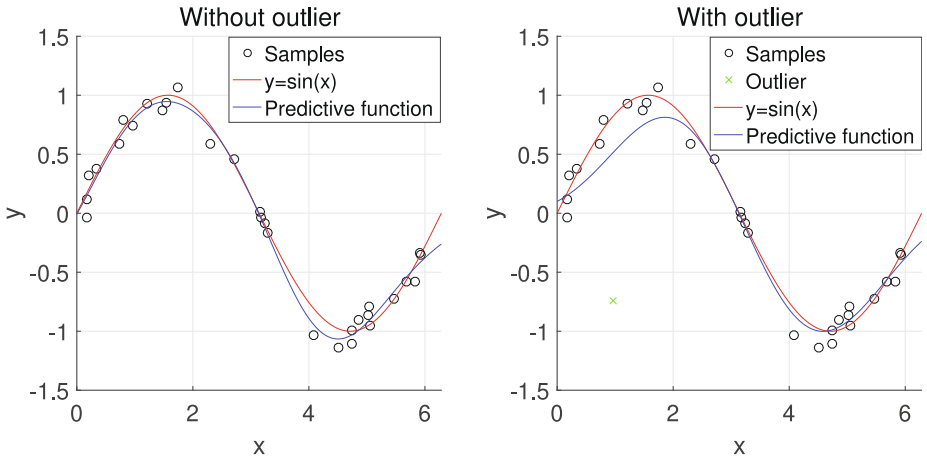


Fig. 2. Gaussian processes are sensitive to outliers in the training set. **Left**: Gaussian process successfully learn $\sin(x)$ very well from the data when there is no outlier. **Right**: The learned function is remarkably deviated from $\sin(x)$ with the influence of a single outlier.

On the other hand, the Robust Gaussian processes (RGPs) [7–9] have been suggested to overcome the second problem. In conventional Gaussian processes, the noises are assumed to be also Gaussian. Under this assumption, the latent function can be integrated out analytically, and the marginal likelihood can be calculated explicitly. However, Gaussian distribution is not heavy-tailed, which means it is sensitive to outliers. In [8,9], student-t distribution or Laplace distribution is utilized to model the noise so that

the modified Gaussian process becomes robust. However, this robust Gaussian process becomes much more complicated than the conventional Gaussian process because its exact solving process of the parameters via Maximum likelihood (ML) is intractable. Therefore, certain approximate mechanisms should be adopted into the ML learning of the parameters such as Expectation Propagation (EP) [10], Laplace approximation [11], Variational Bayesian (VB) [12] and so on.

In practical applications, it is common that the data source is non-stationary, while there exist outliers at the same time, due to technical reasons or some factors beyond our control. Thus it is vital to develop a regression model that is not only able to model non-stationary data but also robust to outliers. In this paper, by combining the ideas of MGP and RGP together, we propose a novel model: Mixture of Robust Gaussian Processes (MRGP), which inherits the advantages of both MGP and RGP to solve the multi-model data and outlier-sensitive problems. Moreover, we design a hard-cut EM algorithm with variational bounding approximation for the parameter learning of MRGP. It is demonstrated by the experimental results on both synthetic and real-world datasets that our proposed MRGP model with the hard-cut EM algorithm is much more effective and robust than the competitive nonlinear regression models.

The contributions of this paper are summarized as follows:

- We establish a model that is both robust to outliers and able to model non-stationary data, which overcomes the problems of traditional Gaussian processes simultaneously.
- We develop an effective learning algorithm for the proposed model based on the variational bounding technique.
- We conduct extensive experiments on various datasets to compare common non-linear regression techniques in the presence of outliers.

The rest of this paper is organized as follows. In Section 2, we review related works on MGP, approximate inference in the training of GP, as well as the robust modeling. Then we introduce the GP and MGP models in Section 3. The MRGP model and its hard-cut EM algorithm are presented in Section 4. We summarize the experimental results on both synthetic and real-world datasets in Section 5. Finally, we conclude this paper in Section 6.

2. Related works

2.1. Mixture of Gaussian processes and learning algorithms

The mixture of experts (ME) was originally introduced in [2] with the idea that the final prediction result is a weighted summation of the prediction results obtained by certain local experts, and their weights are calculated by a gating function adaptively. Actually, this kind of ME architecture is discriminative, as shown in the left panel of Fig. 3. Tresp [4] adopted the idea of ME to the case of Gaussian processes and further introduced the mixture of Gaussian processes with the architecture in a similar way, as shown in the left panel of Fig. 4. The major difference between ME and MGP is that the samples of MGP are not independent but correlated, and the prediction is based on the correlation relationships between these samples. In the MGP model, both local experts and gating networks are Gaussian processes. So, we can utilize an MGP to model any general conditional probability density and address the input-dependent bandwidth problem of a stochastic process. The MGP model is further extended to the Dirichlet process based infinite mixture of Gaussian processes in [13].

From an alternative view of ME [3], the ME architecture can be fully generative and uses the posterior responsibilities from a mixture distribution as the gating network, as shown in the right panel of Fig. 3. Based on this architecture, an infinite mixture of Gaussian processes model [6] was also developed. In fact, the generative finite mixture of Gaussian processes has been investigated extensively in recent years [14–18], and the corresponding probabilistic graphical model is shown in Fig. 4. It is clear that the MGP model is generative in the sense that for each sample x_i, we assume there is a latent variable z_i such that (x_i, y_i) is generated from the z_i-th component.

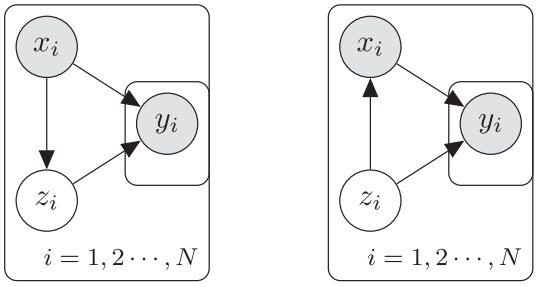


Fig. 3. Illustrations of two mixture of experts architectures. Here, x_i denotes i-th input, y_i denotes i-th output, and the latent variable z_i represents the expert index corresponding to i-th observation. **Left**: the discriminative mixture of experts model. **Right**: the generative mixture of experts model.

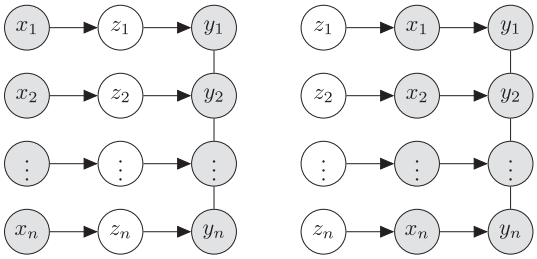


Fig. 4. Illustrations of two kinds of mixture of Gaussian processes. Note that the outputs $y_1, y_2, \cdots, y_n$ are no longer independent. Here, x_i denotes i-th input, y_i denotes i-th output, and the latent variable z_i represents the expert index corresponding to i-th sample. Note that the samples are not identically independently distributed in these models. **Left**: the discriminative mixture of Gaussian processes. **Right**: the generative mixture of Gaussian processes.

To date, there are three major approaches to parameter learning of the mixture model: EM algorithm [19,20], variational Bayesian inference [21], and MCMC [22]. As a stochastic simulation method, MCMC, or more precisely Gibbs sampling method [23] has been successfully applied in MGP [13,6]. However, its time consumption can be prohibitively large if you want to achieve accurate results on a large dataset. Variational Bayesian inference is efficient, but the conditional independent assumption sometimes leads to unsatisfactory results. Nevertheless, variational Bayesian inference has already been employed for the parameter learning of MGP [24,25]. The EM algorithm is a general framework for parameter estimation from incomplete data, which is both effective and efficient. When we use the EM algorithm to estimate the parameters of MGP, the main challenge is that there are exponentially many summations in the Q-function because the samples are not independent. To overcome this difficulty, several approximate EM methods have been proposed [14,26,15]. Certainly, it is a good way to combine these three methods to design new learning algorithms. For example, an MCMC-EM algorithm was already proposed to approximate the Q-function via MCMC sampling [17,18]. Moreover, the MCMC-EM algorithm can be extended to more general models such as the two-layer mixture of Gaussian processes [27] for functional data analysis.

2.2. Robust modeling and heavy-tailed distributions

In regression analysis, there are probably outliers among the observations. Since they deviate far away from the normal observations or samples, they may influence the parameter learning as well as the prediction result strongly. As well-known, the common least square based methods are sensitive to outliers, and thus it is urgent to develop robust regression methods for outliers.

In fact, robust regression models have been investigated extensively, and one popular approach to robust modeling with outliers is to assume heavy-tailed distributions for the noises. In probability theory, heavy-tailed distributions are probability distributions whose tails are not exponentially attenuated. Intuitively, a distribution whose probabilistic density function has a heavier tail than the exponential distribution is heavy-tailed. According to this definition, student-t is heavy-tailed. Although Laplace distribution is not so heavy-tailed, it has a much heavier tail than Gaussian distribution. Actually, it has been used as the distribution of the noise in the least absolute deviation regression [28]. Moreover, student-t distribution and skew-t distribution have been adopted in the mixture-distribution models to enhance the robustness [29–31]. In Fig. 5, we sketch the probabilistic density functions of Gaussian distribution, Laplace distribution and student-t distribution with mean 0 and variance 2.

In conventional Gaussian process analysis, the noise assumption of Gaussian distribution makes it tractable to get the exact ML solutions of the parameters, but the resulted model is very sensitive to outliers. As the noise assumption of heavy-tail distributions can make the model robust to outliers, Laplace distribution and student-t distribution have been adopted into the model of the Gaussian process to obtain a robust regression model [32,8,9]. However, none of these works extends such a robust modeling approach to the MGP model for the regression analysis of multi-modal or non-stationary temporal data.

2.3. Likelihood approximation methods in Gaussian processes

When the noises are assumed to be heavy-tail distributions, it is intractable to get the exact solutions of the parameters by maximizing the likelihood of the regression model on a given dataset. Therefore, we have to approximate the likelihood to make it tractable for finding the ML solution of the parameters as well as the corresponding inference. A comprehensive overview of approximate likelihood inference for Gaussian processes is summarized in [33]. Laplace likelihood approximation can be made with a second-order Taylor expansion of the posterior probability distribution around the posterior mode to construct a Gaussian approximation. Expectation propagation [10] can approximate each likelihood term with an un-normalized Gaussian, and iteratively update the parameters to match marginal moments. In [8], EP is employed to estimate the parameters in robust Gaussian processes with a student-t likelihood. KL-divergence minimization method can approximate the posterior by a Gaussian $\mathcal{N}(\mathbf{h}, \mathbf{A})$, and determine the mean vector and covariance matrix, $\mathbf{h}$, $\mathbf{A}$, by minimizing the reverse KL-divergence between the exact and approximate posteriors. The variational bounding method can be considered as a special case of the KL-divergence minimization method which takes a variational lower bound instead of each likelihood term. By such a variational bound approximation, the intractable integral in each marginal likelihood is converted to a convex optimization problem based on Fenchel-Legendre duality [34]. In this paper, we will apply an optimization procedure to obtain a tighter lower bound of each marginal likelihood iteratively. A double-loop algorithm for variational bounding approximation is also proposed in [35,36] to solve the optimization problem in the majorization-minimization way [37].

The approximate likelihood approaches mentioned above have their advantages under different application scenarios, and there is no generally best one. We adopt the variational bounding method in this paper because it enjoys nice theoretical properties and guaranteed convergence [35,36].

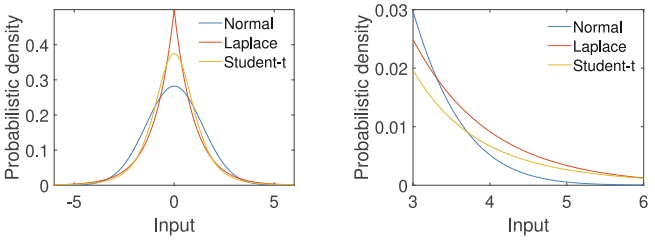


Fig. 5. The probabilistic density functions (pdf) of Gaussian distribution, Laplace distribution and student-t distribution with mean equals to 0 and variance equals to 2. **Left**: pdf over interval $(-6, 6)$; **Right**: pdf over interval $(3, 6)$.

2.4. Recent advances in Gaussian processes

Recently, Deep Gaussian processes (DGP) [38–40] has been an active topic in Gaussian process community. DGP can be seen as a deep extension of the Gaussian Processes Latent Variable Model (GP-LVM) [41]. Originally, GP-LVM was developed as a non-linear dimensional reduction technique. DGP is a generative model aiming at modeling high-dimensional data. This manuscript concerns the nonlinear regression problem, which is supervised, while dimensional-reduction and generative models are unsupervised.

Warped Gaussian processes (WGP) [42] attempts to model non-Gaussian processes by mapping the observations into a latent space. Compared with MGP model, this method resolves the problem that traditional GPs cannot model non-stationary data in a different way. Besides, robustness has not been taken into consideration in WGPs.

Various variational techniques have been developed for learning Gaussian processes in recent years, such as [43–46]. These techniques mainly target at making Gaussian processes scalable on large datasets. Scalability of Gaussian processes is also an important topic, which has also been discussed thoroughly in [47,48]. However, in this manuscript we concern about the robustness and non-stationarity.

3. Gaussian processes and mixture of Gaussian processes

3.1. Gaussian processes

The Gaussian process is one of the dominant models in non-parametric statistics for temporal data regression and classification. Recently, it has been adopted into the field of machine learning and applied widely for the learning and analysis of time series since Gaussian distributions involved in the process can be easily analyzed and processed. Mathematically, the Gaussian process model is defined as follows.

Suppose that there is a dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ and we let $\mathbf{x} = [x_1, x_2, \cdots, x_N]^{\mathrm{T}}$, $\mathbf{y} = [y_1, y_2, \cdots, y_N]^{\mathrm{T}}$ for brevity. We say $\mathbf{y}$ is a Gaussian process with input $\mathbf{x}$ if they are linked by an underlying function f:

$$y_i = f(x_i) + \varepsilon_i, \ \forall i = 1, 2, \cdots, N,$$

where $f(x_i)$ is always subject to a Gaussian distribution and $\{\varepsilon_i\}_{i=1}^N$ are independent noises. Let $\mathbf{f} = [f(x_1), f(x_2), \cdots, f(x_N)]^{\mathrm{T}}$, the definition of Gaussian process is equivalent to say $\mathbf{f}|\mathbf{x}$ is a multivariate Gaussian random vector, *i.e.*, $\mathbf{f}|\mathbf{x} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{C})$. Usually, the mean $\boldsymbol{\mu}$ is

assumed to be zero for simplicity, while the covariance $\mathbf{C}$ is determined by $\mathbf{x}$. Given a covariance function $c(\cdot,\cdot;\boldsymbol{\theta})$ parameterized by $\boldsymbol{\theta}$, then $\mathbf{C}_{ij} = c(x_i, x_j; \boldsymbol{\theta})$.

There are many choices for covariance functions [1]. Generally, the choice of proper covariance function is highly related to the data. In this paper, we consider the squared exponential covariance function which has been widely used, but the extension to other kinds of covariance functions are direct. Mathematically, the squared exponential covariance function is defined as

$$c(x_i, x_j|\boldsymbol{\theta}) = \theta_1^2 \exp\left(-\theta_2^2 \frac{(x_i - x_j)^2}{2}\right).$$

As indicated in [1,49], θ_1 is the vertical scale of variations, θ_2^{-1} is the length scale or so-called bandwidth. However, $\mathbf{f}$ are latent variable and only $\mathbf{x}, \mathbf{y}$ are observed. In order to inference over hyperparameters, one usually assume $\{\varepsilon_i\}_{i=1}^N$ are independent Gaussian noise, *i.e.*, $y_i|f_i \sim \mathcal{N}(f_i, \gamma^2)$. Integrating out $\mathbf{f}$, we obtain

$$\mathbf{y}|\mathbf{x} \sim \mathcal{N}(0, \mathbf{C} + \gamma^2 \mathbf{I}_N).$$

Thus, one may learn parameters $\boldsymbol{\theta}, \gamma$ via Type-II maximum likelihood estimation (marginal likelihood maximization) [1]. Since the dimension of the Gaussian distribution is N, the time complexity of parameter learning is $\mathcal{O}(N^3)$, which is very high when N is relatively large.

Given a new input x_*, the aim of prediction is to estimate the corresponding output y_*. Note that it is equivalent to predict f_*, since $y_*|f_* \sim \mathcal{N}(f_*, \gamma^2)$. From the definition of Gaussian process we immediately have

$$\begin{bmatrix} \mathbf{y} \\ f_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ 0 \end{bmatrix}, \begin{bmatrix} \mathbf{C} + \gamma^2 \mathbf{I}_N & \mathbf{c}(\mathbf{x}, x_*) \\ \mathbf{c}(x_*, \mathbf{x}) & c(x_*, x_*) \end{bmatrix}\right).$$

Using the conditional properties of Gaussian distribution, we further have that $f_*|\mathscr{D}, x_*$ also obeys a Gaussian distribution,

$$\begin{aligned} &f_*|\mathscr{D}, x_* \sim \mathcal{N}(\mu_f, \sigma_f^2), \\ &\mu_f = \mathbf{c}(x_*, \mathbf{x})(\mathbf{C} + \gamma^2 \mathbf{I}_N)^{-1}\mathbf{y} \\ &\sigma_f^2 = c(x_*, x_*) - \mathbf{c}(x_*, \mathbf{x})(\mathbf{C} + \gamma^2 \mathbf{I}_N)^{-1}\mathbf{c}(\mathbf{x}, x_*). \end{aligned} \tag{1}$$

The time complexity for prediction is $\mathcal{O}(N^2)$ if the precision matrix $(\mathbf{C} + \sigma^2 \mathbf{I}_N)^{-1}$ has been calculated.

Although the assumption of Gaussian noises makes the inference very easy, this conventional Gaussian process is not robust because Gaussian distributions are not heavy-tailed. In practical applications, the data may contain some outliers due to technical reasons and we certainly do not want these outliers to influence the final result too much. To get rid of this problem, heavy-tailed distributions have been adopted to model the noises, such as Laplace noise and student-t noise. The corresponding models are referred to as robust Gaussian processes. In these cases, it is intractable to integrate out $\mathbf{f}$ to obtain the marginal likelihood, and thus the approximate inference methods such as Laplace approximation, expectation propagation, and variational Bayesian approximation are often employed to tackle this critical problem.

3.2. Mixture of Gaussian processes

As the previously defined Gaussian processes are always stationary, the mixture of Gaussian processes (MGP) has been introduced to model multi-model data or non-stationary time series. Moreover, the time complexity of learning a MGP is greatly decreased reduced since the training samples are divided by these Gaussian processes and the computation for the inverses of their covariance matrices becomes much easier. In fact, the time complexity of learning a single Gaussian process is $\mathcal{O}(N^3)$, which becomes prohibitively high as the number N of samples is large. Here, we assume that the MGP model is generative and there are K independent Gaussian processes involved in the mixture along the input region. Mathematically, suppose that $\{z_i\}_{i=1}^N$ are latent variables indicating which component (x_i, y_i) belongs to. The mixing proportions $\{\pi_k\}_{k=1}^K$ are non-negative scalars satisfying $\sum_{k=1}^K \pi_k = 1$, and

$$p(z_i = k) = \pi_k.$$

Conditioned on $z_i = k$, we assume the input x_i is generated from a Gaussian distribution, that is,

$$x_i|z_i = k \sim \mathcal{N}(\mu_k, \sigma_k^2),$$

where μ_k and σ_k are mean and standard deviation, respectively. In fact, this is equivalent to assume that all the inputs are generated by a Gaussian mixture model. We further divide the samples by these z_i and define the component Gaussian processes as follows.

$$\begin{aligned} &\mathbf{x}_k = \{x_i|z_i = k, i = 1, \cdots, N\}, \quad \mathbf{f}_k = \{f_i|z_i = k, i = 1, \cdots, N\}, \\ &\mathbf{y}_k = \{y_i|z_i = k, i = 1, \cdots, N\}, \end{aligned}$$

and $\mathbf{C}_k$ be the covariance matrix of k-th Gaussian processes parameterized by $\boldsymbol{\theta}_k$, then

$$\mathbf{f}_k|\mathbf{x}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_k), \ k = 1, 2, \cdots, K.$$

As for the noises, we usually assume $\{\varepsilon_i|z_i = k\}$ shares the same Gaussian distribution, with standard deviation γ_k. Integrating out $\mathbf{f}_k$, we have

$$\mathbf{y}_k|\mathbf{x}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{C}_k + \gamma_k^2 \mathbf{I}).$$

Similar to the Gaussian mixture model, we can use the EM algorithm to learn the parameters $\boldsymbol{\Theta} = \{\pi_k, \mu_k, \sigma_k, \boldsymbol{\theta}_k, \gamma_k\}_{k=1}^K$ of MGP iteratively. In fact, the log likelihood function of the MGP model is given by

$$\mathscr{L}(\boldsymbol{\Theta}, \mathbf{z}) = \sum_{k=1}^K \left(\sum_{i=1}^N \mathbb{I}(z_i = k)[\log \pi_k + \log p(x_i; \mu_k, \sigma_k)] + \log p(\mathbf{y}_k|\mathbf{x}_k; \boldsymbol{\theta}_k, \gamma_k)\right). \tag{2}$$

The M-step of the EM algorithm is relatively easy: given $\mathbf{z}$, $\{\pi_k, \mu_k, \sigma_k\}_{k=1}^K$ can be updated by the standard EM formulae for Gaussian mixtures, while the update of $\{\boldsymbol{\theta}_k, \gamma_k\}_{k=1}^K$ can be obtained by learning the parameters of K Gaussian processes from their datasets separately. On the other hand, the E-step of the EM algorithm is rather difficult because $\mathscr{L}$ is not separable with respect to $\mathbf{z}$, thus there are exponentially many $\mathcal{O}(K^N)$ terms in the summation. In order to avoid the summation of these exponentially many terms, the hard-cut EM algorithm simply ignores the dependency between samples in the E-step. More precisely, in the E-step of hard-cut EM algorithm, we update z_i by

$$z_i = \arg\max_{k=1,2,\cdots,K} \pi_k p(x_i; \mu_k, \sigma_k) p(y_i|x_i, \boldsymbol{\theta}_k, \gamma_k).$$

Since the noises are assumed to be Gaussian, $y_i|x_i, \boldsymbol{\theta}_k, \gamma_k$ is just $\mathcal{N}(0, \theta_{k,1}^2 + \gamma_k^2)$, which can be calculated directly. Using this approximation, the time consumption of E-step is almost negligible, while a single M-step requires about $\mathcal{O}(K(N/K)^3) = \mathcal{O}(N^3/K^2)$. As a result, if the number of required EM iteration is T, the overall time complexity of the hard-cut EM algorithm is $\mathcal{O}(TN^3/K^2)$.

As long as the parameters of MGP are estimated by the EM algorithm, we can predict the output $\hat{y}$ at x_*. We firstly calculate the probability that x_* belongs to each component as

$$p(z_* = k) \propto \pi_k \mathcal{N}(x_*; \mu_k, \sigma_k^2),$$

and then get the final prediction result by

$$\hat{y} = \sum_{k=1}^{K} p(z_* = k)\hat{y}^{(k)},$$

where $\hat{y}^{(k)}$ is assumed to be the prediction result of the k-th Gaussian process at x_* given by Eq. (1).

4. The proposed mixture of robust Gaussian processes

4.1. Model formulation

Although the mixture of Gaussian processes is quite effective to model non-stationary temporal data which has a multi-model structure, it is still sensitive to outlier and thus not so robust in practical applications. In order to overcome this problem, we propose a mixture of robust Gaussian processes in which the noises of each component Gaussian process are assumed to be heavy-tail distributions instead of Gaussian ones. Here, we mainly consider Laplace noise and student-t noise. For a Laplace noise, the corresponding likelihood term takes the following form:

$$p(y_i|f_i;\gamma) = \frac{1}{2\gamma}\exp\left(-\frac{|y_i - f_i|}{\gamma}\right).$$

where γ is the dispersion parameter. For a student-t noise, the likelihood term is given by

$$p(y_i|f_i;\gamma) = \frac{\Gamma(\frac{\gamma+1}{2})}{\Gamma(\frac{\gamma}{2})}\frac{1}{\sqrt{\gamma\pi}}\left(1 + \frac{(y_i - f_i)^2}{\gamma}\right)^{-\frac{\gamma+1}{2}},$$

where $\Gamma(\cdot)$ is the Gamma function and γ is now the degree of freedom. The mixture of robust Gaussian processes can be defined in the same way as the mixture of Gaussian processes in Section 3.2 except that the probability density function of each noise is that of Laplace or student-t. Although the formal modification seems minor, it makes the analysis far more challenging because $\mathbf{y}_k|\mathbf{x}_k$ is no longer Gaussian. In fact, the marginal likelihood of the k-th component Gaussian process, i.e., the conditional probability of $\mathbf{y}_k$ with respect to $\mathbf{x}_k$, is

$$p(\mathbf{y}_k|\mathbf{x}_k;\theta_k,\gamma_k) = \int p(\mathbf{y}_k|\mathbf{f}_k;\gamma_k)p(\mathbf{f}_k|\mathbf{x}_k;\theta_k)\mathrm{d}\mathbf{f}_k. \tag{3}$$

The main difficulty arises from the intractable integral in $p(\mathbf{y}_k|\mathbf{x}_k;\theta_k,\gamma_k)$. On one hand, in the M-step, the updating of $\{\theta_k,\gamma_k\}$ becomes difficult because $p(\mathbf{y}_k|\mathbf{x}_k;\theta_k,\gamma_k)$ does not have a closed-form expression. On the other hand, in the E-step, the allocation of z_i involves the calculation of $p(\mathbf{y}_i|x_i,\theta_k,\gamma_k)$ that is also intractable. Instead of calculating the marginal likelihood explicitly, we employ a variational bounding method to calculate the marginal likelihood approximately. Moreover, we need the posterior $\mathbf{f}_k$ of each component Gaussian process in the prediction stage. Since the noises are non-Gaussian, $[\mathbf{y}_k;f_*]$ is not Gaussian but $[\mathbf{f}_k;f_*]$ is still Gaussian and we can predict f_* by conditioning on $\mathbf{f}_k$. As we will see, this approximate marginal likelihood can lead to an approximate posterior at the same time.

4.2. Variational approximation of marginal likelihood

For ease of notation, we omit the subscript k in marginal likelihood and consider one single Gaussian process temporarily, since we learn each Gaussian process expert separately in the M-step. The variational bounding method introduced here has been used in large scale linear models [35], image processing [36] and Gaussian processes [50], but there are no complete derivations for the case of robust Gaussian processes to the best of our knowledge. For completeness, we give the strict derivations of the variational bounding approximation results for the marginal likelihood of robust Gaussian processes in detail, which are summarized in the following theorem.

Theorem 1. *Let $t(s;\gamma)$ be a super-Gaussian probabilistic density function centered at $0, g(x) = \log t(\sqrt{x})$ and $h(\lambda) = 2g^*(-1/(2\lambda))$ where g^* is the Fenchel-Legendre dual function of g. Furthermore, let*

$$\boldsymbol{\lambda} = \begin{bmatrix}\lambda_1\\ \lambda_2,\\ \vdots\\ \lambda_n\end{bmatrix},\ \boldsymbol{\Lambda} = \begin{bmatrix}\lambda_1 & 0 & \cdots & 0\\ 0 & \lambda_2 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & \lambda_n\end{bmatrix},\ h(\boldsymbol{\lambda}) = \sum_{i=1}^{n} h(\lambda_i),$$

we have the following theoretical results:

(a) *The marginal likelihood $p(\mathbf{y}|\mathbf{x};\theta,\gamma)$ takes the following variational form*

$$\begin{aligned} p(\mathbf{y}|\mathbf{x};\theta,\gamma) &= |\mathbf{C}|^{-1/2}\exp\left(-\tfrac{1}{2}\min_{\boldsymbol{\lambda}\geq 0,\mathbf{f}}\varphi(\boldsymbol{\lambda},\mathbf{f})\right),\\ \varphi(\boldsymbol{\lambda},\mathbf{f}) &= h(\boldsymbol{\lambda}) + \log|\mathbf{A}| + (\mathbf{y}-\mathbf{f})^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}(\mathbf{y}-\mathbf{f}) + \mathbf{f}\mathbf{C}^{-1}\mathbf{f}. \end{aligned}$$

(b) *Given $\mathbf{v} = \nabla_{\lambda^{-1}}\log|\mathbf{A}| = \mathrm{diag}((\mathbf{C}^{-1}+\boldsymbol{\Lambda}^{-1})^{-1})$,*

$$\phi_{\mathbf{v}}(\boldsymbol{\lambda},\mathbf{f}) = \sum_{i=1}^{n}\frac{v_i + (y_i - f_i)^2}{\lambda_i} + h(\lambda_i) + \mathbf{f}\mathbf{C}^{-1}\mathbf{f} - g_1^*(\mathbf{v})$$

is an upper bound for $\varphi(\boldsymbol{\lambda},\mathbf{f})$.

(c) *$\phi_{\mathbf{v}}(\boldsymbol{\lambda},\mathbf{f})$ can be further optimized by the alternative minimization of $\boldsymbol{\lambda}$ and $\mathbf{f}$): the minimum of $\boldsymbol{\lambda}$ is $\lambda_i = -(2g\prime(v_i + (y_i - f_i)^2))^{-1}$ when $\mathbf{f}$ is fixed, while the minimum of $\mathbf{f}$ can be obtained by the gradient descent algorithm when $\boldsymbol{\lambda}$ is fixed.*

Proof. We prove the three results one by one. Since $t(s;\gamma)$ is a super-Gaussian [34] probabilistic density function (for example, Laplace distribution or Student-t distribution) centered at $0, \log t(s;\gamma)$ is symmetric and monotone decreasing with respect to s. Furthermore, $\sqrt{s} \to \log t(s;\gamma)$ is convex whenever $s \geqslant 0$, so $g(x) = \log t(\sqrt{x})$ is convex and monotone decreasing. According to Fenchel-Legendre duality, we have

$$g(x) = \sup_{w}(xw - g^*(w)),$$

where $g^*(w) = \sup_{x\geqslant 0}(xw - g(x))$ is the dual function of $g(x)$. Because $g(x)$ is decreasing, the domain of $g^*(w)$ must be $\mathbf{bR}_-$. Therefore, it is equivalent to have

$$g(x) = \sup_{w\leqslant 0}(xw - g^*(w)) = \sup_{w\geqslant 0}(-xw - g^*(-w)).$$

By taking $\lambda = 1/(2w), h(\lambda) = 2g^*(-1/(2\lambda))$, we have

$$\begin{aligned}\log t(s;\gamma) = g(s^2) &= \sup_{w\geqslant 0}\left(-\frac{s^2}{2\lambda} - g^*(-\frac{1}{2\lambda})\right)\\ &= \sup_{\lambda\geqslant 0}\left(-\frac{s^2}{2\lambda} - \frac{1}{2}h(\lambda)\right).\end{aligned} \tag{4}$$

Since $p(\mathbf{y}|\mathbf{f};\gamma) = \prod_{i=1}^{n}p(y_i|f_i;\gamma)$ is decomposable with respect to individual observations, we can consider each individual term separately. Putting $p(y_i|f_i;\gamma) = p(f_i - y_i|0;\gamma) = t(f_i - y_i)$ into Eq. (4), we obtain a variational representation of $\log p(y_i|f_i;\gamma)$ as follows:

$$\log p(y_i|f_i;\gamma) = \sup_{\lambda_i\geqslant 0}\left(-\frac{(y_i - f_i)^2}{2\lambda_i} - \frac{1}{2}h(\lambda_i)\right).$$

Consequently, we have

$$\log p(\mathbf{y}|\mathbf{f};\gamma) = \sup_{\boldsymbol{\lambda}\geqslant 0}\left(-\frac{1}{2}(\mathbf{y}-\mathbf{f})^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}(\mathbf{y}-\mathbf{f}) - \frac{1}{2}h(\boldsymbol{\lambda})\right).$$

With this representation, we further have $p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta},\gamma)$:

$$\begin{aligned} p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta},\gamma) &= \int p(\mathbf{y}|\mathbf{f};\gamma)p(\mathbf{f}|\mathbf{x};\boldsymbol{\theta})\mathrm{d}\mathbf{f} \\ &= \int \exp\left(\log p(\mathbf{y}|\mathbf{f};\gamma)\right)\mathcal{N}(\mathbf{f}|0,\mathbf{C})\mathrm{d}\mathbf{f} \\ &= \max_{\boldsymbol{\lambda}\geqslant 0}\exp\left(-\tfrac{1}{2}h(\boldsymbol{\lambda})\right)\int \exp\left(-\tfrac{1}{2}(\mathbf{y}-\mathbf{f})\boldsymbol{\Lambda}^{-1}(\mathbf{y}-\mathbf{f})\right)\mathcal{N}(\mathbf{f}|0,\mathbf{C})\mathrm{d}\mathbf{f}. \end{aligned} \tag{5}$$

The term inside the integral is actually a lower bound of the posterior of $\mathbf{f}$. By completing the squares, we can find out that the approximate posterior of $\mathbf{f}$ given observations $\mathbf{x},\mathbf{y}$ and parameter $\boldsymbol{\lambda}$ is Gaussian $\mathcal{N}(\mathbf{h},\mathbf{A}^{-1})$ where

$$\mathbf{A} = \boldsymbol{\Lambda}^{-1} + \mathbf{C}^{-1}, \mathbf{h} = \mathbf{A}^{-1}\boldsymbol{\Lambda}^{-1}\mathbf{y}. \tag{6}$$

The integral in Eq. (5) can be calculated explicitly, but we seek a variational form for further purpose. Given $\mathbf{h}$ and $\mathbf{A}$ defined above, we have

$$\begin{aligned} &\frac{\exp\left(-\frac{1}{2}(\mathbf{y}-\mathbf{f})\boldsymbol{\Lambda}^{-1}(\mathbf{y}-\mathbf{f})\right)\mathcal{N}(\mathbf{f}|0,\mathbf{C})}{\int \exp\left(-\frac{1}{2}(\mathbf{y}-\mathbf{f})\boldsymbol{\Lambda}^{-1}(\mathbf{y}-\mathbf{f})\right)\mathcal{N}(\mathbf{f}|0,\mathbf{C})\mathrm{d}\mathbf{f}} \\ &= \frac{|\mathbf{A}|^{1/2}}{(2\pi)^{n/2}}\exp\left(-\frac{1}{2}(\mathbf{f}-\mathbf{h})^{\mathrm{T}}\mathbf{A}(\mathbf{f}-\mathbf{h})\right). \end{aligned}$$

Taking maximum with respect to $\mathbf{f}$, we further have

$$\begin{aligned} &\int \exp\left(-\tfrac{1}{2}(\mathbf{y}-\mathbf{f})\boldsymbol{\Lambda}^{-1}(\mathbf{y}-\mathbf{f})\right)\mathcal{N}(\mathbf{f}|0,\mathbf{C})\mathrm{d}\mathbf{f} \\ &= \tfrac{(2\pi)^{n/2}}{|\mathbf{A}|^{1/2}}\max_{\mathbf{f}}\exp\left(-\tfrac{1}{2}(\mathbf{y}-\mathbf{f})\boldsymbol{\Lambda}^{-1}(\mathbf{y}-\mathbf{f})\right)\mathcal{N}(\mathbf{f}|0,\mathbf{C}) \\ &= \tfrac{1}{|\mathbf{AC}|^{1/2}}\max_{\mathbf{f}}\exp\left(-\tfrac{1}{2}(\mathbf{y}-\mathbf{f})\boldsymbol{\Lambda}^{-1}(\mathbf{y}-\mathbf{f}) - \tfrac{1}{2}\mathbf{f}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{f}\right). \end{aligned} \tag{7}$$

By combining Eq. (5) and (7), we can rewrite $p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta},\gamma)$ into a variational form with respect to $\mathbf{f}$ and $\boldsymbol{\lambda}$ as

$$\begin{aligned} p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta},\gamma) &= |\mathbf{C}|^{-1/2}\exp\left(-\tfrac{1}{2}\min_{\boldsymbol{\lambda}\geq 0,\mathbf{f}}\varphi(\boldsymbol{\lambda},\mathbf{f})\right), \\ \varphi(\boldsymbol{\lambda},\mathbf{f}) &= h(\boldsymbol{\lambda}) + \log|\mathbf{A}| + (\mathbf{y}-\mathbf{f})^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}(\mathbf{y}-\mathbf{f}) + \mathbf{f}\mathbf{C}^{-1}\mathbf{f}. \end{aligned}$$

This proves part (a) of Theorem 1.

It follows from (a) that the problem of calculating marginal likelihood $p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta},\gamma)$ boils down to solve the optimization problem $\min_{\boldsymbol{\lambda}\geq 0,\mathbf{f}}\varphi(\boldsymbol{\lambda},\mathbf{f})$. The main difficulty lies in the term $\log|\mathbf{A}|$ that depends on $\boldsymbol{\lambda}$. We use a majorization-minimization method to tackle this problem. As shown in [36], let $\boldsymbol{\lambda}^{-1} = [\lambda_1^{-1},\lambda_2^{-2},\cdots,\lambda_n^{-1}]^{\mathrm{T}}$, then $\boldsymbol{\lambda}^{-1}\to -\log|\mathbf{A}|$ is a convex function. Again by Fenchel-Legendre transformation, we obtain $\log|\mathbf{A}| = \min_{\mathbf{v}\geqslant 0}\left(\mathbf{v}^{\mathrm{T}}\boldsymbol{\lambda}^{-1} - g_1^*(\mathbf{v})\right)$, where g_1^* is the dual function. For a fixed $\boldsymbol{\lambda}^{-1}$, the equality holds when

$$\mathbf{v} = \nabla_{\boldsymbol{\lambda}^{-1}}\log|\mathbf{A}| = \mathrm{diag}((\mathbf{C}^{-1}+\boldsymbol{\Lambda}^{-1})^{-1}). \tag{8}$$

For a general $\mathbf{v}\geqslant 0, \log|\mathbf{A}| \leqslant \mathbf{v}^{\mathrm{T}}\boldsymbol{\lambda}^{-1} - g_1^*(\mathbf{v})$, and we thus have

$$\begin{aligned} \phi_{\mathbf{v}}(\boldsymbol{\lambda},\mathbf{f}) &= \mathbf{v}^{\mathrm{T}}\boldsymbol{\lambda}^{-1} - g_1^*(\mathbf{v}) + h(\boldsymbol{\lambda}) + (\mathbf{y}-\mathbf{f})^{\mathrm{T}}\boldsymbol{\Lambda}^{-1}(\mathbf{y}-\mathbf{f}) + \mathbf{f}\mathbf{C}^{-1}\mathbf{f} \\ &= \sum_{i=1}^{n}\tfrac{v_i+(y_i-f_i)^2}{\lambda_i} + h(\lambda_i) + \mathbf{f}\mathbf{C}^{-1}\mathbf{f} - g_1^*(\mathbf{v}), \end{aligned}$$

which is an upper bound of $\varphi(\boldsymbol{\lambda},\mathbf{f})$. This proves part (b) of Theorem 1.

We further employ the double loop optimization method to optimize $\varphi(\boldsymbol{\lambda},\mathbf{f})$, *i.e.*, to get a tighter bound instead. In the inner loop, we minimize the upper bound $\varphi_{\mathbf{v}}(\boldsymbol{\lambda},\mathbf{f})$ given $\mathbf{v}$. While in the outer loop, we update current $\mathbf{v}$ to achieve tighter bounds. In the inner loop there are two variables, and we solve this optimization problem with the two variables alternatively. For the $\boldsymbol{\lambda}$ part, according to the definition of $h(\lambda_i)$, it is easy to get

$$\begin{aligned} \min_{\lambda_i\geq 0}\tfrac{v_i+(y_i-f_i)^2}{\lambda_i} + h(\lambda_i) &= \min_{\lambda_i\geq 0}\left(2\tfrac{v_i+(y_i-f_i)^2}{2\lambda_i} + 2g^*\left(-\tfrac{1}{2\lambda_i}\right)\right) \\ &= -2\max_{\lambda_i\geq 0}\left(-\tfrac{v_i+(y_i-f_i)^2}{2\lambda_i} - g^*(-2\lambda_i)\right) \\ &= -2g(v_i+(y_i-f_i)^2) = -2\log t\left(\sqrt{v_i+(y_i-f_i)^2}\right) \\ &= -2\log p\left(y_i|\sqrt{v_i+(y_i-f_i)^2}+y_i;\gamma\right). \end{aligned}$$

At first glance, this result is a little surprising since we do not need to know the explicit form of h. By the optimal condition, when f_i is fixed, the optimal value is obtain when $\lambda_i = -(2g'(v_i+(y_i-f_i)^2))^{-1}$. After updating $\boldsymbol{\lambda}$, the objective function becomes

$$\mathbf{f}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{f} - 2\sum_{i=1}^{n}\log p\left(y_i|\sqrt{v_i+(y_i-f_i)^2}+y_i;\gamma\right),$$

which can be easily solved using the gradient descent algorithm. This process is very similar to finding the posterior mode of $\mathbf{f}$, the only difference here is the likelihood term is smoothed by $\mathbf{v}$.

To calculate approximate marginal likelihood $p(\mathbf{y}|\mathbf{x};\boldsymbol{\theta},\gamma)$ according to Theorem 1 in practice, we begin to iteratively update $\boldsymbol{\lambda}$ and $\mathbf{f}$ in the inner loop until convergence. Then, we turn to the outer loop and update $\mathbf{v}$ according to Eq. (8) to get a new upper bound for $\varphi(\boldsymbol{\lambda},\mathbf{f})$. The entire optimization process is guaranteed to converge due to the convexity [35,36]. After the process has converged, we also obtain an approximation for the posterior of $\mathbf{f}$ given $\mathbf{y}$ as indicated in Eq. (6). The approximate marginal likelihood in Theorem 1 involves optimization procedure and has no explicit formula, so it is still intractable to calculate gradients with respect to $\boldsymbol{\theta}$ and γ. However, part (c) of Theorem 1 gives an $\mathbf{f}$, which is an estimation of latent function values without noises. Therefore we can approximate the gradients by taking derivates of the surrogate log-likelihood:

$$-\frac{1}{2}\log|\mathbf{C}| - \frac{1}{2}\mathbf{f}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{f} + \sum_{i=1}^{n}\log t(y_i-f_i;\gamma).$$

4.3. The hard-cut EM algorithm

We further establish the hard-cut EM algorithm for mixtures of robust Gaussian processes in this subsection. In the E-step, we need to calculate the expectation of complete log likelihood (2) with respect to posterior distributions of $\mathbf{z}$. Since the samples are dependent, this task involves K^N summations and is thus computational intractable. In order to get rid of this difficulty, we employ a kind of hard-cut allocation to update $\mathbf{z}$ as follows:

$$z_i = \arg\max_{k=1,2,\cdots,K}\pi_k p(x_i;\mu_l,\sigma_k)p(y_i|x_i,\theta_k,\gamma_k). \tag{9}$$

That is, all the samples are assumed to be separable to those components of the mixture in each iteration. Although this assumption may be too strict, it can make the EM algorithm tractable. Moreover, the experiments have already demonstrated that this hard-cut strategy is reasonable and effective. Therefore, we adopt this strategy to design the hard-cut EM algorithm.

In the mixture of robust Gaussian processes, the noises are not Gaussian, so the exact calculation of $p(y_i|x_i,\theta_k,\gamma_k)$ is intractable as we discussed previously. In order to solve this problem efficiently, we implement the variational Bayesian approximated method

given in Section 4.2 in the EM algorithm. With the hard-cut version of $\mathbf{z}$, let $\mathscr{I}_k = \{i|z_i = k\}$, then the Q-function is given by

$$Q(\boldsymbol{\Theta}) = \sum_{k=1}^{K}\sum_{i\in\mathscr{I}_k}(\log\pi_k + \log p(x_i;\mu_k,\sigma_k)) + \sum_{k=1}^{K}\log p(\mathbf{y}_k|\mathbf{x}_k;\boldsymbol{\theta}_k,\gamma_k).$$

The M-step aims to maximize $Q(\boldsymbol{\Theta})$ with respect to all the parameters. For $\{\pi_k,\mu_k,\sigma_k\}_{k=1}^K$, we have

$$\pi_k = \frac{|\mathscr{I}_k|}{N},\quad \mu_k = \frac{1}{|\mathscr{I}_k|}\sum_{i\in\mathscr{I}_k}x_i,\quad \sigma_k = \sqrt{\frac{1}{|\mathscr{I}_k|}\sum_{i\in\mathscr{I}_k}(x_i-\mu_k)^2}. \tag{10}$$

As for hyper-parameters of robust Gaussian processes, we firstly approximate $p(\mathbf{y}_k|\mathbf{x}_k;\boldsymbol{\theta}_k,\gamma_k)$ via the variational bounding method, and then optimize the approximate marginal likelihood via the gradient ascent algorithm. As the approximations are local, we also need to update the approximations once we update $\{\boldsymbol{\theta}_k,\gamma_k\}_{k=1}^K$. In fact, the procedure of learning $\{\boldsymbol{\theta}_k,\gamma_k\}_{k=1}^K$ in a single M-step can be seen as a variational EM algorithm as indicated in [9]. Since we need to run the double loop iteration for each Gaussian process component in every M-step, the total time consumption may be rather large. We summarize the harde-cut EM algorithm for mixtures of robust Gaussian processes in Algorithm 1.

Algorithm 1. The hard-cut EM algorithm for MRGP

Input: observations $\{(x_i,y_i)\}_{i=1}^N$, the number of components K.
Parameters: mixing proportions $\{\pi_k\}_{i=1}^K$, mixture parameters $\{\mu_k,\sigma_k\}_{k=1}^K$, Gaussian process parameters $\{\boldsymbol{\theta}_k,\gamma_k\}_{k=1}^K$.
Latent variables: latent variables $\{z_i\}_{i=1}^N$.
1: Initialize $\{z_i\}_{i=1}^N$ via k-means.
2: **while** not converged **do**
3: **for** $k = 1,2,\cdots,K$ **do**
4: Update mixture parameters of k-th component according to Eq. (10).
5: **while** not converged **do**
6: Approximate $p(\mathbf{y}_k|\mathbf{x}_k;\boldsymbol{\theta}_k,\gamma_k)$ using Theorem 1.
7: Update $\boldsymbol{\theta}_k,\gamma_k$ using gradient ascent.
8: **end while**
9: **end for**
10: **for** $i = 1,2,\cdots,N$ **do**
11: Update z_i via hard-cut allocation according to Eq. (9).
12: **end for**
13: **end while**

4.4. Prediction strategy

We take the following prediction strategy with the obtained mixture of robust Gaussian processes. Given a new input x_*, we firstly calculate the probability that x_* belongs to k-th component as follows:

$$p(z_* = k) = \frac{\pi_k\mathscr{N}(x_*;\mu_k,\sigma_k^2)}{\sum_{l=1}^K\pi_l\mathscr{N}(x_*;\mu_l,\sigma_l^2)}.$$

Next, we consider the prediction conditioned on $z_* = k$. This is different from the case of Gaussian process (1) because $[\mathbf{y}_k;f_*^{(k)}]$ is not jointly Gaussian. Nevertheless, $[\mathbf{f}_k;f_*^{(k)}]$ is still Gaussian and thus $f_*^{(k)}|\mathbf{f}_k$ is easy to calculate, although we don't have access to $\mathbf{f}_k$ directly. The posterior of $\mathbf{f}_k|\mathbf{x}_k,\mathbf{y}_k;\boldsymbol{\theta}_k,\gamma_k$ is approximated as a Gaussian distribution $\mathscr{N}(\mathbf{f}_k|\mathbf{h}_k,\mathbf{A}_k^{-1})$ as in Eq. (6). Therefore, we can integrate out $\mathbf{f}_k$ with respect to its approximate posterior distribution $\mathscr{N}(\mathbf{f}_k|\mathbf{h}_k,\mathbf{A}_k^{-1})$, and the predictive distribution of $f_*^{(k)}$ is given by

$$\begin{aligned} p(f_*^{(k)}|x_*,\mathbf{x}_k,\mathbf{y}_k) &= \int p(f_*^{(k)}|\mathbf{f}_k,x_*,\mathbf{x}_k)p(\mathbf{f}_k|\mathbf{x}_k,\mathbf{y}_k;\boldsymbol{\theta}_k,\gamma_k)\mathrm{d}\mathbf{f}_k \\ &\approx \int p(f_*^{(k)}|\mathbf{f}_k,x_*,\mathbf{x}_k)\mathscr{N}(\mathbf{f}_k|\mathbf{h}_k,\mathbf{A}_k^{-1})\mathrm{d}\mathbf{f}_k. \end{aligned}$$

Simple calculation reveals that the approximate predictive distribution of $f_*^{(k)}$ is also Gaussian, whose mean is $(\mathbf{c}_*^{(k)})^{\mathrm{T}}\mathbf{C}_k^{-1}\mathbf{h}_k$ where $(\mathbf{c}_*^{(k)}) = c(\mathbf{x}_k,x_*;\boldsymbol{\theta}_k)$. Therefore, the prediction at x_* conditioned on $z_* = k$ is $\hat{y}^{(k)} = (\mathbf{c}_*^{(k)})^{\mathrm{T}}\mathbf{C}_k^{-1}\mathbf{h}_k$ and the final prediction is obtained by

$$\hat{f} = \sum_{k=1}^K p(z_* = k)\hat{y}^{(k)}.$$

5. Experimental results

5.1. On synthetic data

We firstly evaluate the proposed MRGP models with the hard-cut EM algorithm on a group of typical synthetic datasets, as in [17,18,27]. These synthetic datasets are generated from typical mixtures of Gaussian processes, being referred to as $\mathscr{S}_1,\cdots,\mathscr{S}_{12}$, respectively. Among them, $\mathscr{S}_1$ and $\mathscr{S}_7$ are basic and their parameters are listed in Table 1. Based on $\mathscr{S}_1$ and $\mathscr{S}_7$, we vary the noise level, overlapping level and mixing proportions to obtain other datasets.

Specifically, $\mathscr{S}_1$ consists of 900 samples, and we randomly select 1/3 of them for training and the rest for testing. There are three actual Gaussian processes in $\mathscr{S}_1$, *i.e.*, $K = 3$ in this dataset. $\mathscr{S}_2 - \mathscr{S}_6$ are based on $\mathscr{S}_1$ with the variations as follows:

(a) $\mathscr{S}_2$ (a more noisy dataset): $[\boldsymbol{\theta}_1]_3 = [\boldsymbol{\theta}_2]_3 = [\boldsymbol{\theta}_3]_3 = 0.2$.
(b) $\mathscr{S}_3$ (a less noisy dataset): $[\boldsymbol{\theta}_1]_3 = [\boldsymbol{\theta}_2]_3 = [\boldsymbol{\theta}_3]_3 = 0.05$.
(c) $\mathscr{S}_4$ (an unbalanced dataset): $\pi_1 = 0.2,\pi_2 = 0.5,\pi_3 = 0.3$.
(d) $\mathscr{S}_5$ (a mildly overlapping dataset): $\sigma_1^2 = \sigma_2^2 = \sigma_3^2 = 1.2$.
(e) $\mathscr{S}_6$ (a heavily overlapping dataset): $\sigma_1^2 = \sigma_2^2 = \sigma_3^2 = 2$.

$\mathscr{S}_7$ is also a basic dataset with five actural Gaussian processes. It contains 1500 samples, in which 500 are used for training and 1000 are used for testing. From $\mathscr{S}_7,\mathscr{S}_8 - \mathscr{S}_{12}$ are obtained in a similar way as above. In particular, for $\mathscr{S}_{10}$, the mixing proportions are set as $\pi_1 = 0.15,\pi_2 = 0.25,\pi_3 = 0.2,\pi_4 = 0.25,\pi_5 = 0.15$. The sketches of five typical synthetic datasets are shown in Fig. 6.

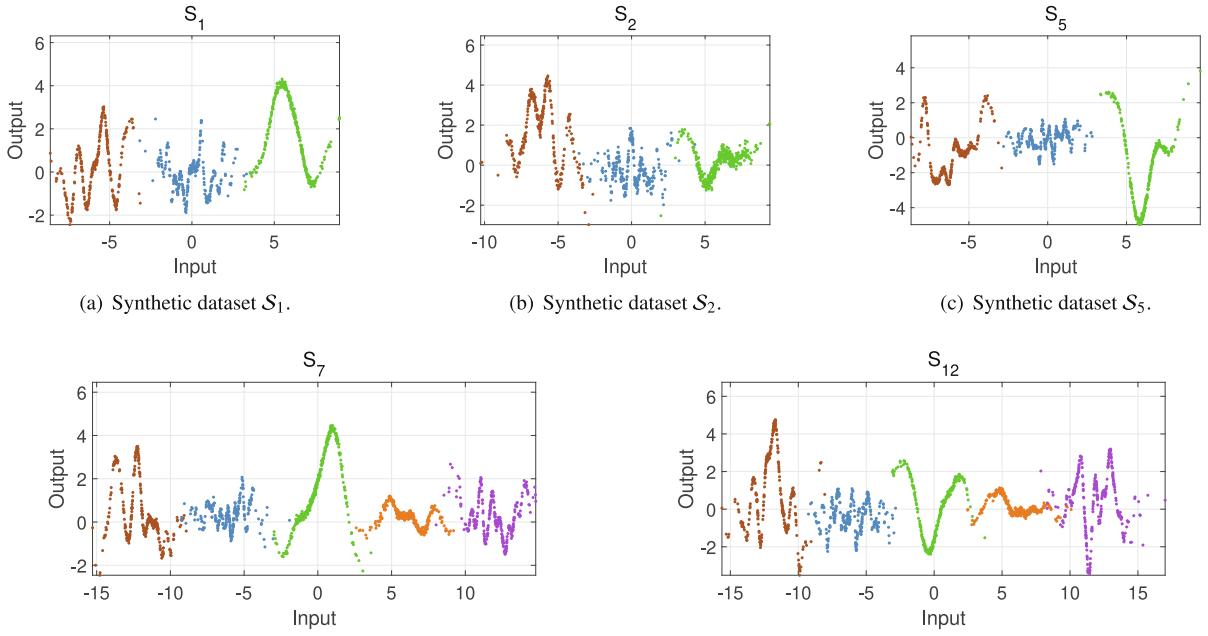
After generating these datasets, we add certain outliers to these datasets for testing the robustness of the proposed MRGP models. Suppose that the maximum of $\{y_i\}_{i=1}^N$ is ω, we randomly select $\eta N/3$ samples from the training set, and then add randomly generated noises from $(-v\omega,v\omega)$ to the corresponding targets. However, these modified points are not necessarily outliers, since it is possible that the random noise generated from $(-v\omega,v\omega)$ is close to 0. On the whole, we can use two parameters: outlier ratio η and outlier level v. Intuitively, there will be more outliers as we increase η, while the outliers will be more extreme as we increase v. We consider three outlier ratios: low, medium, high, corresponding to $\eta = 0.05,0.1,0.15$, respectively and three outlier levels: low, medium, high, corresponds to $v = 1.5,2.0,2.5$, respectively.

We also implement the competitive regression methods (based on GPML toolbox [50]) for comparison. The details about the competitive methods are summarized as follows:

(a) GP: Gaussian process with Gaussian noises.
(b) RGP (Laplace): Robust Gaussian process with Laplace noises.
(c) RGP (Student-t): Robust Gaussian process with student-t noises.
(d) MGP: Mixture of Gaussian processes with Gaussian noises.
(e) SVM: Support vector regression with Gaussian kernel.

Table 1
Parameters of basic datasets $\mathscr{S}_1$ and $\mathscr{S}_7$.

Dataset	Component	π_k	μ_k	σ_k^2	$\boldsymbol{\theta}_k$
$\mathscr{S}_1$	1	1/3	−6	1.5	[2.00, 3.33, 0.10]
	2	1/3	0	1.5	[0.75, 10.00, 0.10]
	3	1/3	6	1.5	[1.50, 1.25, 0.10]
$\mathscr{S}_7$	1	0.2	−12	1.5	[2.00, 3.33, 0.10]
	2	0.2	−6	1.5	[0.75, 10.00, 0.10]
	3	0.2	0	1.5	[1.50, 1.25, 0.10]
	4	0.2	6	1.5	[0.50, 2.50, 0.10]
	5	0.2	12	1.5	[1.50, 5.00, 0.10]



(a) Synthetic dataset $\mathcal{S}_1$. (b) Synthetic dataset $\mathcal{S}_2$. (c) Synthetic dataset $\mathcal{S}_5$.

(d) Synthetic dataset $\mathcal{S}_7$. (e) Synthetic dataset $\mathcal{S}_{12}$.

Fig. 6. Visualizations of synthetic datasets used in experiments: $\mathscr{S}_1, \mathscr{S}_2, \mathscr{S}_5, \mathscr{S}_7$ and $\mathscr{S}_{12}$. Different components are represented by different colors. $\mathscr{S}_1$ and $\mathscr{S}_7$ are basic datasets. $\mathscr{S}_2$ is more noisy than $\mathscr{S}_1$, $\mathscr{S}_5$ is a mildly overlapping dataset, and $\mathscr{S}_{12}$ is a heavily overlapping dataset.

(f) FNN: Feedforward neural network with three hidden layers containing 10, 10, 5 units, respectively.

For the fairest, we adopt the "two-stage" versions of GP, MGP, SVM, and FNN. That is, we firstly use a common outlier detection method to remove possible outliers from the training set, and then perform the corresponding algorithms on the improved training set.

Our proposed methods are the MRGP models with Laplace noises and Student-t noises through the parameter learning of the hard-cut EM algorithm, being referred to as MRGP (Laplace) and MRGP (Student-t), respectively. For all the MGP methods, the number of Gaussian processes is set to be the number of actual Gaussian processes in the dataset, *i.e.*, 3 for $\mathscr{S}_1 - \mathscr{S}_6$ and 5 for $\mathscr{S}_7 - \mathscr{S}_{12}$. The implementation is based on the GPML toolbox [50], and all experiments are conducted on a personal computer [Intel(R) Core(TM) i7-6700HQ CPU 2.60 GHz, 8G RAM].

We begin to fix outlier level v to be 2, and vary outlier ratio in $\{0.05, 0.10, 0.15\}$. For each dataset, we add outliers to the training set with low, medium and high ratios, then we test the prediction performances of all the methods. The performance is measured by Rooted Mean Square Error (RMSE), which is defined as

$$\text{RMSE} = \sqrt{\frac{1}{M}\sum_{j=1}^{M}(\hat{y}_j - \tilde{y}_j)^2},$$

where M is the size of test set, $\{\tilde{y}_j\}_{j=1}^M$ are ground-truth values, while $\{\hat{y}_j\}_{j=1}^M$ are the predicted results. Since the final results of the algorithms depend on the initializations of $\{z_i\}_{i=1}^N$ and $\{\boldsymbol{\theta}_k, \gamma_k\}_{k=1}^K$, different runs may lead to different results due to randomness. Therefore, we repeat the experiments for 10 times, and the average RMSEs of all the methods are listed in Table 2. To investigate the effect of the outlier level, we then fix the outlier ratio η to be 0.10, and vary outlier level v in $\{1.5, 2.5\}$. The average RMSEs of all the methods in this situation are listed in Table 3.

From these two tables, we have the following significant observations. First, all the robust models obtain better results than conventional Gaussian process based models, which demonstrates that Laplace or Student-t noises improve the robustness of the regression model. Second, two-stage methods usually achieve lower RMSEs in comparison with the original methods, but in certain cases, the process of removing outliers makes the results worse. One possible explanation is that it is very difficult to identify which samples are outliers, and thus the outlier detection algorithm is likely to falsely remove non-outliers or fail to identify outliers. Third, the adoption of a mixture structure usually improves the performance of the regression model because MGP is more flexible than GP. However, GP outperforms MGP occasionally, the reason is that in the learning process, each component has fewer samples and it is more easily to be affected by extreme out-

Table 2
The average RMSEs of our MRGP and competitive regression methods on 12 synthetic datasets over 10 trials, where the outlier level v is fixed to be 2 and the outlier ratio η varies in $\{0.05, 0.10, 0.15\}$, and the best results are in bold.

Outlier ratio	Method	$\mathscr{S}_1$	$\mathscr{S}_2$	$\mathscr{S}_3$	$\mathscr{S}_4$	$\mathscr{S}_5$	$\mathscr{S}_6$	$\mathscr{S}_7$	$\mathscr{S}_8$	$\mathscr{S}_9$	$\mathscr{S}_{10}$	$\mathscr{S}_{11}$	$\mathscr{S}_{12}$
Low ($\eta = 0.05$)	GP	0.5400	0.4520	0.4493	0.5846	0.4924	0.6082	0.5362	0.7158	0.3761	0.5596	0.5992	0.6276
	GP (TS)	0.5352	0.3326	0.3664	1.1176	0.3421	0.4792	0.8739	0.4638	0.2924	0.5734	0.3223	0.4397
	SVM	0.6340	0.5723	0.6128	0.8040	0.5132	0.7262	0.6830	1.0649	0.8271	0.9303	0.9521	0.9675
	SVM (TS)	0.6192	0.7468	0.6329	0.9790	0.5597	0.9149	0.7302	1.1217	0.7871	0.9526	0.8941	1.2704
	FNN	0.7206	0.7867	0.9111	0.9280	0.8937	1.0610	0.7764	1.0618	0.8158	0.9775	0.7570	1.0105
	FNN (TS)	0.6789	0.5527	0.5610	0.8304	0.4839	0.5518	0.7351	0.8649	0.5026	0.8682	1.0765	0.4729
	RGP (Laplace)	0.2951	0.3232	0.3511	0.3196	0.3409	0.4682	0.2842	0.4228	0.2573	0.3576	0.2232	0.4124
	RGP (Student-t)	0.2673	0.3404	0.3416	0.3175	0.4300	0.4944	0.3128	0.4479	0.2231	0.3732	0.2203	0.3696
	MGP	0.4256	0.4291	0.5794	0.5251	0.3841	0.5765	0.4980	0.8658	0.3597	0.5148	0.6279	0.5949
	MGP (TS)	0.5082	0.3050	0.2618	0.9272	0.2378	0.3934	0.4747	0.4019	0.2844	0.3960	0.2344	0.3426
	MRGP (Laplace)	**0.2337**	0.3067	0.2749	0.2254	0.1983	0.3638	0.2639	0.3806	0.2259	0.2873	0.1907	**0.3180**
	MRGP (Student-t)	0.2423	**0.3042**	**0.2579**	**0.2175**	**0.1913**	**0.3521**	**0.2448**	**0.3346**	**0.1716**	**0.2839**	**0.1843**	0.3498
Medium ($\eta = 0.10$)	GP	0.7016	0.6503	0.6376	0.7378	0.5475	0.7241	0.4490	0.6233	0.4854	0.6545	0.5435	0.6198
	GP (TS)	0.3921	0.4050	0.4421	1.1092	0.9433	0.5273	0.7827	0.5054	0.3799	0.5747	0.3236	0.4574
	SVM	0.5497	0.6017	0.6469	0.7451	0.5128	0.8483	0.6839	1.0515	0.8651	0.9253	0.9332	1.2003
	SVM (TS)	0.6559	0.7621	0.6909	0.9613	0.6052	0.8823	0.7979	1.0965	0.7544	0.9060	0.9030	1.0341
	FNN	0.7998	0.8114	0.9290	1.0415	0.9035	1.0000	0.6563	1.0923	0.8244	0.9356	1.0852	1.0427
	FNN (TS)	0.5656	0.8313	0.4430	1.2403	0.6054	0.5850	0.3775	1.3183	0.8635	0.6244	0.5952	1.1725
	RGP (Laplace)	0.2706	0.3657	0.3815	0.3620	0.3551	0.4985	0.3066	0.4280	0.3028	0.4442	0.2491	0.4260
	RGP (Student-t)	0.2772	0.3483	0.3810	0.3322	0.4151	0.5076	0.3081	0.4983	0.2797	0.3729	0.2510	0.3718
	MGP	0.6027	0.6286	0.5859	0.7606	0.5319	0.6762	0.4662	0.5556	0.5236	0.7324	0.5124	0.6475
	MGP (TS)	0.3319	0.3633	0.3734	1.0008	0.4990	0.4814	0.3042	0.3754	0.3088	0.3588	0.3394	0.4319
	MRGP (Laplace)	**0.2345**	0.3318	0.2847	0.3005	0.2256	0.3876	0.2679	0.3565	0.2276	0.3295	0.2275	0.3517
	MRGP (Student-t)	0.2386	**0.3066**	**0.2487**	**0.2206**	**0.1905**	**0.3519**	**0.2557**	**0.3334**	**0.1756**	**0.2856**	**0.1816**	**0.3483**
High ($\eta = 0.15$)	GP	0.7638	0.7572	0.6772	0.5284	0.6902	0.7660	0.6386	0.9249	0.6012	0.6628	0.6867	0.7110
	GP (TS)	0.6405	0.4680	0.5191	0.9625	0.4749	0.5387	0.4655	0.5018	0.4118	0.5728	0.4870	0.5194
	SVM	0.6512	0.7286	0.6013	0.7446	0.5717	0.9725	0.6919	1.0509	0.7539	0.9679	0.9028	0.8428
	SVM (TS)	0.6774	0.5645	0.6914	1.0055	0.5044	0.7471	0.7239	1.0886	0.7939	0.9586	0.9126	1.0185
	FNN	0.7828	1.0117	1.0115	0.9070	0.8758	0.8234	0.8956	1.2199	0.7831	1.0207	1.1145	1.3124
	FNN (TS)	0.6940	0.8043	1.3943	0.9322	0.6360	0.5590	0.4354	1.0600	0.7791	0.7547	0.6825	0.7442
	RGP (Laplace)	0.4906	0.4616	0.4324	0.3582	0.5282	0.5017	0.3732	0.4730	0.3446	0.5010	0.3987	0.4406
	RGP (Student-t)	0.3534	0.3631	0.3600	0.3154	0.4480	0.5281	0.2961	0.4591	0.2516	0.3972	0.2571	0.4034
	MGP	0.8257	0.7689	0.6554	0.7423	0.6015	0.7788	0.6346	0.9758	0.6540	0.6620	0.7462	0.7363
	MGP (TS)	0.6007	0.4221	0.8251	0.8401	0.4597	0.4720	0.3438	0.4039	0.3982	0.6399	0.6649	0.4876
	MRGP (Laplace)	0.2844	0.3947	0.3573	0.3532	0.2459	0.4042	0.3034	**0.3751**	0.2480	0.4402	0.3555	**0.3943**
	MRGP (Student-t)	**0.2412**	**0.3192**	**0.2664**	**0.2130**	**0.3526**	**0.3531**	**0.2504**	0.3815	**0.1733**	**0.3892**	**0.1814**	0.4388

Table 3
The average RMSEs of our MRGP and competitive regression methods on 12 synthetic datasets over 10 trials, where the outlier level v is fixed to be 0.10 and the outlier ratio η varies in $\{1.5, 2.5\}$, and the best results are in bold.

Outlier level	Method	$\mathscr{S}_1$	$\mathscr{S}_2$	$\mathscr{S}_3$	$\mathscr{S}_4$	$\mathscr{S}_5$	$\mathscr{S}_6$	$\mathscr{S}_7$	$\mathscr{S}_8$	$\mathscr{S}_9$	$\mathscr{S}_{10}$	$\mathscr{S}_{11}$	$\mathscr{S}_{12}$
Low ($v = 1.5$)	GP	0.5815	0.9124	0.5052	0.7265	0.5381	0.6693	0.5401	0.5714	0.4223	0.5323	0.4586	0.6529
	GP (TS)	0.3554	0.4678	0.4806	1.1282	0.4302	0.9689	0.9045	0.5263	0.3263	0.5163	0.4693	0.4512
	SVM	0.5843	0.7907	0.6400	0.8825	0.5117	0.8211	0.6768	1.0485	0.8682	0.9242	0.7092	1.0971
	SVM (TS)	0.6052	0.7901	0.6600	1.0219	0.5726	0.9329	0.7735	1.0686	0.8484	0.9251	0.9031	1.2059
	FNN	0.7795	1.0624	1.0836	0.8211	0.8548	1.0864	0.8595	0.8541	0.9031	0.9287	1.0611	0.9513
	FNN (TS)	0.6085	0.6892	0.8602	0.8547	0.5002	0.6120	0.5915	1.1037	0.7963	0.9556	1.1165	0.7909
	RGP (Laplace)	0.2772	0.3752	0.3822	0.3329	0.3503	0.4865	0.3332	0.4338	0.2727	0.3532	0.2407	0.4248
	RGP (Student-t)	0.3470	0.3427	0.3780	0.3242	0.4209	0.4951	0.2969	0.4253	0.2226	0.3621	0.2427	0.3807
	MGP	0.4734	0.8827	0.5617	0.6385	0.3800	0.6549	0.4887	0.5394	0.3870	0.5980	0.4386	0.6850
	MGP (TS)	0.3577	0.4963	0.5293	1.0362	0.3679	1.1550	0.6118	0.4723	0.2817	0.4068	0.4312	0.4815
	MRGP (Laplace)	0.2454	0.3295	0.3000	0.2994	0.2096	0.3960	0.2641	0.3738	0.2011	0.3012	0.2265	0.3562
	MRGP (Student-t)	**0.2308**	**0.3069**	**0.2486**	**0.2182**	**0.1911**	**0.3568**	**0.2530**	**0.3366**	**0.1738**	**0.2889**	**0.1766**	**0.3521**
High ($v = 2.5$)	GP	0.8333	0.8952	0.6695	0.9217	0.7114	0.8344	0.6561	0.9250	0.5149	0.7747	0.6471	0.7967
	GP (TS)	0.3935	0.3776	0.4518	1.1000	0.3655	0.6355	0.8009	0.4960	0.3593	0.6713	0.3841	0.5105
	SVM	0.5737	0.6755	0.6331	0.8132	0.5771	0.7490	0.6864	1.0585	0.8207	0.8996	0.8179	1.1225
	SVM (TS)	0.5372	0.7552	0.5887	0.9769	0.5172	0.9883	0.8181	1.1089	0.8510	0.9610	0.8921	1.0987
	FNN	1.3731	0.8585	1.0517	1.0505	0.9015	2.8179	0.7714	1.2513	0.8520	0.8899	0.9924	1.2133
	FNN (TS)	0.5754	0.8092	1.0313	1.0387	0.5670	0.7117	0.6830	1.0960	0.8531	0.6736	1.1491	0.8296
	RGP (Laplace)	0.2702	0.3658	0.3746	0.4047	0.5446	0.4743	0.3438	0.4437	0.3086	0.4154	0.2936	0.4544
	RGP (Student-t)	0.3427	0.3486	0.4267	0.3153	0.2953	0.5094	0.2962	0.4442	0.2380	0.3867	0.2554	0.4213
	MGP	0.7807	0.9288	0.7934	0.9149	0.7557	0.8304	0.6083	1.1960	0.7060	0.8164	0.7260	0.6668
	MGP (TS)	0.3843	0.3946	0.3859	0.9135	0.2755	0.6007	0.3322	0.3667	0.3149	0.7110	0.3192	0.4317
	MRGP (Laplace)	**0.2200**	0.3391	0.2865	0.3684	0.3670	0.3947	0.3311	0.3903	0.2251	0.3505	0.2504	**0.3585**
	MRGP (Student-t)	0.2528	**0.3108**	**0.2502**	**0.2229**	**0.1908**	**0.3515**	**0.3193**	**0.3358**	**0.1735**	**0.2841**	**0.1768**	0.3640

liers. Finally, our proposed methods obtain the best results in almost all cases, which demonstrates the effectiveness of the proposed models. We can observe that MRGP (Student-t) outperforms MRGP (Laplace) almost all, but there are a few cases that MRGP (Laplace) outperforms MRGP (Student-t). Therefore, we can not conclude which model is generally better, and this may depend

Table 4
The average classification accuracy rates of MGP, MRGP (Laplace) and MGRP (Student-t) on synthetic datasets under various settings.

Setting	Method	$\mathscr{S}_1$	$\mathscr{S}_2$	$\mathscr{S}_3$	$\mathscr{S}_4$	$\mathscr{S}_5$	$\mathscr{S}_6$	$\mathscr{S}_7$	$\mathscr{S}_8$	$\mathscr{S}_9$	$\mathscr{S}_{10}$	$\mathscr{S}_{11}$	$\mathscr{S}_{12}$
$\eta = 0.05, \nu = 2.0$	MGP	99.3%	99.0%	99.3%	98.7%	100.0%	97.0%	98.8%	98.6%	99.0%	98.8%	99.2%	98.2%
	MRGP (Laplace)	99.3%	99.0%	99.3%	98.7%	100.0%	97.0%	99.0%	98.8%	99.2%	98.8%	99.2%	98.0%
	MRGP (Student-t)	97.7%	98.3%	98.3%	99.3%	100.0%	97.0%	98.4%	99.2%	97.4%	99.6%	98.8%	96.2%
$\eta = 0.10, \nu = 2.0$	MGP	99.3%	98.7%	99.3%	98.3%	100.0%	96.7%	98.8%	74.8%	98.8%	99.4%	99.2%	98.0%
	MRGP (Laplace)	99.3%	99.0%	99.3%	99.0%	100.0%	97.3%	98.8%	98.8%	99.0%	99.2%	99.2%	97.8%
	MRGP (Student-t)	97.7%	97.7%	99.0%	99.3%	100.0%	97.7%	98.8%	99.2%	98.0%	99.8%	99.0%	97.4%
$\eta = 0.15, \nu = 2.0$	MGP	99.3%	98.7%	99.3%	98.7%	100.0%	97.0%	98.6%	99.0%	99.0%	99.4%	99.2%	97.8%
	MRGP (Laplace)	99.3%	99.0%	99.3%	98.7%	100.0%	97.7%	98.6%	97.7%	99.0%	99.4%	99.2%	98.0%
	MRGP (Student-t)	98.0%	98.3%	99.0%	99.0%	100.0%	96.7%	98.2%	99.2%	98.4%	96.8%	99.0%	96.8%
$\eta = 0.10, \nu = 1.5$	MGP	99.3%	99.0%	99.3%	98.7%	100.0%	97.0%	98.8%	99.0%	99.0%	99.4%	99.2%	98.2%
	MRGP (Laplace)	99.3%	99.0%	99.3%	98.7%	100.0%	97.0%	98.8%	98.8%	99.2%	99.2%	99.2%	98.0%
	MRGP (Student-t)	97.7%	98.3%	99.0%	99.3%	100.0%	97.7%	98.4%	99.2%	98.0%	99.6%	99.2%	97.4%
$\eta = 0.10, \nu = 2.5$	MGP	99.3%	98.7%	99.3%	98.7%	99.7%	95.3%	98.8%	99.0%	98.8%	98.8%	99.2%	96.8%
	MRGP (Laplace)	99.3%	99.0%	99.3%	98.7%	100.0%	97.7%	97.4%	99.0%	98.6%	98.0%	99.2%	98.0%
	MRGP (Student-t)	97.7%	97.7%	99.0%	99.3%	100.0%	97.7%	97.8%	99.2%	98.4%	99.8%	99.2%	97.4%

on the dataset. Empirically, we find out that when the outlier ratio is relatively low, MRGP (Laplace) and MRGP (Student-t) lead to similar results, but as the outlier ratio increases, MRGP(Student-t) generally becomes better than MRGP (Laplace).

The hard-cut EM algorithm is an approximation of the original EM algorithm. Specifically, in the E-step, the posterior distributions of latent variables $\{z_i\}_{i=1}^N$ are approximated by deterministic hard-cut allocations. Such an approximation may cause errors in the learning process. We evaluate the quality of approximation empirically. Since the original EM algorithm for MGP consists of exponentially many summation terms and is prohibitively time-consuming, it is intractable to run the original EM algorithm and compare the performances with the results of the hard-cut EM algorithm. Nevertheless, we can calculate the Classification Accuracy Rates (CARs) to validate the effectiveness of the hard-cut EM algorithm. In synthetic datasets, we have the ground-truth component labels, so we can compare the estimated component labels $\{\hat{z}_i\}_{i=1}^N$ by hard-cut EM algorithm with the ground-truth labels $\{z_i\}_{i=1}^N$. Formally, the CAR is defined as

$$\text{CAR} = \max_{\xi \in \Pi_K} \frac{1}{N} \sum_{i=1}^{N} \mathbf{bI}(z_i = \xi(\hat{z}_i)).$$

Here, Π_K denotes the set of K-permutations, and the permutation ξ is employed to account for the label switching problem. Intuitively, CAR measures how well we cluster the observations into correct components. The results are shown in Table 4. Since MGP is also learned by the hard-cit EM algorithm, we also include the CARs of MGP for comparison. From this table, we find that in terms of CARs, the hard-cut EM algorithm based MGP and MRGP methods perform well on all the synthetic datasets, under all settings. This observation demonstrates that the hard-cut EM algorithm is effective and the quality of approximation is satisfying. We also find the CARs are relatively high on $\mathscr{S}_5, \mathscr{S}_{11}$ and relatively low on $\mathscr{S}_6, \mathscr{S}_{12}$. This observation coincides with the fact that $\mathscr{S}_5, \mathscr{S}_{11}$ are mildly overlapped thus easier to cluster the samples, while $\mathscr{S}_6, \mathscr{S}_{12}$ are heavily overlapped thus harder to cluster the observations correctly. Besides, we find that on $\mathscr{S}_6$ and $\mathscr{S}_{12}$, the classification accuracy rates of MGP drop significantly ($97.0\% \rightarrow 95.3\%$ and $98.2\% \rightarrow 96.8\%$) as we increase the outlier ratio and outlier level from $\eta = 0.05, \nu = 2.0$ to $\eta = 0.10, \nu = 2.5$. On the other hand, the classification accuracy rates of MRGP (Laplace) and MRGP (Student-t) are not sensitive to the outlier ratios and outlier levels. This further demonstrates that MRGP models are more robust than the MGP model.

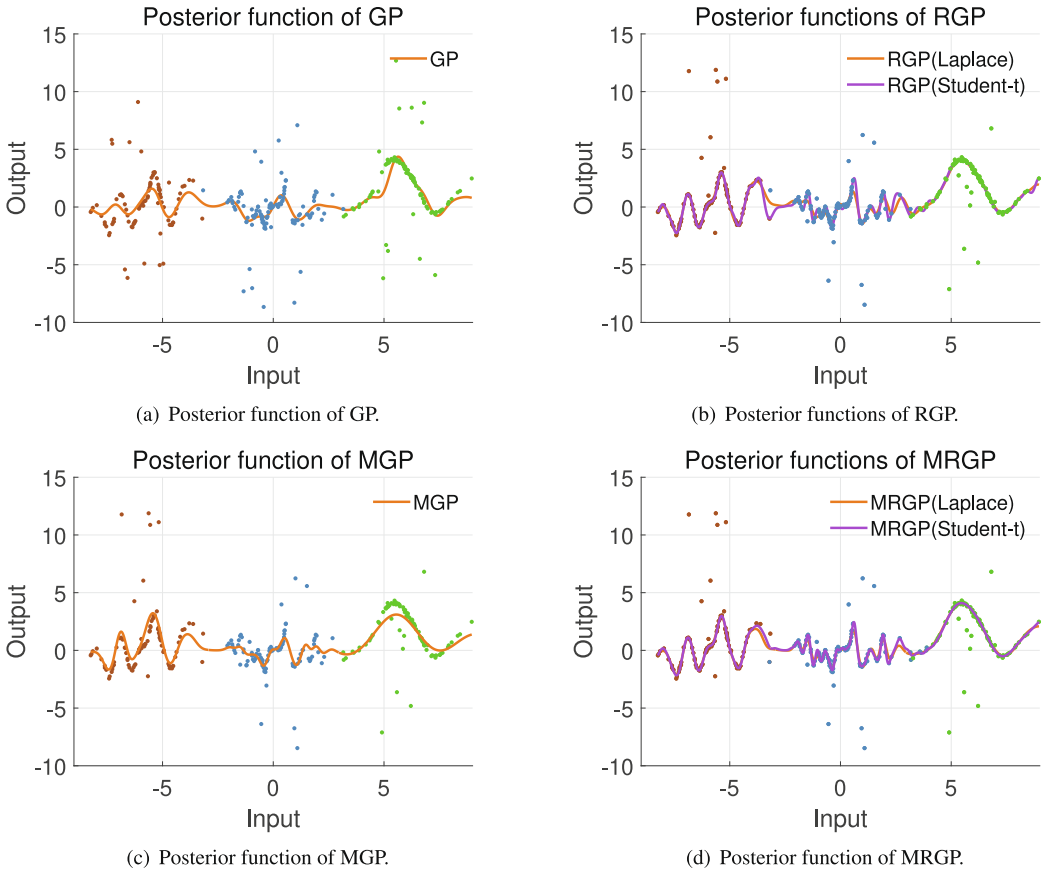
For further comparison, the posterior curves of these methods on $\mathscr{S}_1$ with $\eta = 0.10$ and $\nu = 2.0$ are shown in Fig. 7. It can be observed from Fig. 7(a) that the posterior curve of GP is stable and severely affected by outliers. By taking heavy-tailed distributions as the noises distributions instead of Gaussian distributions, the posterior curves in Fig. 7(b) are less affected by outliers. However, these two curves can not capture the non-stationary trends. The posterior function of RGP (Student-t) oscillates acutely in the third component, while the posterior function of RGP (Laplace) is too smooth in the second component. From Fig. 7(c), we can find that the posterior function of MGP is not stationary, but in each component, the function suffers from outliers severely. Finally, from Fig. 7(d) we can observe that the posterior functions of MRGP (Laplace) and MRGP (Student-t) perfectly fit the trends without being affected by outliers. Therefore, these detailed results further demonstrate that our proposed methods are able to model non-stationary sources and insensitive to outliers.

5.2. On real-world data

We further evaluate our proposed methods on three real-world datasets.

- **Boston housing**. The Boston housing dataset[1] [51] contains information collected by the U.S Census Service concerning housing in the area of Boston Mass. There are 506 samples in total, and each case of the dataset consists of 14 attributes including crime rate, number of rooms, accessibility to radial highways, prices and so on. The aim is to predict the price of a house given other 13 attributes. We randomly choose 250 samples for training and the rest 256 samples are used for testing.
- **Electricity**. The individual household electric power consumption dataset [2] contains 2075259 measurements gathered in a house located in Sceaux (7 km of Paris, France) between December 2006 and November 2010 (47 months). This dataset records six attributes, and we use the global minute-averaged active power (in kilowatt) in our experiment. We further average the records in one day to obtain daily-averaged active power, which results in 1433 valid records. We randomly choose 500 samples for training and 933 samples for testing.
- **Weather**. The PM2.5 dataset [3] [52] collects hourly meteorological records of five Chinese cities. In this experiment, we use the temperature records of Chengdu in 2015. Similarly, we average the records in one day to obtain daily-averaged temperatures.

[1] https://www.cs.toronto.edu/delve/data/boston/bostonDetail.html.
[2] https://archive.ics.uci.edu/ml/datasets/individual+household+electric+power+consumption.
[3] https://archive.ics.uci.edu/ml/datasets/PM2.5+Data+of+Five+Chinese+Cities.



(a) Posterior function of GP.

(b) Posterior functions of RGP.

(c) Posterior function of MGP.

(d) Posterior function of MRGP.

Fig. 7. The posterior curves of GP, RGP (Laplace), RGP (Student-t), MGP, MRGP (Laplace) and MRGP (Student-t) on synthetic dataset $\mathscr{S}_1$ with outlier ratio $\eta = 0.10$ and outlier level $v = 2.0$.

Among the 365 samples, we randomly choose 200 samples for training and 165 samples for testing.

However, it is unreasonable to contain outliers in the *test* set since one only concerns about estimating the underlying regression function and it is unnecessary to predict outliers in real applications. In real applications, one only concerns about estimating the underlying regression function and it is unnecessary to predict outliers. Including outliers in the test set will introduce bias in the evaluation metrics. In practice, we first use Grubbs's outlier detection method to detect the outliers of the *test* set and then move them into the training set.

We compare our proposed methods with GP, SVM, FNN, RGP (Laplace), RGP (Student-t) and MGP(hard-cut) as in experiments on synthetic datasets. For the mixture models, since we do not know the correct number of components a prior, we set the num-

Table 5
The average RMSEs and running times (seconds) of our MRGP and competitive regression methods on Boston dataset and electricity dataset. The best results are in bold.

K	Method	Boston housing		Electricity		Weather	
		RMSE	Time	RMSE	Time	RMSE	Time
1	GP	3.0570 ± 0.0000	0.55 ± 0.07	0.1309 ± 0.0000	3.41 ± 0.01	1.9934 ± 0.0000	0.65 ± 0.35
	SVM	3.6394 ± 0.1033	0.18 ± 0.14	0.1332 ± 0.0007	0.16 ± 0.10	2.0451 ± 0.0218	0.22 ± 0.18
	FNN	4.2566 ± 0.6081	0.62 ± 0.44	0.1367 ± 0.0053	0.57 ± 0.43	2.0868 ± 0.1622	0.81 ± 0.41
	RGP (Laplace)	2.8749 ± 0.0000	12.71 ± 0.17	0.1334 ± 0.0000	65.52 ± 0.03	2.0431 ± 0.0000	15.52 ± 0.31
	RGP (Student-t)	2.8420 ± 0.0000	22.43 ± 0.21	0.1292 ± 0.0000	71.21 ± 0.05	2.1785 ± 0.0000	9.62 ± 0.15
2	MGP	2.9898 ± 0.0000	1.85 ± 0.12	0.1329 ± 0.0001	4.30 ± 0.53	2.2228 ± 0.0000	2.14 ± 0.24
	MGP (Laplace)	2.8246 ± 0.0114	59.59 ± 0.51	0.1323 ± 0.0008	112.64 ± 0.19	1.9469 ± 0.1080	74.02 ± 9.09
	MGP (Student-t)	$\mathbf{2.6062 \pm 0.0000}$	55.26 ± 0.44	0.1318 ± 0.0001	54.92 ± 5.09	2.1716 ± 0.0000	34.40 ± 2.52
3	MGP	3.0862 ± 0.0804	1.98 ± 0.72	0.1326 ± 0.0003	3.92 ± 3.03	1.7108 ± 0.0269	2.07 ± 0.19
	MGP (Laplace)	2.8144 ± 0.0932	53.41 ± 22.30	$\mathbf{0.1289 \pm 0.0000}$	73.82 ± 51.84	$\mathbf{1.6705 \pm 0.0278}$	83.95 ± 3.70
	MGP (Student-t)	2.7502 ± 0.1015	50.21 ± 17.06	0.1294 ± 0.0018	99.46 ± 1.19	1.6726 ± 0.0241	24.42 ± 3.19
4	MGP	3.1318 ± 0.1181	2.7542	0.1345 ± 0.0003	4.33 ± 0.67	2.0571 ± 0.0835	4.29 ± 1.38
	MGP (Laplace)	2.8786 ± 0.1337	101.52 ± 27.86	0.1305 ± 0.0003	148.55 ± 23.73	1.7096 ± 0.0368	90.89 ± 36.94
	MGP (Student-t)	2.7722 ± 0.1209	70.99 ± 24.70	0.1295 ± 0.0008	54.29 ± 13.69	1.7724 ± 0.0390	41.95 ± 13.25
5	MGP	3.1315 ± 0.0423	3.36 ± 0.55	0.1345 ± 0.0004	3.43 ± 0.88	1.8990 ± 0.0306	5.31 ± 0.75
	MGP (Laplace)	2.8936 ± 0.0834	100.52 ± 32.59	0.1337 ± 0.0001	86.97 ± 55.25	1.7317 ± 0.0515	140.68 ± 12.30
	MGP (Student-t)	2.8590 ± 0.1405	68.17 ± 12.73	0.1314 ± 0.0007	66.05 ± 3.10	1.9688 ± 0.1120	57.88 ± 3.16

ber of components in $\{2, 3, 4, 5\}$ respectively. For each dataset, we run each method for 10 times, and list the average RMSE (together with standard deviations) and running time in Table 5. From this table, we can find out MRGP (Laplace) and MRGP (Student-t) outperform the other methods on the prediction performance. Specifically, on Boston housing dataset MRGP (Student-t) with $K = 2$ achieves the lowest RMSE, and MRGP (Laplace) with $K = 3$ obtains the best results on Electricity and Weather datasets. This indicates both MRGP (Laplace) and MRGP (Student-t) have their own advantages and it is generally difficult to determine which model is more suitable for the task in hand. The choice of K is also subtle. From Table 5, we can see K influence the prediction performances severely. The choice of K also depends on the dataset. On Electricity and Weather datasets, 2 components are far from enough to fit the data, while too many components ($i.e.$,$K = 5$) also lead to large errors due to over-fitting. On the Boston housing dataset, increasing K almost always leads to larger errors. One possible explanation is the inputs lie in a 13-dimensional space, but we only have 250 training samples. Therefore, dividing these samples to several components will cause difficulty for learning in each individual component since the samples are too sparse in the 13-dimensional space. We can observe that the proper number of components for different mixture models are almost the same: $K = 2$ for Boston housing dataset and $K = 3$ for Electricity and Weather dataset. Thus, how to set K relies heavily on the dataset rather than the mixture model. Finally, we can also find out that our proposed methods are much more time-consuming than the other methods because. Therefore, the proposed methods may not be well-suited for real-time tasks.

6. Conclusion and discussion

We have established the mixture of robust Gaussian processes (MRGP) by adopting Laplace or student-t noises with heavy-tailed property into Gaussian processes. In such a way, the MRGP model has the ability to model non-stationary temporal data effectively and also to be insensitive to outliers. The hard-cut EM algorithm is further developed for the MRGP model with the help of the variational bounding method to make the marginal likelihood of the robust Gaussian process be tractable in the ML solving process. It is demonstrated by the experimental results on both synthetic and real-world datasets that our proposed MRGP methods are much more effective and robust than the competitive nonlinear regression models.

How to set the number of components adaptively in real applications is an interesting direction. We can further develop automated model selection methods for the mixture of robust Gaussian processes. In fact, split-and-merge EM algorithm [53,54], rival penalized EM algorithm [55], reversible jump MCMC [56,57], entropy penalty [58–60] and Bayesian Ying-Yang (BYY) harmony learning [61–65] have been shown to be effective for the automated mode selection on mixture models. However, these methods are not so easy to apply to the mixture of Gaussian processes since the samples are not independent and highly correlated. A synchronously balancing criterion [66] has been proposed for model selection of MGP, but its penalty coefficient is still difficult to determine. The automatic model selection for the mixture of robust Gaussian processes is certainly a potential future direction. It is also promising to reduce the computational cost by introducing inducing points to our proposed model. Using inducing points [47,67,43] in Gaussian processes can improve the computational complexity significantly, and the extension to a mixture of Gaussian processes has been studied in [16]. However, sparse Gaussian processes usually have lower prediction accuracy, and how to balance the trade-off between performance and computational time generally depends on the particular task in hand. The extensions of the proposed model to classification task [33] or state-space model [68–70] is also promising. Finally, further incorporating domain-specific priors in the model [71] is a potential direction.

CRediT authorship contribution statement

Tao Li: Conceptualization, Investigation, Methodology, Software, Validation, Visualization, Writing - original draft. **Di Wu:** Conceptualization, Formal Analysis, Investigation, Methodology, Writing - review & editing. **Jinwen Ma:** Conceptualization, Formal Analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Supervision, Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by the National Key R&D Program of China (2018YFC0808305).

References

[1] Carl Edward Rasmussen, Gaussian processes in machine learning, in: Summer School on Machine Learning, Springer, 2003, pp. 63–71.
[2] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, Geoffrey E Hinton, et al., Adaptive mixtures of local experts, Neural Computation 3 (1) (1991) 79–87.
[3] Lei Xu, Michael I. Jordan, Geoffrey E. Hinton, An alternative model for mixtures of experts, in: Advances in Neural Information Processing Systems, 1995, pp. 633–640..
[4] Volker Tresp, Mixtures of gaussian processes, in: Advances in Neural Information Processing Systems, 2001, pp. 654–660..
[5] Jian Qing Shi, Roderick Murray-Smith, D. Michael Titterington, Hierarchical Gaussian process mixtures for regression, Statistics and Computing 15 (1) (2005) 31–41..
[6] Edward Meeds, Simon Osindero, An alternative infinite mixture of gaussian process experts, in: Advances in Neural Information Processing Systems, 2006, pp. 883–890..
[7] Oliver Stegle, Sebastian V. Fallert, David J.C. MacKay, Søren Brage, Gaussian process robust regression for noisy heart rate data, IEEE Transactions on Biomedical Engineering 55 (9) (2008) 2143–2151.
[8] Pasi Jylänki, Jarno Vanhatalo, Aki Vehtari, Robust gaussian process regression with a student-t likelihood, Journal of Machine Learning Research 12 (Nov) (2011) 3227–3257..
[9] Rishik Ranjan, Biao Huang, Alireza Fatehi, Robust gaussian process modeling using em algorithm, Journal of Process Control 42 (2016) 125–136.
[10] Thomas P. Minka, Expectation propagation for approximate bayesian inference, in: Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers Inc., 2001, pp. 362–369..
[11] Luke Tierney, Joseph B Kadane, Accurate approximations for posterior moments and marginal densities, Journal of the American Statistical Association 81 (393) (1986) 82–86.
[12] Mark N Gibbs, David JC MacKay, Variational gaussian process classifiers, IEEE Transactions on Neural Networks 11 (6) (2000) 1458–1464.
[13] Carl E. Rasmussen, Zoubin Ghahramani, Infinite mixtures of gaussian process experts, in: Advances in Neural Information Processing Systems, 2002, pp. 881–888..
[14] Yan Yang, Jinwen Ma, An efficient em approach to parameter learning of the mixture of gaussian processes, in: International Symposium on Neural Networks, Springer, 2011, pp. 165–174..
[15] Ziyi Chen, Jinwen Ma, Yatong Zhou, A precise hard-cut em algorithm for mixtures of gaussian processes, in: International Conference on Intelligent Computing, Springer, 2014, pp. 68–75..
[16] Ziyi Chen, Jinwen Ma, The hard-cut em algorithm for mixture of sparse gaussian processes, in: International Conference on Intelligent Computing, Springer, 2015, pp. 13–24..
[17] Di Wu, Ziyi Chen, Jinwen Ma, An mcmc based em algorithm for mixtures of gaussian processes, in: International Symposium on Neural Networks, Springer, 2015, pp. 327–334..
[18] Wu. Di, Jinwen Ma, An effective em algorithm for mixtures of gaussian processes via the mcmc sampling and approximation, Neurocomputing 331 (2019) 366–374.

[19] Arthur P Dempster, Nan M Laird, Donald B Rubin, Maximum likelihood from incomplete data via the em algorithm, Journal of the Royal Statistical Society: Series B (Methodological) 39 (1) (1977) 1–22.
[20] Geoffrey McLachlan, Thriyambakam Krishnan, The EM Algorithm and Extensions, vol. 382, John Wiley & Sons, 2007.
[21] David M Blei, Alp Kucukelbir, Jon D McAuliffe, Variational inference: A review for statisticians, Journal of the American Statistical Association 112 (518) (2017) 859–877.
[22] Steve Brooks, Andrew Gelman, Galin Jones, Xiao-Li Meng, Handbook of Markov Chain Monte Carlo, CRC Press, 2011.
[23] Stuart Geman, Donald Geman, Stochastic relaxation, gibbs distributions, and the bayesian restoration of images, IEEE Transactions on Pattern Analysis and Machine Intelligence 6 (1984) 721–741.
[24] Chao Yuan, Claus Neubauer, Variational mixture of gaussian process experts, in: Advances in Neural Information Processing Systems, 2009, pp. 1897–1904..
[25] Shiliang Sun, Xu. Xin, Variational inference for infinite mixtures of gaussian processes with applications to traffic flow prediction, IEEE Transactions on Intelligent Transportation Systems 12 (2) (2011) 466–475.
[26] Trung Nguyen, Edwin Bonilla, Fast allocation of gaussian process experts, in: International Conference on Machine Learning, 2014, pp. 145–153.
[27] Wu. Di, Jinwen Ma, A two-layer mixture model of gaussian process functional regressions and its mcmc em algorithm, IEEE Transactions on Neural Networks and Learning Systems 99 (2018) 1–11.
[28] James L Powell, Least absolute deviations estimation for the censored regression model, Journal of Econometrics 25 (3) (1984) 303–325.
[29] Tsung I Lin, Jack C Lee, Wan J Hsieh, Robust mixture modeling using the skew t distribution, Statistics and Computing 17 (2) (2007) 81–92.
[30] Tsung-I Lin, Robust mixture modeling using multivariate skew t distributions, Statistics and Computing 20 (3) (2010) 343–356.
[31] Faicel Chamroukhi, Skew t mixture of experts, Neurocomputing 266 (2017) 390–408.
[32] Jarno Vanhatalo, Pasi Jylänki, Aki Vehtari, Gaussian process regression with student-t likelihood, in: Advances in Neural Information Processing Systems, 2009, pp. 1910–1918..
[33] Hannes Nickisch, Carl Edward Rasmussen, Approximations for binary gaussian process classification, Journal of Machine Learning Research 9 (Oct) (2008) 2035–2078..
[34] Jason Palmer, Kenneth Kreutz-Delgado, Bhaskar D. Rao, David P. Wipf, Variational em algorithms for non-gaussian latent variable models, in: Advances in Neural Information Processing Systems, 2006, pp. 1059–1066..
[35] Hannes Nickisch, Matthias W. Seeger, Convex variational bayesian inference for large scale generalized linear models, in: Proceedings of the 26th Annual International Conference on Machine Learning, ACM, 2009, pp. 761–768..
[36] Matthias W Seeger, Hannes Nickisch, Large scale bayesian inference and experimental design for sparse linear models, SIAM Journal on Imaging Sciences 4 (1) (2011) 166–199.
[37] Ying Sun, Prabhu Babu, Daniel P Palomar, Majorization-minimization algorithms in signal processing, communications, and machine learning, IEEE Transactions on Signal Processing 65 (3) (2016) 794–816.
[38] Andreas Damianou, Neil Lawrence, Deep gaussian processes, in: Artificial Intelligence and Statistics, 2013, pp. 207–215..
[39] Hugh Salimbeni, Marc Deisenroth, Doubly stochastic variational inference for deep gaussian processes, in: Advances in Neural Information Processing Systems, 2017, pp. 4588–4599..
[40] Marton Havasi, José Miguel Hernández-Lobato, Juan José Murillo-Fuentes, Inference in deep gaussian processes using stochastic gradient hamiltonian monte carlo, in: Advances in Neural Information Processing Systems, 2018, pp. 7506–7516..
[41] Neil D. Lawrence, Gaussian process latent variable models for visualisation of high dimensional data, in: Advances in Neural Information Processing Systems, 2004, pages 329–336..
[42] Edward Snelson, Zoubin Ghahramani, Carl E. Rasmussen, Warped gaussian processes, in: Advances in Neural Information Processing Systems, 2004, pp. 337–344..
[43] Michalis Titsias, Variational learning of inducing variables in sparse gaussian processes, in: Artificial Intelligence and Statistics, 2009, pp. 567–574..
[44] James Hensman, Nicolò Fusi, Neil D Lawrence, Gaussian processes for big data, in: Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence, 2013, pp. 282–290.
[45] James Hensman, Alexander Matthews, Zoubin Ghahramani, Scalable variational gaussian process classification, 2015..
[46] Alexander Graeme de Garis Matthews, Scalable Gaussian process inference using variational methods, PhD thesis, University of Cambridge, 2017..
[47] Joaquin Quiñonero-Candela, Carl Edward Rasmussen, A unifying view of sparse approximate gaussian process regression, Journal of Machine Learning Research 6 (2005) (Dec) 1939–1959..
[48] Haitao Liu, Yew-Soon Ong, Xiaobo Shen, Jianfei Cai, When gaussian process meets big data: A review of scalable gps, IEEE Transactions on Neural Networks and Learning Systems (2020).
[49] Jian Qing Shi, Taeryon Choi, Gaussian Process Regression Analysis for Functional Data, Chapman and Hall/CRC, 2011.
[50] Carl Edward Rasmussen, Hannes Nickisch, Gaussian processes for machine learning (gpml) toolbox, Journal of Machine Learning Research 11 (Nov) (2010) 3011–3015..
[51] David Harrison Jr, Daniel L Rubinfeld, Hedonic housing prices and the demand for clean air, Journal of Environmental Economics and Management 5 (1) (1978) 81–102.
[52] Xuan Liang, Shuo Li, Shuyi Zhang, Hui Huang, Song Xi Chen, Pm2. 5 data reliability, consistency, and air quality assessment in five chinese cities, Journal of Geophysical Research: Atmospheres 121 (17) (2016) 10–220..
[53] Naonori Ueda, Ryohei Nakano, Zoubin Ghahramani, Geoffrey E Hinton, Smem algorithm for mixture models, Neural Computation 12 (9) (2000) 2109–2128.
[54] Zhihua Zhang, Chibiao Chen, Jian Sun, Kap Luk Chan, Em algorithms for gaussian mixtures with split-and-merge operation, Pattern Recognition 36 (9) (2003) 1973–1983.
[55] Yiu-ming Cheung, red Maximum weighted likelihood via rival penalized EM for density mixture clustering with automatic model selection, IEEE Transactions on Knowledge and Data Engineering 17 (6) (2005) 750–761.
[56] Zhihua Zhang, Kap Luk Chan, Wu. Yiming, Chibiao Chen, Learning a multivariate gaussian mixture model with the reversible jump mcmc algorithm, Statistics and Computing 14 (4) (2004) 343–355.
[57] Zhe Qiang, Jinwen Ma, Automatic model selection of the mixtures of gaussian processes for regression, in: International Symposium on Neural Networks, Springer, 2015, pp. 335–344..
[58] JinWen Ma, TaiJun Wang, Entropy penalized automated model selection on gaussian mixture, International Journal of Pattern Recognition and Artificial Intelligence 18 (08) (2004) 1501–1512.
[59] Jinwen Ma, Automated model selection (ams) on finite mixtures: a theoretical analysis, in: The 2006 IEEE International Joint Conference on Neural Network Proceedings, IEEE, 2006, pp. 4139–4145..
[60] T. Li, J. Ma, Fuzzy clustering with automated model selection: entropy penalty approach, in: 2018 14th IEEE International Conference on Signal Processing (ICSP), 2018, pp. 571–576.
[61] Jinwen Ma, Taijun Wang, Xu. Lei, A gradient byy harmony learning rule on gaussian mixture with automated model selection, Neurocomputing 56 (2004) 481–487.
[62] Jinwen Ma, Jianfeng Liu, The byy annealing learning algorithm for gaussian mixture with automated model selection, Pattern Recognition 40 (7) (2007) 2029–2037.
[63] Jinwen Ma, Xuefeng He, A fast fixed-point byy harmony learning algorithm on gaussian mixture with automated model selection, Pattern Recognition Letters 29 (6) (2008) 701–711.
[64] Wenli Zheng, Zhijie Ren, Yifan Zhou, Jinwen Ma, Byy harmony learning of log-normal mixtures with automated model selection, Neurocomputing 151 (2015) 1015–1026.
[65] Yunsheng Jiang, Chenglin Liu, Jinwen Ma, Byy harmony learning of t-mixtures with the application to image segmentation based on contourlet texture features, Neurocomputing 188 (2016) 262–274.
[66] Longbo Zhao, Ziyi Chen, Jinwen Ma, An effective model selection criterion for mixtures of gaussian processes, in: International Symposium on Neural Networks, Springer, 2015, pp. 345–354..
[67] Edward Snelson, Zoubin Ghahramani, Sparse gaussian processes using pseudo-inputs, in: Advances in Neural Information Processing Systems, 2006, pp. 1257–1264..
[68] Malte Kuss, Carl E. Rasmussen, Gaussian processes in reinforcement learning, in: Advances in Neural Information Processing Systems, 2004, pp. 751–758..
[69] Jack Wang, Aaron Hertzmann, David J. Fleet, Gaussian process dynamical models, in: Advances in Neural Information Processing Systems, 2006, pp. 1441–1448..
[70] Roger Frigola, Yutian Chen, Carl Edward Rasmussen, Variational gaussian process state-space models, in: Advances in Neural Information Processing Systems, 2014, pp. 3680–3688..
[71] Shahin Boluki, Mohammad Shahrokh Esfahani, Xiaoning Qian, Edward R. Dougherty, Constructing pathway-based priors within a Gaussian mixture model for Bayesian regression and classification, IEEE/ACM Transactions on Computational Biology and Bioinformatics 16 (2) (2017) 524–537..

Tao Li received the B.S. degree in mathematics from University of Science and Technology of China, Hefei, China, in 2017. Currently he is pursuing the Ph.D. degree in applied mathematics from Peking University, Beijing, China. His research interests include machine learning, Bayesian inference, and neural networks.

Di Wu received the B.S. degree in mathematics from Shaanxi Normal University, Xi'an, China, in 2012, and the Ph.D. degree in applied mathematics from Peking University, Beijing, China, in 2018. Since 2018, he has been a Lecturer with the School of Computer Science, Shaanxi Normal University. His current research interests include machine learning, data mining and neural network.

Jinwen Ma received the M.S. degree in applied mathematics from Xi'an Jiaotong University in 1988 and the Ph.D. degree in probability theory and statistics from Nankai University in 1992. From July 1992 to November 1999, he was a lecturer or associate professor at the Department of Mathematics, Shantou University. From December 1999, he became a full professor at the Institute of Mathematics, Shantou University. From September 2001, he has joined the Department of Information Science at the School of Mathematical Sciences, Peking University, where he is currently a full professor and Ph.D. tutor. During 1995 and 2003, he also visited several times at the Department of Computer Science and Engineering, the Chinese University of Hong Kong as a Research Associate or Fellow. He also worked as Research Scientist at Amari Research Unit, RIKEN Brain Science Institute, Japan from September 2005 to August 2006. He has published over 100 academic papers on neural networks, pattern recognition, bioinformatics, and information theory.