Neurocomputing 235 (2017) 274–286

Contents lists available at ScienceDirect

Neurocomputing

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

Self-adaptive robust nonlinear regression for unknown noise via mixture of Gaussians

Haibo Wang[a], Yun Wang[b,*], Qinghua Hu[b]

[a] School of Economics and Management, Hubei University of Technology, Wuhan, Hubei, China
[b] School of Computer Science and Technology, Tianjin University, Tianjin, China

ARTICLE INFO

Communicated by Wei Chiang Hong

Keywords:
Self-adaptive nonlinear regression
Unknown noise
Mixture of Gaussians
Expectation maximization

ABSTRACT

For most regression problems, the optimal regression model can be obtained by minimizing a loss function, and the selection of loss functions has great effect on the performance of the derived regression model. Squared loss is widely used in regression. It is theoretically optimal for Gaussian noise. However, real data are usually polluted by complex and unknown noise, especially in the era of big data, the noise may not be fitted well by any single distribution. To address the above problem, two novel nonlinear regression models for single-task and multi-task problems are developed in this work, where the noise is fitted by Mixture of Gaussians. It was proved that any continuous distributions can be approximated by Mixture of Gaussians. To obtain the optimal parameters in the proposed models, an iterative algorithm based on Expectation Maximization is designed. The proposed models turn to be a self-adaptive robust nonlinear regression models. The experimental results on synthetic and real-world benchmark datasets show that the proposed models produce good performance compared with current regression algorithms and provide superior robustness.

1. Introduction

Regression, which is concerned to extract hidden rules from data, is an very old, but still a hot topic today [1]. The goal of regression is to predict the value of target variables given the value of a D-dimensional vector of independent variables [2]. Currently, regression analysis is widely used in various domains, such as gold returns [3], solar power output forecasting [4], face recognition [5] and so on. What attracts more attention is the performance of regression models in real complex conditions.

To develop regression algorithms, three important issues should be taken into consideration, namely model structures, objective functions and their corresponding optimization methods [1]. According to model structures, regression algorithms can generally be divided into two large categories: linear regression models and nonlinear regression models. As to objective functions, loss functions have great effect on the performance of regression models. The selection of loss functions is mostly dependent on the types of noises [6,7]. For example, squared loss is good for Gaussian noise, least absolute deviation loss for Laplace noise [8], and so on. After obtaining the objective functions, we should develop optimization methods to search the optimal solution under the optimization objective functions.

However, for some real-world applications, training sets are usually subject to unknown but complex noises. The underlying assumption of Gaussian distributed error term in traditional models will be not reliable in such case. There are two solutions to solve the regression problems [9]. The first solution is to diagnose the outliers, which can be seen as special noise with long tail [10], then the training samples processed by removing the detected outliers will be fed into the regression models [11]. The second solution is to construct a regression model which is robust to outliers directly [12].

For the first solutions, generally, outliers can be identified by using five basic plots (Graph of predicted residuals, Williams graph, Pregibon graph, McCulloh-and-Meeter graph, L-R graph) [13,14] and other additional methods mentioned in chemometrical textbooks [15]. Besides, there are many outlier detection techniques have been proposed recently, which can be divided four categories [16]: statistical [17], distance-based [18], density-based [19] and soft computing [20]. The final constructed regression model requires two-step procedure [13]. The final regression accuracy depends largely on the goodness of outliers detection results. And those outliers detections methods have the risks of identifying normal points as outliers. In this case, certain information in training samples will lose due to the reduction of useful normal samples, that will have great effects on regression performance, especially for the small size training samples. Moreover, detected outliers may be also contains certain information. For the above

* Corresponding author.
E-mail address: wangyun15@tju.edu.cn (Y. Wang).

http://dx.doi.org/10.1016/j.neucom.2017.01.024
Received 28 July 2016; Received in revised form 23 November 2016; Accepted 4 January 2017
Available online 16 January 2017

considerations, in terms of regression modeling, we focus on the robust regression models.

In order to improve the robustness of the regression model, much effort has been made in the past few years. The common strategy to enhance the robustness of the regression model is to add weights to different samples. In [21], the authors pointed out that samples with large simulation residuals should be given small weights. And in [22], authors claimed that the relatively smaller weights should be given to the sample points with large distance to others. Four different types of weighting functions including Huber, Hampel, Logistic and Myriad are studied in [23], the results show that Logistic and Myriad weighting function are more robust than the other two functions in most cases [24]. However, it is a difficult task to determine the optimal weight to each sample. Some other researchers suggest using robust loss functions to reduce the effect of different noises. In [25], maximum correntropy criterion that comes from information theoretic learning is selected as a loss function, while a truncated least squares loss function is employed in [26]. This loss function is non-convex, which leads to a difficult optimization task. Besides, many other new regression models are proposed currently [27–30].

Another method to obtain a robust linear regression model is to model the noise comprehensively by mixture distributions. Mixture of Gaussian (MoG) [31], which can approximate any continues noise distributions [32] and is successfully applied in many domains and achieves great success, is used to fit the noise in regression problem. In [33], an autoregressive model was proposed with the noise fitted by MoG. Recently, other mixture distributions such as Mixture of t distributions [34] and scale mixtures of skew-normal distributions [35] were also employed to fit the unknown noise in LR model.

Besides, mixture distributions were also applied in nonlinear regression models. In [36,37], nonlinear regression models based on scale mixtures of skew-normal distributions were proposed with all parameters estimated by Bayesian inference and EM algorithm, respectively. Also, heteroscedastic nonlinear regression models based on scale mixtures of skew-normal distributions were proposed with parameters estimated by EM algorithm in [38]. However, the limitation of the above nonlinear regression models is that the nonlinear functions or nonlinear models are known in advance although the mixture distributions can fit the noise in nonlinear regression models. In literatures, there exist many nonlinear regression models such as SVM, LSSVM and ELM etc. to approximate the unknown nonlinear relationship between inputs and outputs. According to the theory of Bayesian inference, square loss function is optimal when the noise is Gaussian distributed [1]. However, in reality, the noise in real-world is complex or unknown, a single distribution to describe the real noise is improper.

In order to solve the problem of mismatch between the loss function and the real unknown or complex noise distribution, and motivated by successful applications of MoG in linear and nonlinear regression problems under the condition that the linear and nonlinear relationships are known, in this paper, a novel nonlinear regression model is proposed to approximate the unknown nonlinear relationships between inputs and outputs with the feature of noise comprehensively described by MoG. In our paper, in order to solve the single-task and multi-tasks regression problems in reality with unknown or complex noise, we propose two robust nonlinear regression models: single-task nonlinear regression model (SNLR-MoG) and multi-tasks nonlinear regression model (MNLR-MoG), which are all under the MoG noise distribution assumption and are all optimized within EM frameworks.

The contributions of this paper are summarized as follows:

(1) A nonlinear regression technique based on MoG is proposed to build nonlinear regression model with unknown noise.
(2) Expectation Maximization is introduced to solve the proposed nonlinear regression model.
(3) Extensive experiments are conducted, and the regression results are compared with seven popular regression models and show that the proposed model has great advantages under complex or unknown noise conditions.

The rest of this paper is organized as follows. Section 2 describes some related works about LR model under different type of noise. The objective functions of proposed two nonlinear regression models are described in section 3, and the corresponding training processes of two models are introduced in Section 4. Experiments on synthetic datasets and real-world benchmark datasets are carried out and the corresponding results are shown in Section 5. Section 6 concludes the paper.

2. Related work

In this section, some related works about linear regression models under different types of noise are presented. The linear regression model is expressed as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{e}, \tag{1}$$

where $\mathbf{y} = [\, y_1, y_2, \ldots, y_n]^T$ means the vector of dependent variable, and $\mathbf{X} = [x_1, x_2, \ldots, x_n]^T$ is the matrix of independent variable, $\boldsymbol{\beta} = [\beta_1, \beta_2, \ldots, \beta_k]^T$ is the vector of regression coefficients, $\boldsymbol{e} = [e_1, e_2, \ldots, e_n]^T$ means the vector of model noise. Assuming that the noise of LR model is a Gaussian with zero mean and unknown variance, namely

$$p(e_i) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{e_i^2}{2\sigma^2}\right) \tag{2}$$

In this case, the likelihood of $\boldsymbol{e}$ can be written as

$$p(\boldsymbol{e}) = \prod_{i=1}^{n} p(e_i) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n \exp\left(-\frac{\sum_{i=1}^{n} e_i^2}{2\sigma^2}\right) \tag{3}$$

By changing from e_i to y_i, the corresponding density is expressed as

$$p(\,\mathbf{y}|\boldsymbol{\beta}) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^n \exp\left(-\frac{\sum_{i=1}^{n} (y_i - x_i\boldsymbol{\beta})^2}{2\sigma^2}\right) \tag{4}$$

Given the likelihood function above, then the log-likelihood can be easily computed as follows:

$$L(\mathbf{y}\,|\boldsymbol{\beta}) = -\frac{n}{2}\log(2\pi\sigma^2) - \frac{\sum_{i=1}^{n} (\,y_i - x_i\boldsymbol{\beta})^2}{2\sigma^2} \tag{5}$$

The estimated values of parameter $\boldsymbol{\beta}$ can be obtained by maximizing the log-likelihood function $L(\mathbf{y}|\boldsymbol{\beta})$, namely

$$\widehat{\boldsymbol{\beta}} = \arg\max_{\beta} L(\mathbf{y}\,|\,\boldsymbol{\beta}) = \arg\min_{\beta} \sum_{i=1}^{n} (\,y_i - x_i\,\boldsymbol{\beta})^2 \tag{6}$$

In this case, maximizing log-likelihood is equivalent to minimizing the sum of squared residuals, and then we have

$$\widehat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \tag{7}$$

Concluded from the above interference, when given the noise distribution $p(e_i)$ in linear regression model, the regression coefficient can be obtained by maximizing the log-likelihood function $L(\mathbf{y}|\boldsymbol{\beta})$. Generally, the assumption that the noise is Gaussian distributed is sometimes improper in real-world applications. Therefore, in linear regression model, the original assumption that the noise obeys Gaussian distribution is replaced by the assumption that the noise obeys different types of distributions, such as Laplace distribution, Beta distribution and Huber distribution etc. And hence, the different improved linear regression models with different single noise distributions are proposed. Table 1 shows the different noise distributions and their corresponding optimal objective functions.

In practice, the noise is complex and unknown if the data are

Table 1
Different noise distributions and their corresponding optimized objective functions.

Noise distribution	PDF of noise	Optimized objective function
Gaussian	$p(e_i)=\frac{1}{\sqrt{2\pi}}exp\left(-\frac{e_i^2}{2}\right)$	$\text{argmin}_\beta \sum_{i=1}^n \frac{e_i^2}{2}$
Beta	$p(e_i)=e_i^{\alpha-1}\cdot(1-e_i)^{\beta-1}\cdot h$ $h=\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)}$	$\text{argmin}_\beta \sum_{i=1}^n (1-\alpha)\log(e_i)+(1-\beta)\log(1-e_i)$
Laplace	$p(e_i)=\frac{1}{\sigma}\exp\left(-\frac{\lvert e_i\rvert}{\sigma}\right)$	$\text{argmin}_\beta \sum_{i=1}^n \lvert e_i\rvert$
ε- insensitive	$p(e_i)=\begin{cases}\frac{1}{2(1+\varepsilon)} & if\ \lvert e_i\rvert\le\varepsilon\\ \frac{e^{\varepsilon-\lvert e_i\rvert}}{2(1+\varepsilon)} & otherwise\end{cases}$	$\text{argmin}_\beta \sum_{i=1}^n \lvert e_i\rvert_\varepsilon$
Huber	$p(e_i)=\begin{cases}e^{-e_i^2/2} & e_i\le\varepsilon\\ e^{\varepsilon^2/2-\varepsilon\lvert e_i\rvert} & otherwise\end{cases}$	$\text{argmin}_\beta \sum_{i=1}^n c(e_i)$ where $c(e_i)=\begin{cases}e_i^2/2 & e_i\le\varepsilon otherwise\\ \varepsilon\lvert e_i\rvert-\varepsilon^2/2\end{cases}$
MoG	$p(e_i)=\sum_{k=1}^K \pi_k N_k(e_i\lvert 0,\sigma_k^2)$	$\text{argmax}_\beta \sum_{i=1}^n \log\sum_{k=1}^K\left(\pi_k\left(\frac{1}{\sqrt{2\pi}\sigma_k}\exp\left(-\frac{e_i^2}{2\sigma_k^2}\right)\right)\right)$

collected in multi-source and heterogeneous environments. We don't know the real distribution of the noise. Therefore, using some specific distribution (such as Laplace and Beta distribution) to model real noise is not optimal. MoG is proved to have good approximation capability for any continuous distributions, and it can fit the complex and unknown noise appropriately and adaptively when we have no prior knowledge about real noise in linear regression model. When the noise is assumed to obey MoG, its corresponding optimal objective is also presented in Table 1.

Since the relationship between dependent variable and a set of independent variables is usually nonlinear, in the following section, we propose a nonlinear regression model with MoG.

3. Objective function of the proposed robust models

3.1. Objective function of robust single-task regression

Generally, the nonlinear regression model is expressed as

$$y_i = \boldsymbol{w}^T\varphi(x_i) + b + e_i \tag{8}$$

where $\varphi(\cdot)$ is a map function, which can map inputs into a high dimension feature space, b is the bias, $\boldsymbol{w}\in R^d$ represents the coefficient vector, and $e_i = y_i - \boldsymbol{w}^T\varphi(x_i) - b$ is a noise term, which is usually assumed be Gaussian distributed. Then the estimation of $\boldsymbol{w}$ and b can be obtained by minimize the following objective function

$$\arg\min_{\boldsymbol{w},b}\frac{1}{2}\parallel\boldsymbol{w}\parallel_2^2+\frac{\gamma'}{2}\parallel\boldsymbol{e}\parallel_2^2\ s.t.\ \boldsymbol{e}=\mathbf{y}-\mathbf{G}^T\boldsymbol{w}-b\mathbf{1}_n \tag{9}$$

where $\mathbf{G}=[\varphi(x_i),\varphi(x_2),\ldots,\varphi(x_n)]\in R^{d\times n}$, $\mathbf{1}_n=(1,1,\ldots,1)^T\in R^n$, $\mathbf{y}=(y_1,y_2,\ldots,y_n)^T\in R^n$ is the vector of target, and $\boldsymbol{e}=[e_1,e_2,\ldots,e_n]^T\in R^n$ means the noise term, and γ' is the regularization coefficient.

However, the above model is sensitive to gross noises and outliers, which are often introduced in data acquisition. And, in real world, the noise is unknown or very complex, the assumption that the noise is Gaussian distributed is improper, it is natural to use a MoG to model the noise term for its great ability of approximating to any continuous distributions [32]. So in this paper, we assume that the noise is a random variable and obeys a MoG distribution, which can be directly expressed by the following equation [39]:

$$p(e)=\sum_{k=1}^K \pi_k N_k(e\lvert 0,\sigma_k^2) \tag{10}$$

where $p(e)$ represents the probability density function of the noise e, $N_k(e\lvert 0,\sigma_k^2)$ is the Gaussian distribution with zero mean and variance σ_k^2, K is the number of independent Gaussian distribution in MoG model, π_k is the weight coefficient, and $\pi_k\ge 0$, $\sum_{k=1}^K\pi_k=1$.

Therefore, according to the result in Table 1, the objective function can be expressed as

$$\arg\min_{\boldsymbol{w},\boldsymbol{e}}\frac{1}{2}\parallel\boldsymbol{w}\parallel_2^2-\frac{\gamma}{2}\sum_{i=1}^n\log\sum_{k=1}^K\left(\pi_k\left(\frac{1}{\sqrt{2\pi}\sigma_k}\exp\left(-\frac{e_i^2}{2\sigma_k^2}\right)\right)\right)$$
$$s.t.\ \boldsymbol{e}=\mathbf{y}-\mathbf{G}^T\boldsymbol{w}-b\mathbf{1}_n \tag{11}$$

3.2. Objective function of robust multi-task regression

For multi-task regression model, the number of tasks is m, and target matrix is represented by $\boldsymbol{Y}=(\boldsymbol{y}_1,\boldsymbol{y}_2,\ldots,\boldsymbol{y}_m)\in R^{n\times m}$, $\boldsymbol{E}\in R^{n\times m}$ is the noise term, $\boldsymbol{b}=(b_1,b_2,\ldots,b_m)^T\in R^m$ then the multi-task regression model can be expressed as

$$\boldsymbol{Y}=\boldsymbol{G}^T\boldsymbol{W}+repmat(\boldsymbol{b}^T,n,1)+\boldsymbol{E} \tag{12}$$

where $\boldsymbol{W}=(\boldsymbol{w}_1,\boldsymbol{w}_2,\ldots,\boldsymbol{w}_m)\in R^{d\times m}$ represents the coefficient matrix, and function $repmat(\cdot)$ is to replicate matrix.

Similar to the above single task regression model, the objective function of proposed robust multi-task regression model is

$$\arg\min_{\boldsymbol{w}_j,\boldsymbol{b},\boldsymbol{E}}\frac{1}{2}\sum_{j=1}^m\parallel\boldsymbol{w}_j\parallel_2^2-\frac{\gamma}{2}\sum_{j=1}^m\sum_{i=1}^n\log\sum_{k=1}^K\left(\pi_k\left(\frac{1}{\sqrt{2\pi}\sigma_k}\exp\left(-\frac{e_{ij}^2}{2\sigma_k^2}\right)\right)\right)$$
$$s.t.\ \boldsymbol{Y}-\boldsymbol{G}^T\boldsymbol{W}-repmat(\boldsymbol{b}^T,n,1)=\boldsymbol{E} \tag{13}$$

4. Optimization with expectation maximization

In this section, we provides the optimization of all parameters in proposed single-task and multi-task nonlinear regression model.

4.1. Single-task nonlinear regression based on MoG (SNLR-MoG)

For single-task regression problem, the likelihood of $\boldsymbol{e}$ is

$$p(\boldsymbol{e}|\Theta) = \prod_{i=1}^{n} p(e_i|\Theta) = \prod_{i=1}^{n} \sum_{k=1}^{K} \pi_k N_k(e_i|0, \sigma_k^2) \tag{14}$$

where $\Theta = (\pi_1, \ldots, \pi_K, \sigma_1^2, \ldots, \sigma_K^2, \boldsymbol{w}, b)$ is the mixture set of parameters. Then the log-likelihood function is calculated as

$$L(\boldsymbol{e}|\Theta) = \sum_{i=1}^{n} \left(\log \sum_{k=1}^{K} \pi_k \left(\frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{e_i^2}{2\sigma^2} \right) \right) \right) \tag{15}$$

The estimated values of parameter set Θ can be obtained by maximizing the log-likelihood function $L(\boldsymbol{e}|\Theta)$. However, as to many specific problems, due to the complex expression of the log-likelihood function, the parameter set Θ cannot be calculated directly. So we should take the other methods to estimate Θ, and EM (expectation maximization) algorithm [40,41] is an effective algorithm to solve such problems by assuming that the observed data is incomplete [40]. The complete data consists of the observed data $\boldsymbol{e} = (e_1, e_2, \ldots, e_n)^T$ and the unobservable component indicator matrix $\boldsymbol{Z} = (z_1, z_2, \ldots, z_n)^T$, in which $z_i = (z_{i1}, z_{i2}, \ldots, z_{iK})$ is an unobservable component indicator vector (K is the number of components) [41]. If observed data e_i comes from the j th component, then $z_{ij} = 1$ and the other elements of z_i are 0, so we have $\sum_{k=1}^{K} z_{ik} = 1$ and $\sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} = n$. The log likelihood of the complete data $\chi = (\boldsymbol{e}, \boldsymbol{Z})$ can be obtained by rewriting Eq. (11), namely

$$L(\chi|\Theta) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \left(\log \pi_k - \frac{1}{2} \log(2\pi\sigma_k^2) - \frac{e_i^2}{2\sigma_k^2} \right) \tag{16}$$

In E-step of EM algorithm, based on the given observed data $\boldsymbol{e}$ and the estimated parameters Θ^s in the sth iteration, the Q function $Q(\Theta|\Theta^s)$ can be obtained by computing the conditional expectation of $L(\chi|\Theta)$ with respect to $\boldsymbol{Z}$ [41], namely

$$Q(\Theta|\Theta^s) = \sum_{i=1}^{n} \sum_{k=1}^{K} \gamma_{ik} (\log \pi_k) + \sum_{i=1}^{n} \sum_{k=1}^{K} \gamma_{ik} \left(-\frac{1}{2} \log 2\pi - \frac{1}{2} \log \sigma_k^2 - \frac{e_i^2}{2\sigma_k^2} \right) \tag{17}$$

where γ_{ik} means the posterior responsibility that the ith observation e_i comes from to the kth component of the mixture [8]:

$$\gamma_{ik} = E(z_{ik}) = \frac{\pi_k^{(s)} N_k(e_i^{(s)}|0, \sigma_k^{2(s)})}{\sum_{k=1}^{K} \pi_k^{(s)} N_k(e_i^{(s)}|0, \sigma_k^{2(s)})} \tag{18}$$

In M-step of EM algorithm, the Q function should be maximized to get the updated formulation of all parameters.

Update π

We have a constraint on π_k, namely $\sum_{k=1}^{K} \pi_k = 1$, by introducing Lagrange multiplier method, the final updated formula on π_k is [8,41]

$$\pi_k = \frac{\sum_{i=1}^{n} \gamma_{ik}}{n}, \; k = 1, 2, \ldots, K \tag{19}$$

Update Σ

To update σ_k^2, we have [8,41]

$$\sigma_k^2 = \sum_{i=1}^{n} e_i^2 \left(\sum_{i=1}^{n} \gamma_{ik} \right)^{-1}, \; k = 1, 2, \ldots, K \tag{20}$$

Update $\boldsymbol{e}$ To maximize the Q function is to maximize the following function, which is rewritten as

$$\begin{aligned} J_{w,b} &= -\sum_{i=1}^{n} \sum_{k=1}^{K} \gamma_{ik} \frac{e_i^2}{2\sigma_k^2} \\ &= -\sum_{i=1}^{n} \left(\sum_{k=1}^{K} \frac{\gamma_{ik}}{2\sigma_k^2} \right) (y_i - \boldsymbol{w}^T \varphi(x_i) - b)^2 \\ &= -\| \boldsymbol{\eta} \odot (\mathbf{y} - \mathbf{G}^T \boldsymbol{w} - b\mathbf{1}_n) \|_2^2 \end{aligned} \tag{21}$$

where $\boldsymbol{\eta} = (\eta_1, \eta_2, \ldots, \eta_n)^T \in R^n$ is weight vector, $\eta_i = \sqrt{\sum_{k=1}^{K} \frac{\gamma_{ik}}{2\sigma_k^2}}$, and $\odot$ means the Hadamard product. In order to update $\boldsymbol{e}$, the estimator $\boldsymbol{w}$, b should be obtained firstly by maximizing $J_{w,b}$, Eq. (21) can be transformed into

$$\begin{aligned} &\arg\min_{\boldsymbol{w},b,\boldsymbol{e}} \| (\boldsymbol{\eta} \odot \boldsymbol{e}) \|_2^2 \\ &s.\,t.\; \mathbf{y} - \mathbf{G}^T \boldsymbol{w} - b\mathbf{1}_n = \boldsymbol{e} \end{aligned} \tag{22}$$

In the model of machine learning, in order to control over-fitting, a regularization term is added, so that the total objective function to be minimized takes the form

$$\begin{aligned} &\arg\min_{\boldsymbol{w},b,\boldsymbol{e}} \| (\boldsymbol{\eta} \odot \boldsymbol{e}) \|_2^2 + C \| \boldsymbol{w} \|_2^2 \\ &s.\,t.\; \mathbf{y} - \mathbf{G}^T \boldsymbol{w} - b\mathbf{1}_n = \boldsymbol{e} \end{aligned} \tag{23}$$

where C is the regularization coefficient that controls the relative importance of the data-dependent error $\| (\boldsymbol{\eta} \odot \boldsymbol{e}) \|_2^2$ and the regularization term $\| \boldsymbol{w} \|_2^2$. For the sake of convenience, the above equation can be transformed into

$$\begin{aligned} &\arg\min_{\boldsymbol{w},b,\boldsymbol{e}} \frac{1}{2} \| \boldsymbol{w} \|_2^2 + \frac{\gamma}{2} \| (\boldsymbol{\eta} \odot \boldsymbol{e}) \|_2^2 \\ &s.\,t.\; \mathbf{y} - \mathbf{G}^T \boldsymbol{w} - b\mathbf{1}_n = \boldsymbol{e} \end{aligned} \tag{24}$$

where $\gamma = 1/C$. To solve the upper optimization problem with equality constraints, the Lagrange function is constructed, which can be expressed as

$$L(\boldsymbol{w}, \boldsymbol{e}, b, \boldsymbol{\alpha}) = \frac{1}{2} \| \boldsymbol{w} \|_2^2 + \frac{\gamma}{2} \| (\boldsymbol{\eta} \odot \boldsymbol{e}) \|_2^2 - \boldsymbol{\alpha}^T (\boldsymbol{e} - \mathbf{y} + \mathbf{G}^T \boldsymbol{w} + b\mathbf{1}_n) \tag{25}$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)^T$ means the Lagrange multiplier. According to the KKT conditions,

$$\begin{cases} \dfrac{\partial L}{\partial \boldsymbol{w}} = 0 \rightarrow \boldsymbol{w} = \boldsymbol{G}\boldsymbol{\alpha} \\ \dfrac{\partial L}{\partial b} = 0 \rightarrow \boldsymbol{\alpha}^T \mathbf{1}_n = 0 \\ \dfrac{\partial L}{\partial \boldsymbol{e}} = 0 \rightarrow diag(\boldsymbol{e}) = \gamma^{-1} diag(\boldsymbol{\eta})^{-2} diag(\boldsymbol{\alpha}) \\ \dfrac{\partial L}{\partial \alpha} = 0 \rightarrow \boldsymbol{e} - \mathbf{y} + \mathbf{G}^T \boldsymbol{w} + b\mathbf{1}_n = 0\mathbf{1}_n \end{cases} \tag{26}$$

where diagonal matrix is produced by function $diag(\cdot)$. By eliminating $\boldsymbol{w}$ and $\boldsymbol{e}$, the solution is given by the following set of linear equation:

$$\begin{bmatrix} 0 & \boldsymbol{I}_n^T \\ \mathbf{1}_n & \boldsymbol{K} + \gamma^{-1} diag(\boldsymbol{\eta})^{-2} \end{bmatrix} \times \begin{bmatrix} b \\ \alpha \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{y} \end{bmatrix}$$

where $\boldsymbol{K} = \boldsymbol{G}^T \boldsymbol{G}$ means the kernel function, and $K(x_i, x_j) = \varphi(x_i)^T \varphi(x_j)$. Let the solution of above equation be $\tilde{b}$, $\tilde{\boldsymbol{\alpha}}$. The final regression model is expressed as

$$\hat{y} = f(x) = \varphi(x)^T \boldsymbol{G} \tilde{\boldsymbol{\alpha}} = \sum_{j=1}^{n} \tilde{\alpha}_j K(x, x_j) + \tilde{b} \tag{27}$$

Then, $\boldsymbol{e}$ can be updated by the following equation:

$$e_i = y_i - \sum_{j=1}^{n} \tilde{\alpha}_j K(x, x_j) - \tilde{b} \tag{28}$$

In this paper, RBF kernel function is employed, and its expression is

$$K(x, y) = \exp\left(-\frac{\| x - y \|^2}{2\sigma^2} \right) \tag{29}$$

With the above updating rules, the optimal parameters in the proposed robust nonlinear regression model can be obtained with the noise fitted by the MoG model properly. Then we name the proposed model as SNLR-MoG.

An important task in SNLR-MoG model is to determine the optimal number of Gaussian mixtures $K_{optimal}$. In this work, we proposed a simple but effective rule to automatically estimate $K_{optimal}$. Firstly, a relative large K is selected in order to model noise distribution

comprehensively; then to check if the relative deviation $\frac{\left|\sigma_i^2-\sigma_j^2\right|}{\sigma_i^2+\sigma_j^2}$ between variances of two Gaussian components is smaller than a given small threshold ε, if $\frac{\left|\sigma_i^2-\sigma_j^2\right|}{\sigma_i^2+\sigma_j^2}<\varepsilon$, the ith and jth Gaussian component are combined into a new Gaussian component, of which $\pi_{new}=\pi_i+\pi_j$, $\sigma_{new}^2=(\sigma_i^2+\sigma_j^2)/2$. Accordingly, $K=K-1$.

Another important problem is that the stop condition in SNLR-MoG model, in this work we introduce a simple ξ-optimality stop condition [42], which is defined as follows:

$$\left\|\frac{\boldsymbol{w}^{old}}{\|\boldsymbol{w}^{old}\|}-\frac{\boldsymbol{w}^{new}}{\|\boldsymbol{w}^{new}\|}\right\|<\xi \tag{30}$$

With these updating rules above, the whole learning process for SNLR-MoG is summarized in Algorithm 1.

Algorithm 1. Single-task Nonlinear Regression Via Mixture of Gaussions.

Input:

1: $\mathbf{X}=(x_1, x_2,\ldots,x_n)\in R^{n\times d}$ – Input matrix of training set;
2: $\mathbf{y}=(y_1, y_2,\ldots,y_n)\in R^{n\times 1}$ – Output vector of training set;
3: K – Initialized number of Gaussian mixtures in MoG;
4: γ – Regularization coefficient;
5: σ^2 – RBF kernel parameter;

Output

6: $K_{optimal}$ – Optimal number of Gaussian mixtures in MoG;
7: $\boldsymbol{\pi}_{optimal}=(\pi_1, \pi_2,\ldots,\pi_{K_{optimal}})$ – Optimal weights of Gaussians;
8: $\boldsymbol{\Sigma}_{optimal}=(\sigma_1^2, \sigma_2^2,\ldots,\sigma_{K_{optimal}}^2)$ – Optimal variances of Gaussians;
9: $\widetilde{b}, \widetilde{\alpha}_1,\ldots,\widetilde{\alpha}_n$ – Optimal parameters in SNLR-MoG model;

Step 1: Parameters initialization:

10: Set the number of Gaussian mixtures K, weight vector π and variance vector $\boldsymbol{\Sigma}$ in MoG. Two small thresholds ε and ξ are set as 10^{-2} and 10^{-5}, respectively. And the initialized $\boldsymbol{e}^{(0)}$ is randomly selected, *iteration*=1;

Step 2: Update:

11: (E Step) Calculate γ_{ik} by Eq. (18) for $i=12,\ldots,n$, $k=1, 2,\ldots,K$;
12: (M Step for $\boldsymbol{\pi}$) Updateπ_k by Eq. (19) for $k=1, 2,\ldots,K$;
13: (M Step for $\boldsymbol{\Sigma}$) Update σ_k^2 by Eq. (20) for $k=1, 2,\ldots,K$;
14: (M Step for $\boldsymbol{e}$) Update $\boldsymbol{e}$ by Eq. (28);

Step 3: Stop Condition:

15: If the ξ-optimality stop condition (Eq. (30)) holds a smaller value than given ξ, then go to step 4 and tuning K according to the rule described above; otherwise, let *iteration*=*iteration* +1, and return to step 2;

Step 4: Obtain Output:

16: Output optimal $\widetilde{b}, \widetilde{\alpha}_1,\ldots,\widetilde{\alpha}_n$ in nonlinear regression model.

Therefore, from the phase of parameters optimization in SNLR-MoG, we can see that optimization for the proposed objective function (Eq. (11)) is equivalent to optimize the Eq. (24). From training phase, SNLR-MoG turns out to be a self-adaptive robust nonlinear regression model with self-adaptive weight $\boldsymbol{\eta}$ learning, and the noise term is described comprehensively via MoG at the same time.

4.2. Multi-task nonlinear regression based on MoG (MNLR-MoG)

In this section, multi-task nonlinear regression model based on MoG is also proposed. And the process of parameters optimization in Eq. (13) is nearly the same with optimization of SNLR-MoG except for the updated equation of $\boldsymbol{E}$. Here, we give the updated equation for γ_{ijk}, π_k, σ_k^2 directly:

$$\gamma_{ijk}=\frac{\pi_k N_k(e_{ij}|0,\sigma_k^2)}{\sum_{k=1}^{K}\pi_k N_k(e_{ij}|0,\sigma_k^2)} \tag{31}$$

$$\pi_k=\frac{\sum_{i=1}^{n}\sum_{j=1}^{m}\gamma_{ijk}}{n},\ k=1, 2,\ldots,K \tag{32}$$

$$\sigma_k^2=\frac{\sum_{i=1}^{n}\sum_{j=1}^{m}e_{ij}^2}{\sum_{i=1}^{n}\sum_{j=1}^{m}\gamma_{ijk}},\ k=1, 2,\ldots,K \tag{33}$$

Inspired by Eq. (24), when the weight matrix $\boldsymbol{\Omega}=(\boldsymbol{\eta}_1, \boldsymbol{\eta}_2,\ldots,\boldsymbol{\eta}_m)\in R^{n\times m}$ is calculated by Eq. (34), to update $\boldsymbol{E}$, $\boldsymbol{W}$ and $\boldsymbol{b}$ should be estimated firstly by Eq. (35).

$$\eta_{ij}=\sqrt{\sum_{k=1}^{K}\frac{\gamma_{ijk}}{2\sigma_k^2}} \tag{34}$$

$$\begin{aligned}&\arg\min_{\boldsymbol{W},\boldsymbol{b},\boldsymbol{E}}\frac{1}{2}trace(\boldsymbol{W}^T\boldsymbol{W})+\frac{\gamma}{2}trace((\boldsymbol{\Omega}\odot\boldsymbol{E})^T(\boldsymbol{\Omega}\odot\boldsymbol{E}))\\&s.\ t.\ \boldsymbol{Y}-\boldsymbol{G}^T\boldsymbol{W}-repmat(\boldsymbol{b}^T, n, 1)=\boldsymbol{E}\end{aligned} \tag{35}$$

By introducing Lagrange multiplier $\boldsymbol{\Lambda}=(\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2,\ldots,\boldsymbol{\alpha}_m)\in R^{n\times m}$, the Lagrange function can be expressed by

$$\begin{aligned}\boldsymbol{L}(\boldsymbol{W},\boldsymbol{b},\boldsymbol{E},\boldsymbol{\Lambda})=&\frac{1}{2}trace(\boldsymbol{W}^T\boldsymbol{W})+\frac{\gamma}{2}trace[(\boldsymbol{\Omega}\odot\boldsymbol{E})^T(\boldsymbol{\Omega}\odot\boldsymbol{E})]\\&-trace\{\boldsymbol{\Lambda}^T[\boldsymbol{E}-\boldsymbol{Y}+\boldsymbol{G}^T\boldsymbol{W}+repmat(\boldsymbol{b}^T, n, 1)]\}\end{aligned} \tag{36}$$

According to KKT condition, we can get

$$\begin{cases}\frac{\partial\boldsymbol{L}}{\partial\boldsymbol{W}}=0\rightarrow\boldsymbol{W}=\boldsymbol{G}\boldsymbol{\Lambda}\\\frac{\partial\boldsymbol{L}}{\partial\boldsymbol{b}}=0\rightarrow\mathbf{1}_n^T\boldsymbol{\Lambda}=\mathbf{0}_m^T\\\frac{\partial\boldsymbol{L}}{\partial\boldsymbol{E}}=0\rightarrow\boldsymbol{E}\odot\boldsymbol{\Omega}=\gamma^{-1}repmat(\mathbf{1}_n, 1, m)\\\frac{\partial\boldsymbol{L}}{\partial\boldsymbol{\Lambda}}=0\rightarrow\boldsymbol{E}-\boldsymbol{Y}+\boldsymbol{G}^T\boldsymbol{W}+repmat(\boldsymbol{b}^T, n, 1)\end{cases} \tag{37}$$

By eliminating $\boldsymbol{W}$ and $\boldsymbol{E}$, the solution will be given by the following linear system:

$$\begin{bmatrix}\mathbf{0}_{(n\times m)\times m} & \boldsymbol{A}^T\\ \boldsymbol{A} & \boldsymbol{B}\end{bmatrix}\times\begin{bmatrix}\boldsymbol{b}\\ \boldsymbol{\Xi}\end{bmatrix}=\begin{bmatrix}\mathbf{0}_m\\ \boldsymbol{Y}'\end{bmatrix}$$

where $\boldsymbol{\Xi}=(\boldsymbol{\alpha}_1^T, \boldsymbol{\alpha}_2^T,\ldots,\boldsymbol{\alpha}_m^T)^T$, $\boldsymbol{Y}'=(\boldsymbol{y}_1^T, \boldsymbol{y}_2^T,\ldots,\boldsymbol{y}_m^T)^T$. Block diagonal matrix is made by function *blockdiag*, then $\boldsymbol{A}=blockdiag\mathbf{1}_n, \mathbf{1}_n,\ldots,\mathbf{1}_n)$, $\mathbf{B}=blockdiag(\mathbf{K}+\gamma^{-1}diag(\boldsymbol{\eta}_1)^{-2}, \boldsymbol{K}+\gamma^{-1}diag(\boldsymbol{\eta}_2)^{-2},\ldots,.$

$\boldsymbol{K}+\gamma^{-1}diag(\boldsymbol{\eta}_m)^{-2})$

To train the above linear system efficiently, we apply the same method described in [43]. Let the solution of above equation be $\widetilde{\boldsymbol{\Lambda}}$ and $\widetilde{\boldsymbol{b}}$. Then the forecasting function can be defined as

$$\hat{y}=f(x)=\boldsymbol{k}(x)\widetilde{\boldsymbol{\Lambda}}+\widetilde{\boldsymbol{b}} \tag{38}$$

where $\boldsymbol{k}(x)=[K(x, x_1), K(x, x_2),\ldots,K(x, x_n)]$. Then, we can use the following formula to update $\boldsymbol{E}$:

$$\boldsymbol{E}=\boldsymbol{Y}-\boldsymbol{K}\widetilde{\boldsymbol{\Lambda}}+\widetilde{\boldsymbol{b}} \tag{39}$$

To select an optimal number of mixture in MoG, we also apply the rule described in section 4.1. And RBF kernel is also employed in multi-task model.

Summarily, the optimization of multi-task objective function (Eq. (13)) is to optimize the Eq. (36), which can also be seen as a self-adaptive robust nonlinear regression model by learning the weight matrix $\boldsymbol{\Omega}$ adaptively within the framework of EM algorithm.

Table 2
Forecasting performance indices and their definitions.

Metrics	Calculation	Evaluation criterion
MAE	$MAE = \frac{1}{n}\sum_{i=1}^{n}\left\|y_i - \hat{y}_i\right\|$	The smaller the better
RMSE	$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$	The smaller the better

Note: y_i and $\hat{y}_i$ represent the actual value and the estimated value at time i, respectively, n is the number of testing set.

5. Experiments and discussions

In this section, synthetic dataset and several benchmark datasets are employed to confirm the effectiveness of the proposed self-adaptive robust nonlinear regression model, SNLR-MoG and MNLR-MoG.

The regularization coefficient γ and kernel parameter σ^2 have a great effect on the generalization performance of propose model. Currently, many algorithms, such as particle swarm optimization [44] and grid search etc., have been proposed to select the optimal parameters. However, among these algorithms, grid search is one of the most popular and universal parameter optimization algorithm and widely used in many literatures [27,45]. Therefore, in this paper, we also employ grid search to get the optimal regularization coefficient γ and kernel parameter σ^2 by varying their values all from the set $\{2^i | i = -9, -8, \ldots, 0, \ldots, 8, 9\}$.

In this paper, two popular regression estimation metrics are employed, the mean absolute error (MAE) and the root mean square error (RMSE) [24,27], to evaluate and compare among different regression algorithms. Table 2 shows the definitions of each evaluation metric.

5.1. Single-task regression results of SNLR-MoG

5.1.1. Synthetic datasets

Two examples are conducted in this subsection. In the first example, the inputs of training samples are uniformly randomly selected from the interval $[-4, 4]$, and the corresponding outputs of the training samples are computed by the following function, namely

$$f(x) = \frac{\sin(3x)}{3x},\ x \in [-4, 4] \tag{40}$$

The number of training samples is 500. Similarly, the 1000 test samples are also generated with the same approach as how training set is created.

Another example, Mackey-Glass system [46] is also taken as test cases. First, a series of Mackey-Glass samples which is randomly generated by Eq. (41) without adding any noise.

$$\frac{dx(t)}{dt} = \frac{0.2x(t-17)}{1 + x^{10}(t-17)} - 0.1x(t) \tag{41}$$

For the series generated, we take the $x(t)$, $x(t-6)$, $x(t-12)$ and $x(t-18)$ as the inputs to forecast the value at time $t + \Delta t$, namely $x(t + \Delta t)$. Here, $\Delta t = 50$ is adopted. In our experiment, 1300 samples are selected and then divided into two sets: training set (contain the first 800 samples) and test set (contain the last 500 samples).

In order to test the validity of SNLR-MoG model, the selected training samples are added with outliers and four different types of noise (described in Table 3), while the selected test samples are noise-free. To generate outliers, we divided the training samples into 10 independent samples, and one of them becomes outliers (all original values in selected samples times 10). For each type of noise, 10 independent groups of noise samples are randomly generated and then randomly added into the training sets in order to avoid biased comparisons.

Here, WLSSVM, LSSVM, SVM, are employed to make performance comparisons. As to all compared models, regularization coefficient γ and kernel parameter σ^2 vary their values all from the set $\{2^i | i = -9, -8, \ldots, 0, \ldots, 8, 9\}$. Optimal parameters in above models are also chosen by grid search.

Table 3
Description of four different types of noise.

Noise type	Noise description
Gaussian noise	Noise obeys Gaussian noise $N(0, 0.25^2)$;
Laplace noise	Noise obeys Laplace noise $Laplace(0, 0.25^2)$;
Sparse noise	60% of the training samples are added with Gaussian noise $N(0, 0.2^2)$;
Mixture noise	20% of the training samples are added with uniformly distributed noise over $[-0.25, 0.25]$, 50% of training samples are contaminated with Gaussian noise $N(0, 0.1^2)$, 10% of the training samples are added with normal distributed noise $N(0, 1^2)$, and the remaining are corrupted Gaussian noise $N(0, 0.5^2)$.

Table 4
Results of models with or without outliers.

Models	Error	Regression model			
		LSSVM	WLSSVM	SVM	SNLR-MOG
$f(x) = \frac{\sin(3x)}{3x}$	MAE	1.4658e−05	9.1283e−04	0.0765	9.2162e−05
	RMSE	2.5165e−05	0.0018	0.0845	3.0296e−04
Mackey-Glass system	MAE	4.5085e−04	0.0024	0.0492	6.3015e−05
	RMSE	5.6408e−04	0.0033	0.0546	1.0251e−04

When outliers are recognized by outliers detection methods, these outliers will be removed from training samples, then the rest of training samples will be fed into the regression model. So we assume that all outliers are detected ideally in this paper, the compared model will be trained by training samples without outliers, and the proposed robust model will trained by training sample with outliers. Experiment results are shown in Table 4.

Concluded from Table 4, if the outliers can be absolutely detected and removed from the training samples, LSSVM model performances the best. However, we can see that the regression performances of proposed model without outliers detection are comparable with the performances of conventional nonlinear regression models without any outliers. In reality, outliers detection is a challengeable task, we cannot ensure that all outliers will be recognized. And for some regression tasks with small training samples, removing outliers will reduce the training samples again and take the risk of ignoring certain important information. Therefore, to construct a robust regression model will be time-saving and avoid the risk of loss of information.

Fig. 1 presents the values of MAE and MAPE under different noises of different models. Concluded from Fig. 1, the proposed model outperforms other compared models (WLSSVM, LSSVM and SVM) under sparse and mixture noise. However, when the training samples are polluted by Gaussian noise and Laplace noise, the performance gaps among all models are small, but the proposed model actually performs a little better that other models. Figs. 2 and 3 shows the experimental results on synthetic dataset with training samples polluted by different types of noise.

From the above two nonlinear regression experiments, the proposed model owns the advantages under the condition that the training set are polluted by complex noise and outliers. In reality, when the noise or outliers in training set is very complex or unknown, to solve these complex problems by using regression models which assume that the noise is Gaussian distributed is improper. But for the proposed SNLR-MoG model, it can model the complex noise distribution and

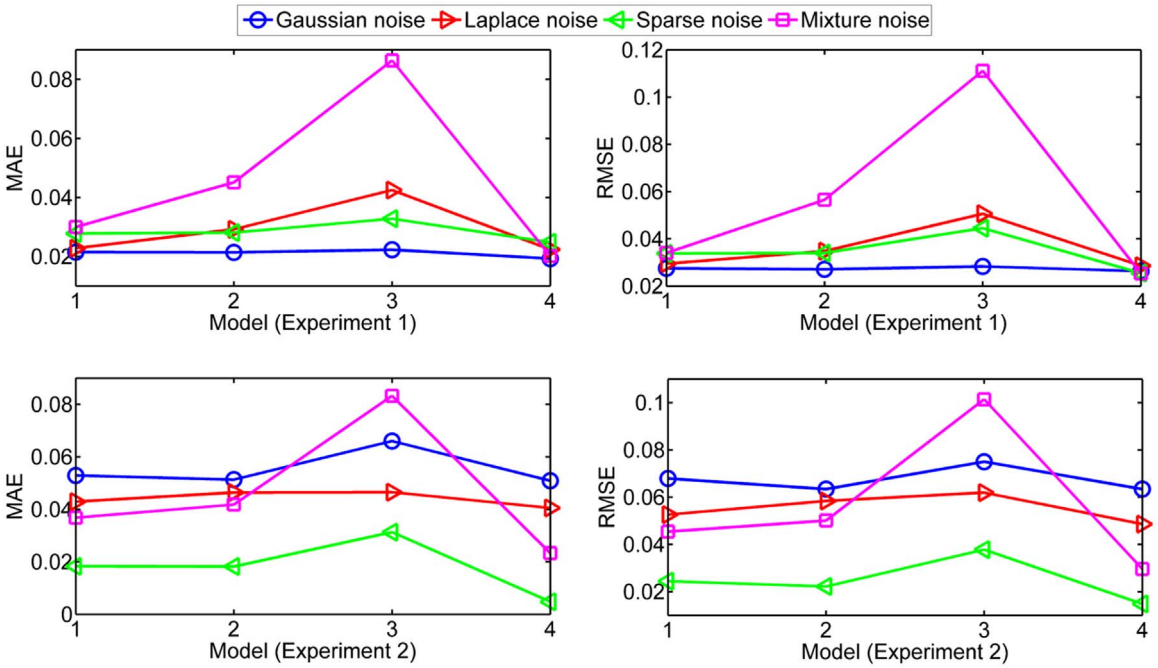

Fig. 1. Experimental results on synthetic dataset with training samples polluted by different types of noise Model 1 - WLSSVM, Model 2 - LSSVM, Model 3 - SVM, Model 4 - SNLR-MOG.

outliers adaptively. Therefore, when we have no prior knowledge about the noise or outliers in reality, it is wise to employ SNLR-MoG to solve the many complex single-task nonlinear regression problems.

5.1.2. Real-world benchmark datasets

In above subsection, SNLR-MoG has been proved its advantages on synthetic datasets. Now, experiments are carried out on the following real-world datasets: Bodyfat from the Statlib collection (http://lib.stat.cmu.edu/datasets); Box and Jenkins gas furnace dataset; Vehicles and River flow from the Time Series Data Library (TSDL); Auto MPG, House, Stock, MCPU, Energy, Wine and Concrete CS from UCI repository; and the inverse dynamics of a flexible robot arm from http://homes.esat.kuleuven.be/smc/daisy/daisydata.

Prior to regression process, both the input variables and target variable in all datasets are normalized into the interval [0,1] according to the following manner:

$$\widehat{x_i} = \frac{x_i - x_{min}}{x_{max} - x_{min}},\ i = 1, 2, \ldots, n \tag{42}$$

where x_{max} and x_{min} denote the maximum and minimum values of the observed series $\{x_i | i = 1, 2, \ldots, n\}$, respectively; $\widehat{x_i}$ represents the corresponding normalized value of x_i.

In this paper, our proposed model focus on the robust regression estimation under different types of noise and outliers, hence the training samples are added with noises (described in Table 5) or outliers to simulate the noise or outliers in reality and the test samples

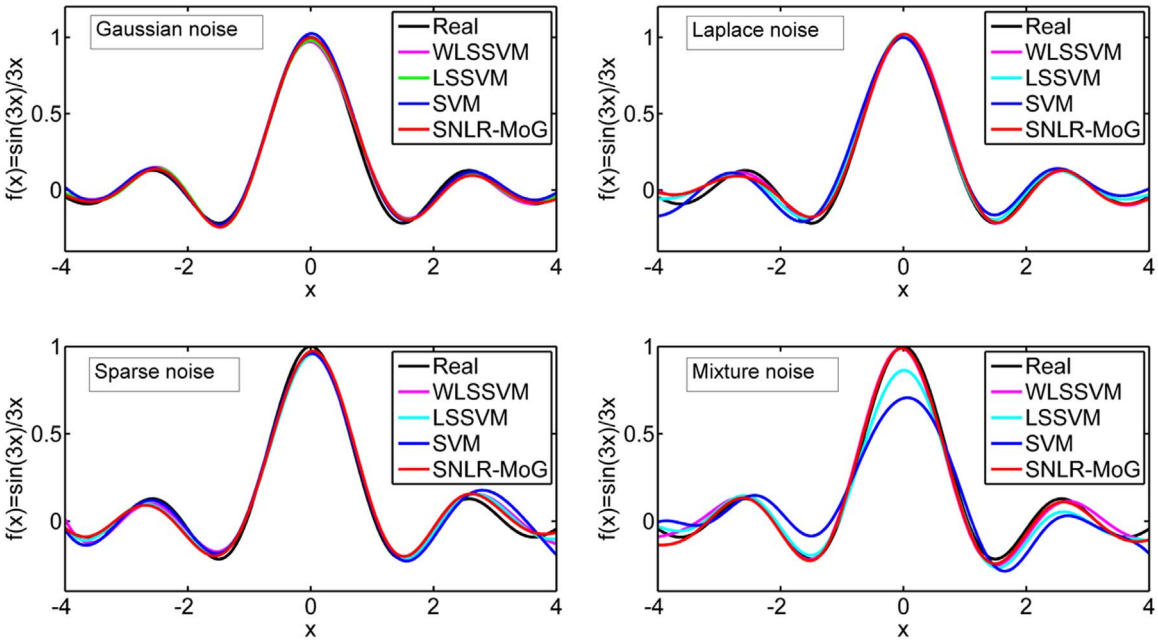

Fig. 2. Results of all models in example 1 with training samples polluted by different types of noise.

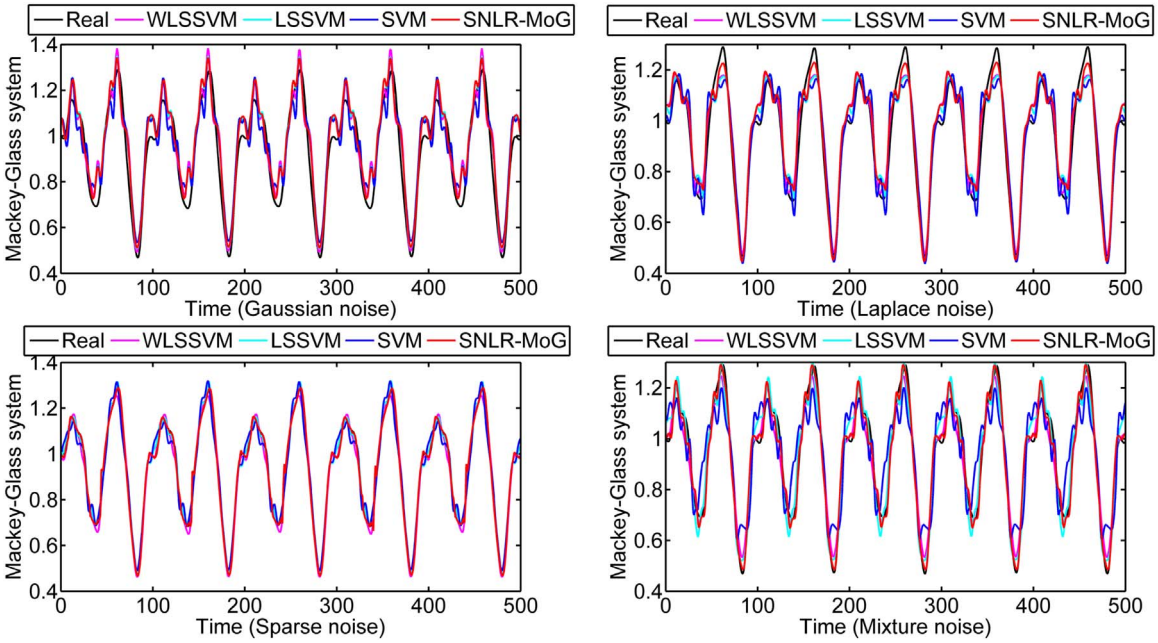

Fig. 3. Results of all models in example 2 with training samples polluted by different types of noise.

Table 5
Description of four different types of noises.

Noise type	Noise description
Gaussian noise	Noise obeys Gaussian noise $N(0, 0.15^2)$;
Laplace noise	Noise obeys Laplace noise $Laplace(0, 0.15^2)$;
Sparse noise	60% of the training samples are added with Gaussian noise $N(0, 0.3^2)$;
Mixture noise	20% of the training samples are added with uniformly distributed noise over [−0.25, 0.25], 50% of training samples are contaminated with Gaussian noise $N(0, 0.1^2)$, 10% of the training samples are added with normal distributed noise $N(0, 1^2)$, and the remaining are corrupted Gaussian noise $N(0, 0.5^2)$.

Table 6
Descriptions of all data sets used in this subsection.

Dataset	Samples	Attributes	(♯train)	(♯test)
Bodyfat	252	14	200	52
Gas furnace	293	6	200	93
Vehicles	246	12	200	46
River flow	588	12	400	188
Auto MPG	392	7	300	92
House	506	13	300	206
Stock	536	8	300	236
MCPU	209	6	150	59
Energy	768	6	500	268
Wine	1599	11	1000	599
Concrete CS	1030	8	700	330
Robot Arm	1019	9	700	319

are noise-free.

In our experiments, the number of samples and the attributes of all datasets used are presented in Table 6. And for each dataset, some samples are randomly selected from the whole data set for training, and the rest are used to test the effectiveness of the model, the specific numbers of training samples (♯train) and test samples (♯test) are also shown in Table 6.

The method to produce the outliers is the same as described in section 5.1.1. As to each type of noise, 10 independent groups of noise samples are randomly generated and then randomly added into the training samples. Finally, the averaged RMSE and MAE on the test samples are selected as the performance measures. Table 7 shows the results of different models. WLSSVM, LSSVM and SVM are trained

Table 7
Accuracy comparison between single-task regression models with or without outliers.

Dataset	Error metric	Regression models				
		WLSSVM	LSSVM	SVM	LR-MOG	SNLR-MoG
Bodyfat	MAE	0.0027	0.0049	0.0409	0.0269	**0.0014**
	RMSE	0.0035	0.0066	0.0508	0.0348	**0.0019**
Gas furnace	MAE	0.0251	0.0244	0.0554	**0.0189**	0.0223
	RMSE	0.0359	0.0346	0.0701	**0.0271**	0.0314
Vehicles	MAE	0.0683	0.0712	0.0763	0.0928	**0.0681**
	RMSE	0.0925	0.0918	0.0964	0.1251	**0.0904**
River flow	MAE	0.0483	**0.0474**	0.0608	0.0613	0.0484
	RMSE	0.0825	**0.0812**	0.0863	0.1087	0.0877
Auto MPG	MAE	0.0876	0.0873	**0.0833**	0.1216	0.0858
	RMSE	0.1210	0.1153	**0.1089**	0.1578	0.1163
House	MAE	0.1042	0.1184	0.0880	0.1831	**0.0792**
	RMSE	0.1588	0.1707	0.1366	0.2847	**0.1355**
Stock	MAE	0.0359	0.0355	0.0352	**0.0334**	0.0346
	RMSE	0.0463	0.0464	0.0458	**0.0432**	0.0453
MCPU	MAE	0.0432	0.0408	0.0478	0.0571	**0.0368**
	RMSE	0.0990	0.0977	0.0956	0.1283	**0.0855**
Energy	MAE	0.0513	0.0467	0.0615	0.0648	**0.0458**
	RMSE	0.0776	**0.0676**	0.0765	0.0821	0.0683
Wine	MAE	0.1015	0.1007	0.1082	0.1132	**0.1004**
	RMSE	0.1329	**0.1308**	0.1348	0.1475	0.1311
Concrete CS	MAE	0.0962	0.0928	**0.0810**	0.0840	0.0824
	RMSE	0.1173	0.1136	**0.1008**	0.1058	0.1029
Robot Arm	MAE	0.0059	0.0055	0.0345	0.0024	**0.0016**
	RMSE	0.0076	0.0072	0.0425	0.0029	**0.0020**

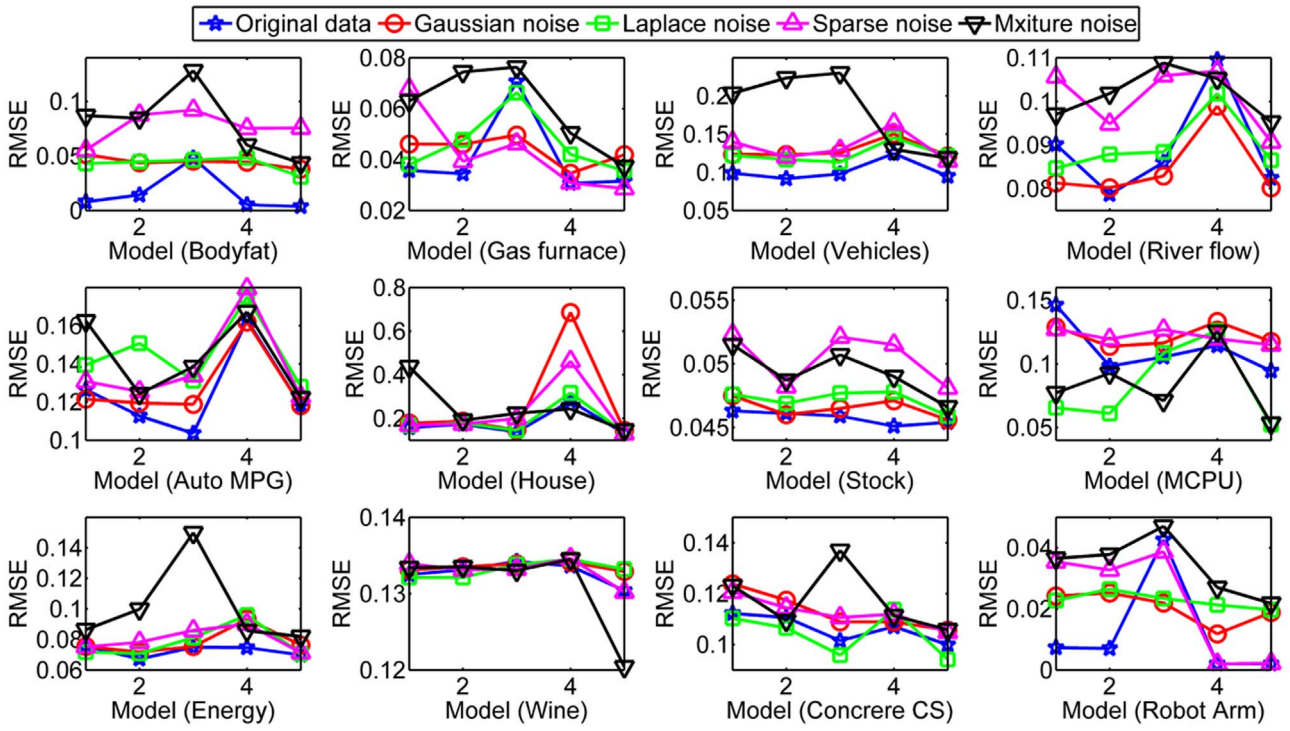

Fig. 4. RMSE of all models on real-world datasets with training samples polluted by different types of noise (Model 1-WLSSVM, Model 2-LSSVM, Model 3-SVM, Model 4-LR-MOG, Model 5-SNLR-MoG).

without outliers, while LR-MOG and SNLR-MoG are trained with outliers.

From Table 7, under MoG noise distribution assumption, LR-MoG and SNLR-MoG trained with outliers can get better regression results or obtain comparable results with conventional nonlinear regression models trained without outliers in most cases. Generally, SNLR-MoG perform better than LR-MoG, the reason may be that the nonlinear relationship between inputs and target.

Figs. 4 and 5 show the results of all models under different types of noises. Concluded from them, when using the original data of all benchmark datasets, the proposed model SNLR-MoG can outperform other compared models in half of real-world benchmark datasets used in this paper. When the samples in training phase are polluted by Gaussian noise, SNLR-MoG model holds the best performance among

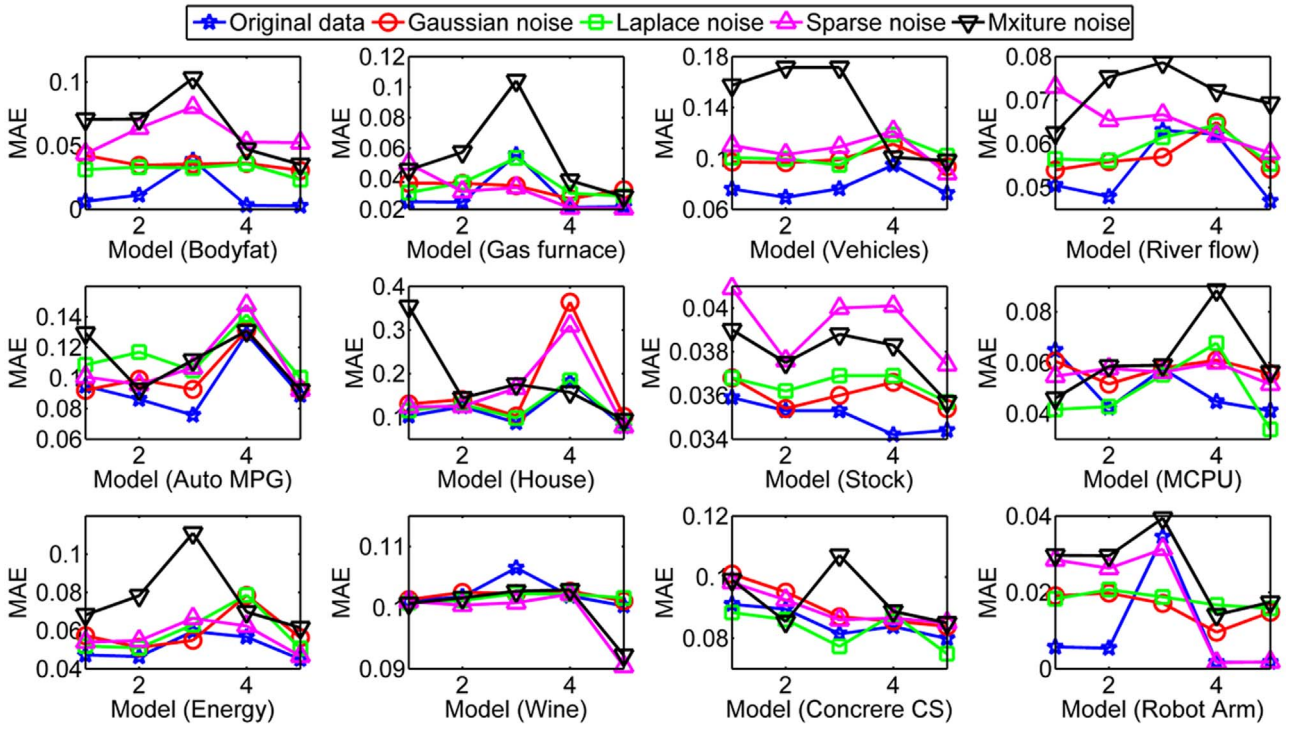

Fig. 5. MAE of all models on real-world datasets with training samples polluted by different types of noise (Model 1-WLSSVM, Model 2-LSSVM, Model 3-SVM, Model 4-LR-MOG, Model 5-SNLR-MoG).

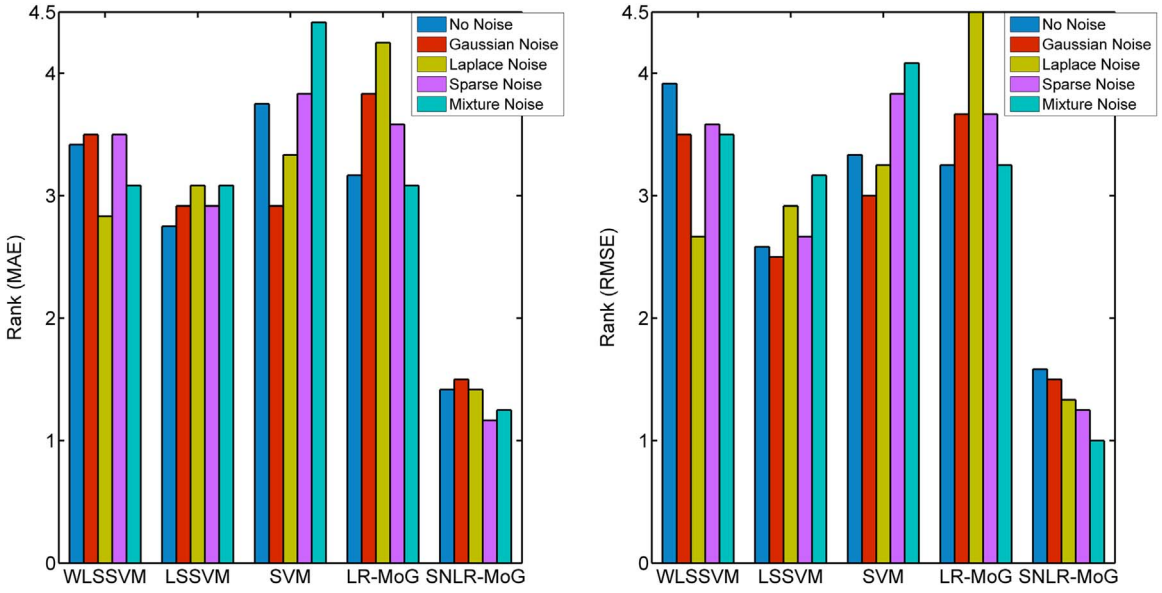

Fig. 6. Average rank of all models based on RMSE and MAE on real-world benchmark datasets with different types of noise.

Table 8
Results of models with or without outliers.

Error	Regression model			
	BPNN	MLSSVM	MSVM	MNLR-MOG
MAE	0.0143	8.5667e−04	0.0308	4.7551e−04
RMSE	0.0185	0.0012	0.0444	6.9901e−04

all regression models in seven of real-world benchmark datasets. However, SNLR-MoG model nearly holds the best performance among the all regression models, except for Vehicles, River flow and Wine, when the Laplace noises are added into training samples. When the training samples are contaminated with Sparse noise, our proposed model SNLR-MoG outperform other compared models in ten real-world benchmark datasets, while for Mixture noise, SNLR-MoG hold advantages among all regression models nearly in all real-world benchmark datasets in terms of RMSE. Fig. 6 presents the average rank of all models based on RMSE and MAE on real-world benchmark datasets with different types of noise.

From Fig. 6, in terms of RMSE and MAE, the average ranks of proposed model SNLR-MoG on twelve real-world benchmark datasets are all higher than other eight compared regression models under no matter what types of noise conditions. Moreover, when using the original benchmark datasets, namely no noises are added into training samples, the SNLR-MoG also get better performance than other regression models. Therefore, in real applications, the SNLR-MoG is more useful whether the samples are contaminated by noise or not.

Table 10
Descriptions of all data sets used in this subsection.

Dataset	Samples	Features	Target	(♯train)	(♯test)
OES97	334	263	16	280	54
OES10	403	298	16	320	83
EDM	154	16	2	110	44
WQ	1060	16	14	800	260
Polymer	61	10	4	41	20

Table 11
Results of models with or without outliers.

Data	Error	Regression model				
		LR	BPNN	MLSSVM	MSVM	MNLR-MOG
OES97	MAE	0.1716	0.0765	0.0396	0.0425	0.0384
	RMSE	0.3200	0.1606	0.1061	0.0872	0.0593
OES10	MAE	0.0852	0.0691	0.0177	0.0240	0.0184
	RMSE	0.1287	0.1092	0.0272	0.0356	0.0211
EDM	MAE	0.1682	0.2439	0.1140	0.1108	0.0192
	RMSE	0.2137	0.3276	0.1843	0.1985	0.0091
WQ	MAE	0.1706	0.1842	0.1634	0.1360	0.0888
	RMSE	0.2297	0.2620	0.2247	0.2321	0.1399
Polymer	MAE	0.0628	0.1008	0.0542	0.0850	0.0111
	RMSE	0.0822	0.1400	0.0727	0.1039	0.0141

Table 9
Performance companions between models under different types of noise.

Models	Gaussian noise		Laplace noise		Sparse noise		Mixture noise	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
BPNN	0.0685	0.0854	0.0670	0.0887	0.0467	0.0598	0.1057	0.1422
MLSSVM	**0.0442**	**0.0564**	0.0509	0.0629	0.0357	0.0443	0.0800	0.1034
MSVM	0.0565	0.0731	0.0668	0.0851	0.0446	0.0576	0.1307	0.1679
MNLR-MOG	0.0490	0.0570	**0.0444**	**0.0543**	**0.0295**	**0.0369**	**0.0447**	**0.0608**

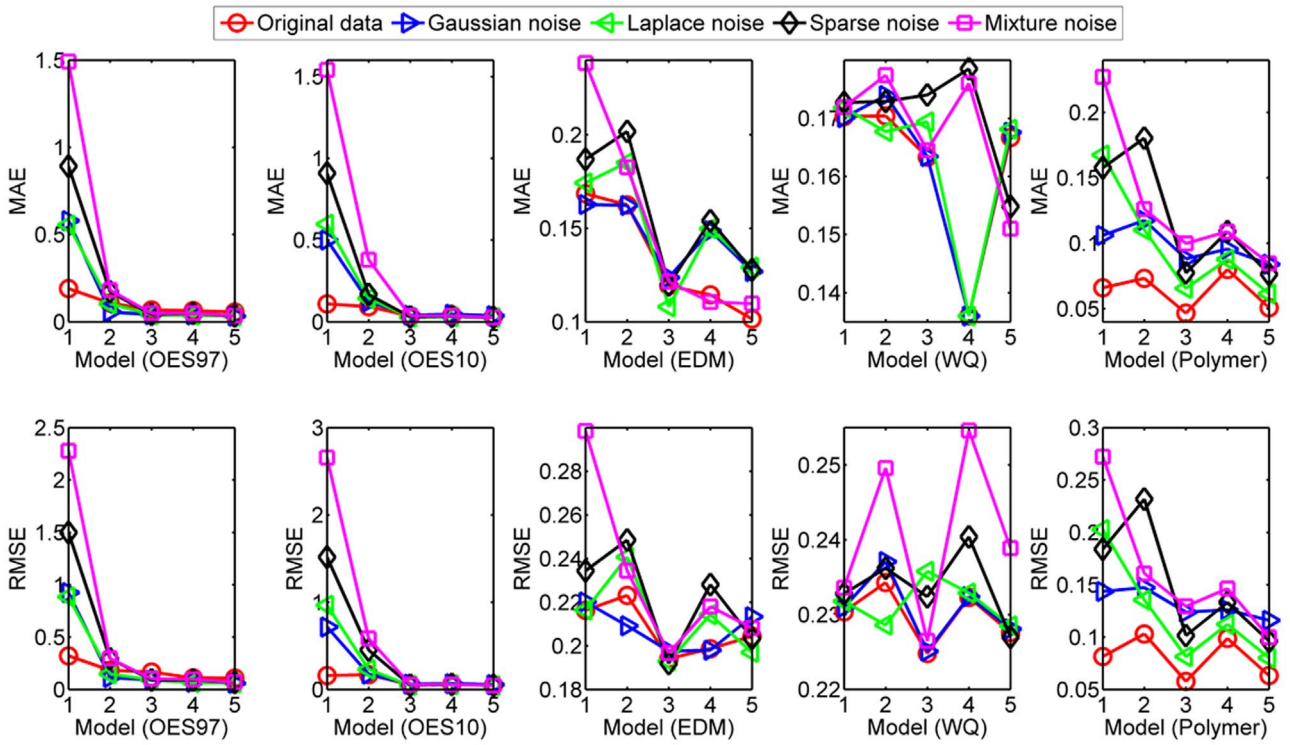

Fig. 7. Results of multi-task models with training samples polluted by different types of noise (Model 1-LR, Model 2-BPNN, Model 3-MLSSVM, Model 4-MSVM, Model 5-MNLR-MOG).

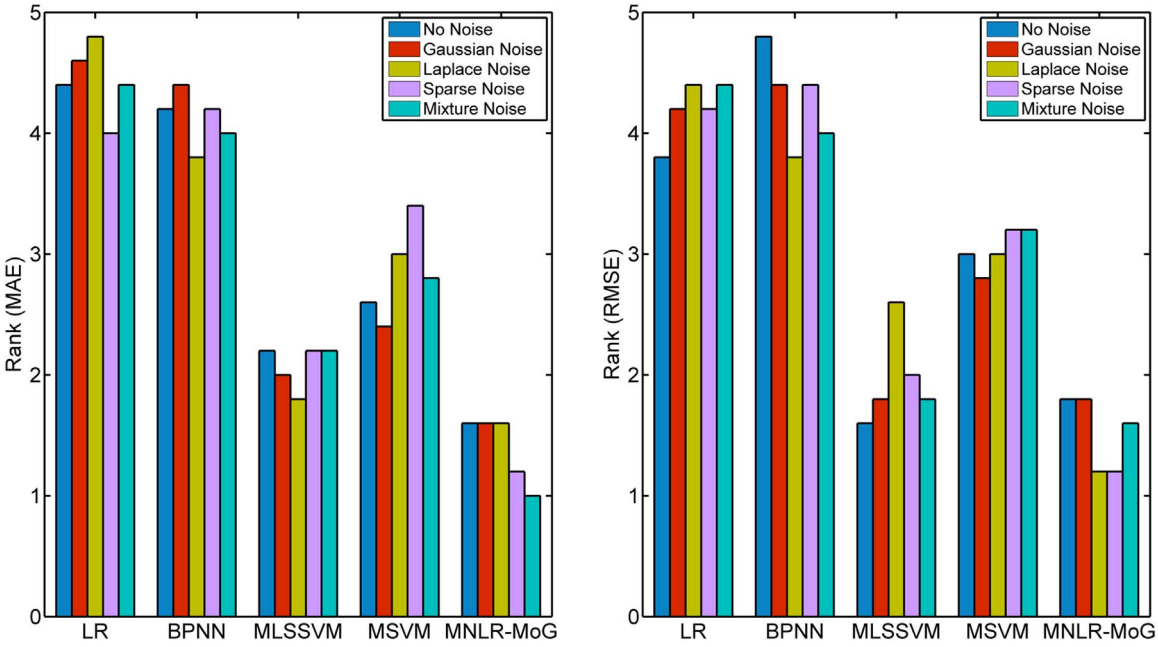

Fig. 8. Average rank of all models based on RMSE and MAE on real-world benchmark datasets with different types of noise.

5.2. Multi-task regression results of MNLR-MoG

5.2.1. Synthetic datasets

Here, we use the series of Mackey-Glass samples generated by Eq. (41), and $x(t)$, $x(t-6)$, $x(t-12)$, $x(t-18)$ are also taken as inputs to forecast $x(t+\Delta t_1)$, $x(t+\Delta t_2)$, $x(t+\Delta t_3)$. Here, we let $\Delta t_1 = 30$, $\Delta t_2 = 40$, $\Delta t_3 = 50$. In our experiment, 300 samples are selected to training the proposed MNLR-MoG model, and 500 samples are employed as test set. In training phase, three target variables are added with four different types of noise described in Table 3 and outliers, and the test samples are noise-free. The performance of the proposed multi-task regression model is compared with other multi-task models (BPNN, MLSSVM, MSVM [47]). The results of all models are shown in Tables 8 and 9.

Results in Table 8 show that the proposed MNLR-MoG trained with outliers performs better than other considered multi-task regression models trained without outliers. From Table 9, we can see that the proposed MNLR-MoG outperforms other compared models under Laplace, Sparse and mixture noise. When the training samples are polluted by Gaussian noise, MLSSVM will perform better than MNLR-

MoG. However, the performance of MNLR-MoG can be compared with WLSSVM model.

5.3. Real-world benchmark datasets

For testing the performance of proposed MNLR-MoG in real world, experiments are conducted on several real-world benchmark datasets: OES97, OES10, EDM, WQ from http://mulan.sourceforge.net/datasets-mtr.html, Polymer data set from ftp://ftp.cis.upenn.edu/pub/ungar/chemdata/. Detailed information about these datasets are presented in Table 10.

Before conducting all experiments, all the data will be normalized by Eq. (42). Table 11 show the results of proposed model trained with outliers and compared model trained without outliers. From Table 11, it can be seen that MNLR-MOG is perform better than other compared models in nearly all five datasets in terms of RMSE.

Training samples are corrupted by different noises generated by the same method described in section 5.1.2. Fig. 7 presents the results of LR, BPNN, MLSSVM, MSVM and MNLR-MoG under different types of noise. Regression results show that when there is no noise added into training samples, and when training samples are polluted by Gaussian noise, MNLR-MoG performs the best in OES97 and OES10. When Laplace noise and Sparse noise are added into training samples, MNLR-MoG outperforms other four models in nearly all datasets except for EDM. And in terms of MAE, the proposed model performs the best in all datasets when training samples are polluted by mixture noise. The average ranks of LR, BPNN, MLSSVM, MSVM and MNLR-MoG are presented in Fig. 8.

Concluded from Fig. 8, in terms of RMSE, the performance of the MLSSVM model is a little better than performance of the proposed MNLR-MoG model when the training samples are noise-free and contaminated with Gaussian noise. However, MNLR-MoG will outperform other models in terms of MAE. On the whole, in real applications, when the type of noise is unknown or complex, the proposed multi-task model can be taken into consideration.

6. Conclusions and future work

The noise in data is usually considered as Gaussian. However, no prior knowledge of noise is given in practice, while the real distribution of noise is complex. Therefore, it is not suitable to assume that the noise is Gaussian in many applications. Inspiring by the theory that any continuous distribution can be approximated by MoG, novel self-adaptive robust single-task and multi-task nonlinear regression model are proposed based on MoG to combat with the effect of unknown and complex noise or outliers.

Two groups of experiments on synthetic and real-world datasets are conducted to test the validity of the proposed SNLR-MoG and MNLR-MoG model. Results show that the performances of SNLR-MoG and MNLR-MoG generally either outperform or can be compared with conventional regression models under different types of noise or outliers. Therefore, SNLR-MoG and MNLR-MoG model can be considered as effective methods to reduce the influence of complex noise and outliers for nonlinear regression problems. Moreover, when using the original data, the proposed models may also output better performance than conventional regression models. Therefore, in reality, the proposed models may provide alternative but effective solutions to nonlinear regression problems when the data is polluted by unknown and complex noise or outliers.

In the future, considering the limited approximation ability of MoG for complex noise, other mixture distributions will be considered to model the real noise. In addition, some improved EM algorithms will be taken into consideration for speeding up the training rate.

Appendix A. Supplementary data

Supplementary data associated with this article can be found in the online version at http://dx.doi.org/10.1016/j.neucom.2017.01.024.

References

[1] Q.H. Hu, S.G. Zhang, Z.X. Xie, J.S. Mi, J. Wan, Noise model based v-support vector regression with its application to short-term wind speed forecasting, Neural Netw. 57 (2014) 1–11.
[2] Y. Anzai, Pattern Recognition and Machine Learning, Academic Press, Inc., New York, 1989.
[3] C. Pierdziochn, M. Risse, S. Rohloff, A real-time quantile-regression approach to forecasting gold returns under asymmetric loss, Resour. Policy 45 (2015) 299–306.
[4] K.P. Lin, P.F. Pai, Solar power output forecasting using evolutionary seasonal decomposition least-square support vector regression, J. Clean. Prod. (2015) 1–7.
[5] J.J. Qian, L. Luo, J. Yang, F.L. Zhang, Z.C. Lin, Robust nuclear norm regularized regression for face recognition with occlusion, Pattern Recognit. 48 (2015) 3145–3159.
[6] G. Papageorgiou, P. Bouboulis, S. Theodoridis, Robust linear regression analysis-A greedy approach, IEEE Trans. Signal Process. 63 (15) (2015) 3872–3887.
[7] Q. Hu, S. Zhang, M. Yu, Z. Xie, Short-term wind speed or power forecasting with heteroscedastic support vector regression, IEEE Trans. Sustain. Energy 7 (1) (2016) 241–249.
[8] D. Meng, F.D.L. Torre, Robust matrix factorization with unknown noise, Comput. Vis. (ICCV) (2013).
[9] Penalized Weighted Least Squares for Outlier Detection 516 and Robust Regression. arXiv preprint arXiv:1603.07427arXiv:1603.07427 (2016).
[10] H. Zhu, H. Leung, Z. He, A variational Bayesian approach to robust sensor fusion based on Student-t distribution, Inform. Sci. 221 (2013) 201–214.
[11] S. Weisberg, Applied Linear Regression 528, John Wiley and Sons, 2005.
[12] M. Qi, Z. Fu, F. Chen, Outliers detection method of multiple measuring points of parameters in power plant units, Appl. Therm. Eng. 85 (2015) 297–303.
[13] M. Meloun, J. Militký, M. Hill, R.G. Brereton, Crucial problems in regression modelling and their solutions, Analyst 127 (2002) 433–450.
[14] M. Meloun, J. Militký, Detection of single influential points in OLS regression model building, Anal. Chim. Acta 439 (2001) 169–191.
[15] M. Meloun, J. Militký, Statistical Data Analysis: A Practical Guide, Woodhead Publishing, 2011.
[16] M.H. Bhuyan, D.K. Bhattacharyya, J.K. Kalita, A multi-step outlier-based anomaly detection approach to network-wide traffic, Inf. Sci. 348 (2016) 243–271.
[17] M. Thottan, C. Ji, Anomaly detection in IP networks, IEEE Trans. Signal Process. 51 (8) (2003) 2191–2204.
[18] E.M. Knorr, R.T. Ng, V. Tucakov, Distance-based outliers: algorithms and applications, VLDB J. 8 (3–4) (2000) 237–253.
[19] A. Koufakou, M. Georgiopoulos, A fast outlier detection strategy for distributed high-dimensional data sets with mixed attributes, Data Min. Knowl. Discov. 20 (2) (2010) 259–289.
[20] F. Shaari, A.A. Bakar, A.R. Hamdan, Outlier detection based on rough sets theory, Intell. Data Anal. 13 (2) (2009) 191–206.
[21] J.A.K. Suykens, J. De Brabanter, L. Lukas, J. Vandewalle, Weighted least squares support vector machines: robustness and sparse approximation, Neurocomputing 48 (2002) 85–105.
[22] W. Wen, Z. Hao, X. Yang, A heuristic weight-setting strategy and iteratively updating algorithm for weighted least-squares support vector regression, Neurocomputing 71 (2008) 3096–3103.
[23] K. Brabanter, K. Pelckmans, J. Brabanter, M. Debruyne, J.A.K. Suykens, M. Hubert, B.D. Moor, Robustness of kernel based regression: a comparison of iterative weighting schemes, Artif. Neural Netw. ICANN 5768 (2009) 100–110.
[24] C.F. Chen, C.Q. Yan, Y.Y. Li, A robust weighted least squares support vector regression based on least trimmed squares, Neurocomputing 168 (2015) 941–946.
[25] X. Chen, J. Yang, J. Liang, Q. Ye, Recursive robust least squares support vector regression based on maximum correntropy criterion, Neurocomputing 97 (2012) 63–73.
[26] X. Yang, L. Tan, L. He, A robust least squares support vector machine for regression and classification withnoise, Neurocomputing 140 (2014) 41–52.
[27] K.N. Wang, P. Zhong, Robust non-convex least squares loss function for regression with outliers, Knowl.-Based Syst. 71 (2014) 290–302.
[28] Y.L. He, Q.X. Zhu, A novel robust regression model based on functional link least square (FLLS) and its application to modeling complex chemical processes, Chem. Eng. Sci. 153 (2016) 117–128.
[29] Y.F. Ye, L. Bai, X.Y. Hua, Y.H. Shao, Z. Wang, N.Y. Deng, Weighted Lagrange ε-twin support vector regression, Neurocomputing 197 (2016) 53–68.
[30] J. Hu, K. Zheng, A novel support vector regression for data set with outliers, Appl. Softw. Comput. 31 (2015) 405–411.
[31] G.J. McLachlan, K.E. Basford, Mixture Models: inference and Applications to Clustering, Marcel Dekker, 1988.
[32] V. Mazya, G. Schmidt, On approximate approximations using Gaussian kernels, IMA J. Numer. Anal. 16 (1) (1996) 13–29.
[33] Y.X. Zhao, X.H. Zhuang, S.J. Ting, Gaussian mixture density modeling of non-gaussian source for autoregressive process, IEEE Trans. Signal Process. 43 (1995) 894–903.
[34] G. Galimberti, G. Soffritti, A multivariate linear regression analysis using finite mixtures of t distributions, Comput. Stat. Data Anal. 71 (2014) 138–150.

[35] C.B., Zeller, C.R.B. Cabral, V.H. Lachos, Robust mixture regression modeling based on scale mixtures of skew-normal distributions, TEST-Springer, 2015.
[36] V.G. Cancho, D.K. Dey, V.H. Lachos, M.G. Andrade, Bayesian nonlinear regression models with scale mixtures of skew-normal distributions: estimation and case influence diagnostics, Comput. Stat. Data Anal. 55 (2011) 588–602.
[37] A.M. Garay, V.H. Lachos, C.A.A. Valle, Nonlinear regression models based on scale mixtures of skew-normal distributions, J. Korean Stat. Soc. 40 (2011) 115–124.
[38] V.H. Lachos, D. Bandyopadhyay, A.M. Garay, Heteroscedastic nonlinear regression models based on scale mixtures of skew-normal distributions, Stat. Probab. Lett. 81 (2011) 1208–1217.
[39] G. Garg, G. Prasad, D. Coyle, Gaussian mixture model-based noise reduction in resting state fMRI data, J. Neurosci. Methods 215 (2013) 71–77.
[40] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood fro min complete data via the EM algorithm (with discussion), J. R. Stat. Soc.-Ser. B39 (1977) 1–38.
[41] G.J. McLachlan, S.K. Ng, The Top Ten Algorithms in Data Mining (2009) 93–115.
[42] X.J. Peng, Y.F. Wang, A normal least squares support vector machine (NLS-SVM) and its learning algorithm, Neurocomputing 72 (2009) 3734–3741.
[43] S. Xu, X. An, X.D. Qiao, L.J. Zhu, L. Li, Multi-output least-squares support vector regression machines, Pattern Recognit. Lett. 34 (2013) 1078–1084.
[44] P.J. Garca Nieto, E. Garca Gonzalo, J.R. Alonso Fernndez, C. Daz Muniz, A hybrid PSO optimized SVM based model for predicting a successful growth cycle of the Spirulina platensis from raceway experiments data, J. Comput. Appl. Math. 291 (1) (2016) 293–303.
[45] S.Y. Wong, K.S. Yap, H.J. Yap, A Constrained Optimization based Extreme Learning Machine for noisy data regression. Neurocomputing ⟨http://dx.doi.org/10.1016/j.neucom.2015.07.065⟩.
[46] J.S. Zhao, X.J. Yu, Adaptive natural gradient learning algorithms for Mackey-Glass chaotic time prediction, Neurocomputing 157 (2015) 41–45.
[47] D. Tuia, J. Verrelst, L. Alonso, F. Perez-Cruz, G. Camps-Valls, Multioutput support vector regression for remote sensing biophysical parameter estimation, IEEE Geosci. Remote Sens. Lett. 8 (2011) 804–808.

Haibo Wang, received the B.S. degree in School of Mathematics and Computer Science, Hubei University, China, in 2002. He received the M.S. degree and Ph.D. from State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing of Wuhan University, China, in 2005 and 2014, respectively. Since 2005, he has been a teacher of Hubei University of Technology.

Yun Wang, received the B.S. degree in Economics School of Anhui University, China, 2012. He received the M.S. degree from school of mathematics and statistics of Lanzhou University, China, in 2012. Now, he is currently a Ph.D. student with the School of Computer Science and Technology in Tianjin University, China. His interests include wind speed/wind power forecasting, robust regression modeling, multi-kernel learning, functional data analysis and Bayesian inference.

Qinghua Hu, received the B.S., M.S.and Ph.D degrees from the Harbin Institute of Technology, Harbin, China, in 1999, 2002, and 2008, respectively. He was a Post-Doctoral Fellow with the Department of Computing, Hong Kong Polytechnic University, Hong Kong, from 2009 to 2011. He is currently a Full Professor and the Vice Dean of the School of Computer Science and Technology, Tianjin University, Tianjin, China. He has authored over 100 journal and conference papers in the areas of granular computing based machine learning, reasoning with uncertainty, pattern recognition, and fault diagnosis. His current research interests include rough sets, granular computing, and data mining for classification and regression. Prof. Hu was the Program Committee Co-Chair of the International Conference on Rough Sets and Current Trends in Computing in 2010, the Chinese Rough Set and Soft Computing Society in 2012 and 2014, and the International Conference on Rough Sets and Knowledge Technology, the International Conference on Machine Learning and Cybernetics in 2014, and the General Co-Chair of IJCRS 2015. He is now PC-Co Chairs of CCML 2017 and CCCV 2017.