Received May 8, 2020, accepted June 3, 2020, date of publication June 10, 2020, date of current version June 24, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3001371

Robust Task Learning Based on Nonlinear Regression With Mixtures of Student-*t* Distributions

CHUNZHENG CAO[1], ZIYUE WANG[1], JIAN QING SHI[2], AND YUNJIE CHEN[1], (Member, IEEE)
[1]School of Mathematics and Statistics, Nanjing University of Information Science and Technology, Nanjing 210044, China
[2]School of Mathematics, Statistics, and Physics, Newcastle University, Newcastle upon Tyne NE1 7RU, U.K.

Corresponding author: Chunzheng Cao (caochunzheng@nuist.edu.cn)

This work was supported in part by the Major Project of the National Social Science Foundation of China under Grant 16ZDA047, in part by the Natural Science Foundation of Jiangsu Province of China under Grant BK20191394, in part by the National Natural Science Foundation of China under Grant 61672291, and in part by the Key Statistical Research Project of Jiangsu Province of China under Grant 2019A005.

ABSTRACT We propose a robust task learning method based on nonlinear regression model with mixtures of t-distributions. The model can adaptively reduce the effects of complex noises and accurately learn the nonlinear structure of targets. By introducing latent variables, the model is expressed into a hierarchical structure, which helps explain the advantage of flexibility compared to the traditional Gaussian based learning model. We develop a two-stage efficient estimation procedure to obtain penalized likelihood estimator of the parameters combined an expectation-maximization algorithm with Lagrange multiplier method. The learning performances of the model are investigated through experiments on both synthetic and real data sets.

INDEX TERMS Task learning, robust nonlinear regression, outlier, EM algorithm, kernel method.

I. INTRODUCTION

Statistical regression is one of the important methods in task learning. Traditional regression models are established under the assumption of normality. However, the Gaussian regression models are insufficient when the noises deviate from the normal distribution. Moreover, the statistical inference of Gaussian regression is sensitive to outliers. The finite mixture model based on mixture distributions can alleviate the problem caused by misspecification of distribution to some extent. For example, the mixture of Gaussians (MoG) [1]–[3] has been proposed for the purposes of regression, clustering, denoising, segmentation, recognition, learning and prediction [4]–[10]. In order to enhance the robustness, the mixture of t-distributions (MoT) has been developed in [11], and extensively studied from various perspectives (see in [12]–[14], among others). The t-distribution is a kind of heavy-tailed distribution, which can adaptively reduce the influence of outliers to achieve desired robust inference. Also, the t-distribution can be viewed as an important member of the distribution family called scale mixtures of normal (SMN) distribution [15]. Recently, SMN distributions have been used in various regression models [16]–[19], which extending the theory of Gaussian models from the framework of distribution family.

In this paper, we introduce a robust learning method for both single task and multi-tasks within the MoT framework. The complex noises in the tasks are adaptively described by MoT. Since the noises in the tasks may come from multiple sources and the data may contain outliers or local variation, the MoT models will be able to take full advantage of adaptability and robustness. How to accurately estimate the unknown nonlinear regression functions in task learning is usually very challenging. To address the problem, we propose a strategy by decomposing the MoT model into a hierarchical structure with three layers. In the bottom layer, the t-distribution is expressed as a SMN distribution by a latent scale variable. Through the latent scale variable, the heavy-tailed feature is created and the weights of outliers in the parameter estimation are reduced automatically. In the middle layer, the mixture distribution is generated by latent labels, which can adaptively allocate different patterns in tasks to prevent overfitting. In the top layer, the nonlinear regression functions in tasks are learned through a kernel method [20],

The associate editor coordinating the review of this manuscript and approving it for publication was Donatella Darsena.

which can avoid the computational complexity caused by high dimensional approximation. When the degrees of freedom in MoT tend to infinity, the MoT model will degenerate into the MoG [21].

Several works on multi-task learning are devoted to sharing task information from different perspectives to improve learning performance [22]–[24]. In regression analysis, this can be realised via shared parameters and task-specific parameters. The shared parameters load the information of commonality while the individual parameters carry the information of speciality. Our method is a parameter-based multi-task learning [25] but via a decomposition. In the new MoT model, the distribution parameters and hyper parameters are shared among tasks which can effectively reduce the risk of over-fitting. This design is particularly effective when some tasks have insufficient information or there are outliers in sparse areas. Through the three-layer hierarchical structure of the model, we propose to use a two-stage estimation procedure combined an EM algorithm with Lagrange multiplier method. Specifically, the parameters in MoT are updated in the first stage using EM-type algorithms. The high dimensional map functions are calculated in the second stage using optimization with penalties through Lagrange multiplier method.

The paper is organized as follows. Section II reviews the task learning regression from likelihood framework. Section III defines the MoT based regression for both single task and multi-task learning. The hierarchical structure and the estimation procedures are also discussed in this section. Section IV includes the experimental results on both synthetic data sets and real data sets. Section V concludes the paper with some discussions.

II. PRELIMINARIES

A. SINGLE TASK REGRESSION

Generally, a nonlinear regression for single-task can be expressed as

$$y = f(\boldsymbol{x}) + \varepsilon, \tag{1}$$

where $f(\boldsymbol{x})$ is an unknown nonlinear regression function with d-dimensional covariate $\boldsymbol{x}$ and ε is random noise. Standard support vector machine (SVM) [26] approximates the regression by a linear function

$$f(\boldsymbol{x}) = \boldsymbol{\varphi}^\top(\boldsymbol{x})\boldsymbol{w} + b, \tag{2}$$

where $\boldsymbol{\varphi}(\cdot)$ is a map function mapping the original input $\boldsymbol{x}$ into h-dimensional feature space, $\boldsymbol{w} \in \mathbb{R}^h$ is the regression coefficient and $b \in \mathbb{R}$ is the bias.

Usually, the noise term ε is assumed to follow Gaussian distribution. Then, the maximum likelihood estimation (MLE) with ridge penalty on $\boldsymbol{w}$ coincides with the least squares SVM (LS-SVM) [27] by minimizing the following objective function

$$\min_{\boldsymbol{w},\, b} \frac{\gamma}{2}\|\boldsymbol{\varepsilon}\|_2^2 + \frac{1}{2}\|\boldsymbol{w}\|_2^2, \quad \text{s.t. } \boldsymbol{y} = \boldsymbol{B}^\top\boldsymbol{w} + b\mathbf{1}_n + \boldsymbol{\varepsilon}, \tag{3}$$

where $\boldsymbol{y} = (y_1, \ldots, y_n)^\top \in \mathbb{R}^n$ is the output, $\boldsymbol{B} = (\boldsymbol{\varphi}(\boldsymbol{x}_1), \ldots, \boldsymbol{\varphi}(\boldsymbol{x}_n)) \in \mathbb{R}^{h\times n}$, $\mathbf{1}_n$ stands for an n-dimensional vector with all elements being one, $\boldsymbol{\varepsilon} = (\varepsilon_1, \ldots, \varepsilon_n)^\top$ is the noise vector consisting of slack variables and γ is a positive regularized parameter.

However, real world data often have non-Gaussian noises or there are outliers. The Gaussian distribution based learning is usually quite sensitive to outliers or distribution misspecification. To develop a robust and adaptive approach, we may use a heavy-tailed distribution; i.e., we assume that $\boldsymbol{\varepsilon}$ follows a heavy-tailed distribution with probability distribution function (pdf) $p(\boldsymbol{\varepsilon})$ and the regression coefficient $\boldsymbol{w}$ has a prior $\pi(\boldsymbol{w})$. Then, a maximum a posteriori (MAP) estimate of the coefficient can be obtained by

$$\max_{\boldsymbol{w},\, b} \log p(\boldsymbol{\varepsilon}) + \log \pi(\boldsymbol{w}), \quad \text{s.t. } \boldsymbol{y} = \boldsymbol{B}^\top\boldsymbol{w} + b\mathbf{1}_n + \boldsymbol{\varepsilon}. \tag{4}$$

The idea of MAP is consistent with least squares approach. The pdfs $p(\boldsymbol{\varepsilon})$ in (4) corresponds to loss functions, while different prior $\pi(\boldsymbol{w})$ is associate with a penalty on $\boldsymbol{w}$. Different distributions can be used, for example, Laplace, Huber, beta distribution [28], MoG [21] and hyperbolic-secant distribution [29], among others. We propose to use mixture of t-distributions, or MoT, in this paper since MoT can enhance the robustness while inherits most of the advantages of Gaussian distribution.

B. MULTI-TASK REGRESSION

Suppose we need to learn m tasks simultaneously. For the j-th task, we have n_j training data $\{\boldsymbol{x}_{ij}, y_{ij}\}_{i=1}^{n_j}$, where $\boldsymbol{x}_{ij} \in \mathbb{R}^d$. The multi-task regression model is given by

$$\boldsymbol{y}_j = \boldsymbol{B}_j^\top\boldsymbol{w}_j + b_j\mathbf{1}_{n_j} + \boldsymbol{\varepsilon}_j, \quad j = 1, \ldots, m, \tag{5}$$

where $\boldsymbol{y}_j = (y_{j,1}, \ldots, y_{j,n_j})^\top$, $\boldsymbol{B}_j = (\boldsymbol{\varphi}(\boldsymbol{x}_{1,j}), \ldots, \boldsymbol{\varphi}(\boldsymbol{x}_{n_j,j}))$, $\boldsymbol{w}_j \in \mathbb{R}^h$, $\boldsymbol{\varepsilon}_j = (\varepsilon_{1,j}, \ldots, \varepsilon_{n_j,j})^\top$ and ε_{ij} follows a distribution with pdf $p(\varepsilon_{ij})$. Generally, we can solve the regression problem by minimize the minus penalized likelihood

$$\min_{\{\boldsymbol{w}_j,\, b_j\}} -\sum_{j=1}^{m} \log p(\boldsymbol{\varepsilon}_j) + \lambda\|\boldsymbol{W}\|, \quad \text{s.t. } \boldsymbol{y}_j = \boldsymbol{B}_j^\top\boldsymbol{w}_j + b_j\mathbf{1}_{n_j} + \boldsymbol{\varepsilon}_j, \quad j = 1, \ldots, m, \tag{6}$$

where $\boldsymbol{W} = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_m)$, the specific penalty terms $\lambda\|\boldsymbol{W}\|$ can determine the similarity between tasks, the sparsity of coefficients or the smoothness of nonlinear regression, etc. Alternatively, we can use prior information to explain penalty terms from the perspective of Bayesian inference. The performance of multi-task regression (5) depends on both the efficiency of optimization and the learning ability. We will show later the optimization algorithm and efficiency of the MoT-based regression.

Multi task learning focuses on the sharing of information or model structure, that is, the model space of each task is not independent. The correlation between different tasks can be reflected in many aspects, such as the commonness of regression function, the distribution of noises and so on. However, the noise's type and intensity of different tasks may be different, which can not be well described by using a simple distribution. We are therefore motivated to use a mixture distribution.

III. MoT BASED TASK REGRESSION

A. MoT BASED SINGLE-TASK REGRESSION (MoT-STR)

We assume the noise term ε in model (1) follows an MoT distribution with pdf

$$p(\varepsilon) = \sum_{k=1}^{K} \pi_k t(\varepsilon|0, \sigma_k^2, \nu), \tag{7}$$

where K is the number of components in the mixture distributions, $t(\varepsilon|0, \sigma_k^2, \nu)$ stands for the pdf of t-distribution with zero mean, scale parameter σ_k^2 and degrees of freedom ν, π_k is the mixing proportion under the constraint $\sum_{k=1}^{K} \pi_k = 1$ ($\pi_k \geqslant 0$).

The log-likelihood of training data is

$$L(\boldsymbol{\varepsilon}|\boldsymbol{\Theta}) = \sum_{i=1}^{n} \log(\sum_{k=1}^{K} \pi_k t(\varepsilon_i|0, \sigma_k^2, \nu)), \tag{8}$$

where $\boldsymbol{\Theta} = \{\pi_1, \ldots, \pi_K, \sigma_1^2, \ldots, \sigma_K^2, \nu, \boldsymbol{w}, \ b\}$ is the parameter set. Unfortunately, the calculation of MLE directly through (8) is intractable. One problem is the optimization does not have a closed form solution. Another problem comes from the computation complexity of the high-dimensional map function.

To address the first problem, we propose to use an EM algorithm. We introduce a latent component indicator $z_i = (z_{i1}, \ldots, z_{iK})^{\top}$ such that

$$z_{ik} = \begin{cases} 1, & \text{if noise } \varepsilon_i \text{ comes from component } k, \\ 0, & \text{otherwise.} \end{cases}$$

Then, we have an equivalent hierarchical structure of the noises as

$$\begin{aligned} \varepsilon_i|u_i, z_{ik} &= 1 \overset{ind}{\sim} \mathrm{N}(0, \sigma_k^2/u_i), \\ u_i &\overset{iid}{\sim} \Gamma(\nu/2, \nu/2), \\ z_i &\overset{iid}{\sim} \mathrm{Multinomial}(1, \boldsymbol{\pi}), \quad i = 1, \ldots, n, \end{aligned} \tag{9}$$

where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)^{\top}$. The hierarchical structure (9) will help us to develop an efficient EM algorithm which will be detailed next.

To address the second problem, we use the Lagrangian multiplier method in the optimize step through a kernel function. Since the map function always appears as $\varphi(\boldsymbol{x})^{\top}\varphi(\boldsymbol{x})$, we can replace the calculation by a kernel function

$$K(\boldsymbol{x}, \boldsymbol{y}) = \varphi(\boldsymbol{x})^{\top}\varphi(\boldsymbol{y}). \tag{10}$$

We therefore don't have to specify each map function. In this paper, we employ the popular Gaussian radial basis function (RBF) kernel $K(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\frac{\|\boldsymbol{x}-\boldsymbol{y}\|^2}{2\sigma^2})$.

The log-likelihood of the complete data $\mathcal{D}_c = \{\varepsilon_i, u_i, z_i\}_{i=1}^{n}$ is given by

$$\begin{aligned} l(\boldsymbol{\Theta}|\mathcal{D}_c) = \sum_{i=1}^{n}\sum_{k=1}^{K} z_{ik}\{&\log \pi_k + \log p(u_i; \nu) \\ &+ \frac{1}{2}\log u_i - \frac{1}{2}\log(2\pi\sigma_k^2) - \frac{1}{2\sigma_k^2} u_i \varepsilon_i^2\}, \end{aligned} \tag{11}$$

where $p(u_i; \nu)$ is the pdf of u_i which follows the distribution $\Gamma(\nu/2, \nu/2)$.

E-step: Let $\boldsymbol{\Theta}^{(t)}$ be the current estimation of $\boldsymbol{\Theta}$ in the t-th step. We calculate the Q-function by

$$\begin{aligned} Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(t)}) = \sum_{i=1}^{n}\sum_{k=1}^{K} \gamma_{ik}^{(t)}[\log \pi_k - \frac{1}{2}\log \sigma_k^2] \\ - \sum_{i=1}^{n}\sum_{k=1}^{K} \eta_{ik}^{(t)} \frac{\varepsilon_i^2}{2\sigma_k^2}, \end{aligned} \tag{12}$$

where

$$\gamma_{ik}^{(t)} = \mathrm{E}(z_{ik}|\varepsilon_i, \boldsymbol{\Theta}^{(t)}) = \frac{\pi_k t(\varepsilon_i|0, \sigma_k^2, \nu)}{\sum_{k=1}^{K} \pi_k t(\varepsilon_i|0, \sigma_k^2, \nu)}\bigg|_{\boldsymbol{\Theta}^{(t)}}, \tag{13}$$

and

$$\eta_{ik}^{(t)} = \mathrm{E}(u_i z_{ik}|\varepsilon_i, \boldsymbol{\Theta}^{(t)}) = \gamma_{ik}^{(t)} \frac{(\nu+1)\sigma_k^2}{\nu\sigma_k^2 + \varepsilon_i^2}\bigg|_{\boldsymbol{\Theta}^{(t)}}. \tag{14}$$

M-step: Maximizing $Q(\boldsymbol{\Theta}|\boldsymbol{\Theta}^{(t)})$ with respect to $\boldsymbol{\Theta}$, we can update the estimate $\boldsymbol{\Theta}^{(t+1)}$ by

$$\pi_k^{(t+1)} = \frac{1}{n}\sum_{i=1}^{n} \gamma_{ik}^{(t)}, \tag{15}$$

$$\sigma_k^{2(t+1)} = \frac{\sum_{i=1}^{n} \eta_{ik}^{(t)} \varepsilon_i^{2(t)}}{\sum_{i=1}^{n} \gamma_{ik}^{(t)}}, \quad k = 1, \ldots, K. \tag{16}$$

Following the ECME algorithm [30], the degree of freedom ν can be updated in $\boldsymbol{\Theta}$ alone from its marginal log-likelihood, i.e.

$$\nu^{(t+1)} = \arg\max_{\nu} \sum_{i=1}^{n} \log\{\sum_{k=1}^{K} \pi_k^{(t)} t(\varepsilon_i^{(t)}|0, \sigma_k^{2(t)}, \nu)\}. \tag{17}$$

Alternatively, we may update ν together with other parameters in the EM algorithm [11], [12], i.e.

$$\nu^{(t+1)} = \arg\max_{\nu} \sum_{i=1}^{n} \mathrm{E}(\log p(u_i; \nu)|\varepsilon_i, \boldsymbol{\Theta}^{(t)}). \tag{18}$$

The regression coefficients $\boldsymbol{w}$ and the bias b can be obtained by minimizing the following objective function

$$\begin{aligned} &\min_{\boldsymbol{w}, b} \frac{\lambda}{2}\|\boldsymbol{\eta} \odot \boldsymbol{\varepsilon}\|_2^2 + \frac{1}{2}\|\boldsymbol{w}\|_2^2, \\ &\text{s.t. } \boldsymbol{y} = \boldsymbol{B}^{\top}\boldsymbol{w} + b\boldsymbol{1}_n + \boldsymbol{\varepsilon}, \end{aligned} \tag{19}$$

where $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_n)^{\top}$ is weight vector with $\eta_i = \sqrt{\sum_{k=1}^{K} \frac{\eta_{ik}}{2\sigma_k^2}}$, $\odot$ means the Hadamard product.

Similar to LS-SVM, we employ Lagrange multiplier method to solve the optimization problem (19). The Lagrangian function can be expressed as

$$L(\boldsymbol{w}, b, \boldsymbol{\varepsilon}, \boldsymbol{\alpha}) = \frac{\lambda}{2}\|\boldsymbol{\eta} \odot \boldsymbol{\varepsilon}\|_2^2 + \frac{1}{2}\|\boldsymbol{w}\|_2^2 - \boldsymbol{\alpha}^\top(\boldsymbol{\varepsilon} - \boldsymbol{y} + \boldsymbol{B}^\top \boldsymbol{w} + b\boldsymbol{1}_n), \tag{20}$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_n)^\top \in \mathbb{R}^n$ is a vector consisting of Lagrange multipliers. According to the KKT conditions, the solution of the problem (19) is determined by solving the linear equations

$$\begin{cases} \dfrac{\partial L}{\partial \boldsymbol{w}} = \boldsymbol{0} \Longrightarrow \boldsymbol{w} = \boldsymbol{B}\boldsymbol{\alpha} \\ \dfrac{\partial L}{\partial b} = 0 \Longrightarrow \boldsymbol{\alpha}^\top \boldsymbol{1}_n = 0 \\ \dfrac{\partial L}{\partial \boldsymbol{\varepsilon}} = \boldsymbol{0} \Longrightarrow \operatorname{diag}(\boldsymbol{\varepsilon}) = \lambda^{-1}\operatorname{diag}(\boldsymbol{\eta})^{-2}\operatorname{diag}(\boldsymbol{\alpha}) \\ \dfrac{\partial L}{\partial \boldsymbol{\alpha}} = \boldsymbol{0} \Longrightarrow \boldsymbol{\varepsilon} - \boldsymbol{y} + \boldsymbol{B}^\top \boldsymbol{w} + b\boldsymbol{1}_n = \boldsymbol{0}. \end{cases} \tag{21}$$

By eliminating $\boldsymbol{w}$ and $\boldsymbol{\varepsilon}$, one can obtain the solution by the following linear system

$$\begin{bmatrix} 0 & \boldsymbol{1}_n^\top \\ \boldsymbol{1}_n & \boldsymbol{H} \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{y} \end{bmatrix}, \tag{22}$$

where $\boldsymbol{H} = \boldsymbol{K} + \lambda^{-1}\operatorname{diag}(\boldsymbol{\eta})^{-2}$, $\boldsymbol{K} = \boldsymbol{B}^\top \boldsymbol{B}$ is defined by the kernel function with elements $k_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Let the solution of (22) be b^* and $\boldsymbol{\alpha}^*$. Then the regression function is obtained as

$$\hat{f}(\boldsymbol{x}) = \varphi(\boldsymbol{x})^\top \boldsymbol{B}\boldsymbol{\alpha}^* + b^* = \sum_{j=1}^{n} \alpha_j^* K(\boldsymbol{x}, \boldsymbol{x}_j) + b^*. \tag{23}$$

Finally, the residuals $\boldsymbol{\varepsilon}$ can be updated by

$$\varepsilon_i^{(t+1)} = y_i - \hat{f}(\boldsymbol{x}_i)\big|_{\boldsymbol{\Theta}^{(t)}}, \quad i = 1, \ldots, n. \tag{24}$$

Algorithm 1 summaries the learning procedure of MoT-STR.

Algorithm 1 Learning Procedure of MoT-STR

```
Training
Input:
1. Training data $\{\boldsymbol{x}_j, y_j\}_{j=1}^n$;
2. Number of mixing components $K$;
3. Hyperparameters $\lambda$, $\sigma^2$.
Output:
1. Optimal parameters of MoT $\nu, \pi_k, \sigma_k^2$ $(k = 1, \ldots, K)$;
2. Optimal parameters of STR $b, \boldsymbol{\alpha}$;
{Step 1} Initialize parameters $\nu, \pi_k, \sigma_k^2$ $(k = 1, \ldots, K)$;
{Step 2} while $\|\boldsymbol{\alpha}^{new} - \boldsymbol{\alpha}^{old}\| >$ threshold do
    Calculate $\gamma_{ik}$ and $\eta_{ik}$ by Eq. (13) and (14);
    Update $\pi_k, \sigma_k^2$ by Eq. (15) and (16);
    Update $\nu$ by Eq. (17) or (18);
    Update the solution $b, \boldsymbol{\alpha}$ of Eq. (22);
    Update the regression function $f(\boldsymbol{x})$ by
        Eq. (23);
    Update the error term $\boldsymbol{\varepsilon}$ by Eq. (24).
Testing
Input: $\boldsymbol{x}^*, b, \boldsymbol{\alpha}, \sigma^2$.
Output: Calculate $y^*$ by Eq. (23).
```

B. MoT BASED MULTI-TASK REGRESSION (MoT-MTR)

Suppose the error term ε_{ij} in (5) follows the MoT distribution (7). The hierarchical structure of model (5) can be expressed as

$$\begin{aligned} \varepsilon_{ij}|u_{ij}, z_{ijk} = 1 &\overset{ind}{\sim} \mathrm{N}(0, \sigma_k^2/u_{ij}), \\ u_{ij} &\overset{iid}{\sim} \Gamma(\nu/2, \nu/2), \\ \boldsymbol{z}_{ij} &\overset{iid}{\sim} \mathrm{Multinomial}(1, \boldsymbol{\pi}), \\ j = 1, \ldots, m&, i = 1, \ldots, n_j, \end{aligned} \tag{25}$$

where $\boldsymbol{z}_{ij} = (z_{ij1}, \ldots, z_{ijK})^\top$.

The log-likelihood of the complete data $\mathcal{D}_c = \{\varepsilon_{ij}, u_{ij}, \boldsymbol{z}_{ij} | j = 1, \ldots, m, i = 1, \ldots, n_j\}$ is given by

$$l(\boldsymbol{\Theta}|\mathcal{D}_c) = \sum_{j=1}^{m}\sum_{i=1}^{n}\sum_{k=1}^{K} z_{ijk}\{\log \pi_k + \log p(u_{ij}; \nu) + \frac{1}{2}\log u_{ij} - \frac{1}{2}\log(2\pi\sigma_k^2) - \frac{1}{2\sigma_k^2}u_{ij}\varepsilon_{ij}^2\}. \tag{26}$$

Similar to the derivation process of single-task, we update the parameters $\boldsymbol{\Theta}$ by

$$\pi_k^{(t+1)} = \frac{\sum_{j=1}^{m}\sum_{i=1}^{n_j}\gamma_{ijk}}{\sum_{j=1}^{m} n_j}\Bigg|_{\boldsymbol{\Theta}^{(t)}}, \tag{27}$$

$$\sigma_k^{2(t+1)} = \frac{\sum_{j=1}^{m}\sum_{i=1}^{n_j}\eta_{ijk}\varepsilon_{ij}^2}{\sum_{j=1}^{m}\sum_{i=1}^{n_j}\gamma_{ijk}}\Bigg|_{\boldsymbol{\Theta}^{(t)}}, \tag{28}$$

where

$$\gamma_{ijk} = \mathrm{E}(z_{ijk}|\varepsilon_{ij}) = \frac{\pi_k t(\varepsilon_{ij}|0, \sigma_k^2, \nu)}{\sum_{k=1}^{K}\pi_k t(\varepsilon_{ij}|0, \sigma_k^2, \nu)}, \tag{29}$$

$$\eta_{ijk} = \mathrm{E}(z_{ijk}u_{ij}|\varepsilon_{ij}) = \gamma_{ijk}\frac{(\nu + 1)\sigma_k^2}{\nu\sigma_k^2 + \varepsilon_{ij}^2}. \tag{30}$$

The regression coefficients $\boldsymbol{w}_j$'s and the biases b_j's could be obtained by minimizing the following objective function with constraints

$$\min_{\boldsymbol{w}_j, b_j} \frac{\lambda}{2}\sum_{j=1}^{m}\|\boldsymbol{\eta}_j \odot \boldsymbol{\varepsilon}_j\|_2^2 + \frac{1}{2}\sum_{j=1}^{m}\|\boldsymbol{w}_j\|_2^2,$$
$$\text{s.t. } \boldsymbol{y}_j = \boldsymbol{B}_j^\top \boldsymbol{w}_j + b_j\boldsymbol{1}_{n_j} + \boldsymbol{\varepsilon}_j, \quad j = 1, \ldots, m, \tag{31}$$

where $\boldsymbol{\eta}_j = (\eta_{1j}, \ldots, \eta_{n_j,j})^\top$ with $\eta_{ij} = \sqrt{\sum_{k=1}^{K}\frac{\eta_{ijk}}{2\sigma_k^2}}$.

In the objective functions (19) for MoT-STR and (31) for MoT-MTR, $\boldsymbol{\eta}$ are the weight of noise $\boldsymbol{\varepsilon}$. From (14) and (30), $\boldsymbol{\eta}$ is inversely proportional to the noise under the t-distribution. Therefore, MoT based learning can adaptively reduce the influence of outliers.

The Lagrangian function for the problem (31) is

$$\begin{aligned} &L(\{\boldsymbol{w}_j\}_{j=1}^m, \{b_j\}_{j=1}^m, \{\boldsymbol{\varepsilon}_j\}_{j=1}^m, \{\boldsymbol{\alpha}_j\}_{j=1}^m) \\ &\quad = \frac{\lambda}{2}\sum_{j=1}^m \|\boldsymbol{\eta}_j \odot \boldsymbol{\varepsilon}_j\|_2^2 + \frac{1}{2}\sum_{j=1}^m \|\boldsymbol{w}_j\|_2^2 \\ &\qquad - \sum_{j=1}^m \boldsymbol{\alpha}_j^\top(\boldsymbol{\varepsilon}_j - \boldsymbol{y}_j + \boldsymbol{B}_j^\top \boldsymbol{w}_j + b_j \boldsymbol{1}_{n_j}), \end{aligned} \tag{32}$$

where $\boldsymbol{\alpha}_j = (\alpha_{1j}, \alpha_{2j}, \ldots, \alpha_{n_j,j})^\top$'s consist of the Lagrange multipliers. According to the KKT conditions, we need to solve the linear equations ($j = 1, \ldots, m$)

$$\begin{cases} \dfrac{\partial L}{\partial \boldsymbol{w}_j} = \boldsymbol{0} \Longrightarrow \boldsymbol{w}_j = \boldsymbol{B}_j \boldsymbol{\alpha}_j \\ \dfrac{\partial L}{\partial b_j} = 0 \Longrightarrow \boldsymbol{\alpha}_j^\top \boldsymbol{1}_{n_j} = 0 \\ \dfrac{\partial L}{\partial \boldsymbol{\varepsilon}_j} = \boldsymbol{0} \Longrightarrow \mathrm{diag}(\boldsymbol{\varepsilon}_j) = \lambda^{-1}\mathrm{diag}(\boldsymbol{\eta}_j)^{-2}\mathrm{diag}(\boldsymbol{\alpha}_j) \\ \dfrac{\partial L}{\partial \boldsymbol{\alpha}_j} = \boldsymbol{0} \Longrightarrow \boldsymbol{\varepsilon}_j - \boldsymbol{y}_j + \boldsymbol{B}_j^\top \boldsymbol{w}_j + b_j \boldsymbol{1}_{n_j} = \boldsymbol{0}. \end{cases} \tag{33}$$

Eliminating $\{\boldsymbol{w}_j\}_{j=1}^m$ and $\{\boldsymbol{\varepsilon}_j\}_{j=1}^m$, one can deduce the linear system

$$\begin{bmatrix} 0 & \boldsymbol{A}^\top \\ \boldsymbol{A} & \widetilde{\boldsymbol{B}} \end{bmatrix} \begin{bmatrix} \widetilde{\boldsymbol{b}} \\ \widetilde{\boldsymbol{\alpha}} \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{y} \end{bmatrix}, \tag{34}$$

where $\boldsymbol{y} = (\boldsymbol{y}_1^\top, \ldots, \boldsymbol{y}_m^\top)^\top$, $\widetilde{\boldsymbol{b}} = (b_1, \ldots, b_m)^\top$, $\widetilde{\boldsymbol{\alpha}} = (\boldsymbol{\alpha}_1^\top, \ldots, \boldsymbol{\alpha}_m^\top)^\top$, $\boldsymbol{A} = \mathrm{blockdiag}(\boldsymbol{1}_{n_1}, \ldots, \boldsymbol{1}_{n_m})$, $\widetilde{\boldsymbol{B}} = \mathrm{blockdiag}(\boldsymbol{K}_1 + \lambda^{-1}\mathrm{diag}(\boldsymbol{\eta}_1)^{-2}, \ldots, \boldsymbol{K}_m + \lambda^{-1}\mathrm{diag}(\boldsymbol{\eta}_m)^{-2})$ and $\boldsymbol{K}_j = \boldsymbol{B}_j^\top \boldsymbol{B}_j$ is defined by the kernel function with elements $k_{sl,j} = K(\boldsymbol{x}_{sj}, \boldsymbol{x}_{lj})$.

Let the solution of (34) be $\widetilde{\boldsymbol{b}}^*$ and $\widetilde{\boldsymbol{\alpha}}^*$. Then the regression function is obtained as

$$\hat{f}_j(\boldsymbol{x}) = \varphi(\boldsymbol{x})^\top \boldsymbol{B}_j \boldsymbol{\alpha}_j^* + b_j^* = \sum_{i=1}^{n_j} \alpha_{ij}^* K(\boldsymbol{x}, \boldsymbol{x}_{ij}) + b_j^*. \tag{35}$$

Hence, the residuals $\boldsymbol{\varepsilon}_j$'s can be updated by

$$\varepsilon_{ij}^{(t+1)} = y_{ij} - \hat{f}_j(\boldsymbol{x}_{ij})\big|_{\boldsymbol{\Theta}^{(t)}}, \; j = 1, \ldots, m, i = 1, \ldots, n_j. \tag{36}$$

The specific learning procedure of MoT-MTR is similar to Algorithm 1, but the corresponding updating formulas should be replaced by the ones above.

IV. EXPERIMENTS

In this section, we study the performance of the MoT-STR and MoT-MTR using several synthetic data sets and benchmark data sets. The regularization parameter λ and the kernel parameter σ^2 is obtained by grid search from the set $\{2^i | i = -8, -6, \ldots, 0, \ldots, 6, 8\}$. The mean absolute error (MAE) and the root mean square error (RMSE) are used to evaluate the effectiveness of all models.

A. SYNTHETIC DATA FOR SINGLE TASK

The first synthetic data is generated from the sinc function

$$f(x) = \frac{\sin \pi x}{\pi x}, \tag{37}$$

where 100 training points, together with 100 test points of x are randomly selected from the interval $[-3, 3]$. To investigate the robustness, one tenth of the training data are randomly selected and the response y values are multiplied by 10. Those data can be treated as outliers. In the same time, half of the training data are contaminated by different types of noises. The descriptions of all types of noises are listed in Table 1.

TABLE 1. Descriptions of all types of noises on synthetic data for STR.

Case	Noise Distribution	Parameters	Rate
1	Normal	(0, 1)	0.25
	Student t	(0, 0.25; 4)	0.25
2	Laplace	(0, 1)	0.25
	Student t	(0, 0.25; 4)	0.25
3	Student t	(0, 0.25; 4)	0.25
	Student t	(0, 0.64; 4)	0.25
4	Normal	(0, 1)	1/6
	Laplace	(0, 0.64)	1/6
	Student t	(0, 0.25; 4)	1/6
5	Laplace	(0, 1)	1/6
	Student t	(0, 0.25; 4)	1/6
	Student t	(0, 0.64; 4)	1/6
6	Student t	(0, 0.25; 4)	1/6
	Student t	(0, 0.64; 4)	1/6
	Student t	(0, 1; 4)	1/6

The test data is outlier-free and noise-free. The SVM, LS-SVM [27], WLS-SVM [31], MoG-STR [21] and the proposed MoT-STR are employed for comparisons. The MAE and RMSE on test data are averaged on 5 trails and listed in Table 2. It shows that the performance of MoT-STR is significantly better than other models. It can largely improve the accuracy of estimation in terms of MAE or RMSE. Figure 1 show estimation curves of the sinc function under the WLS-SVM, MoG and the proposed model. We find that the WLS-SVM doesn't perform well in most of the cases especially in the areas near the extreme points and the tails. The MoG performs a bit better, but the MoT improves the results significantly in all the cases.

To evaluate the robustness of our model under distribution misspecification, 20, 40, 60, 80 and 100 percent of randomly selected training data are contaminated respectively in the case 2 noises. We also average the values of MAE and RMSE under LS-SVM, WLS-SVM, MoG and MoT on 5 trails. The results are presented in Figure 2. We can clearly see that as the rate of noises increases, the performance of all models is deteriorated as expected, but MoT performs much better than the others. MoT-STR is therefore the best choice in the presence of outliers or distribution misspecification.

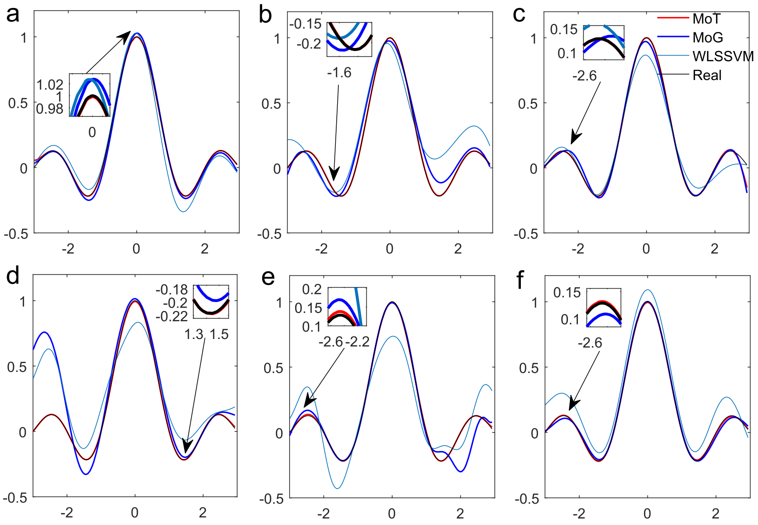


FIGURE 1. The estimated curves of the sinc function under six types of noises. (a) case 1, (b) case 2, (c) case 3, (d) case 4, (e) case 5, (f) case 6.

TABLE 2. Results of STR on synthetic data.

Case	Error	Regression model				
		SVM	LS-SVM	WLS-SVM	MoG	MoT
1	MAE	0.2062	0.4219	0.0570	0.0188	0.0026
	RMSE	0.2611	0.4500	0.0689	0.0227	0.0063
2	MAE	0.3283	0.4776	0.1310	0.0512	0.0017
	RMSE	0.3830	0.5778	0.1551	0.0627	0.0023
3	MAE	0.1010	0.1130	0.0519	0.0159	0.0057
	RMSE	0.1217	0.1286	0.0653	0.0319	0.0212
4	MAE	0.5977	0.4636	0.1612	0.1331	0.0022
	RMSE	0.6327	0.5443	0.2142	0.2458	0.0029
5	MAE	0.1140	0.3149	0.1310	0.0469	0.0023
	RMSE	0.1482	0.3587	0.1535	0.0944	0.0034
6	MAE	0.2133	0.2815	0.1052	0.0115	0.0032
	RMSE	0.2653	0.3142	0.1154	0.0147	0.0064

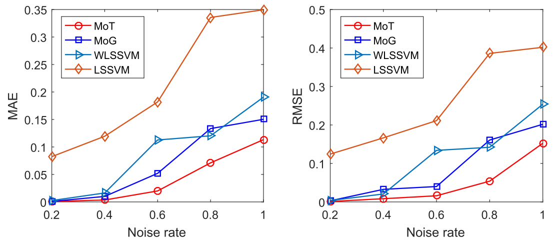


FIGURE 2. The predicted errors of STR on synthetic data with case 2 noises.

TABLE 3. Descriptions of all real data sets for STR.

Data set	Samples	Attributes	Train	Test
Waterdynamic	308	6	200	108
Computer Hardware	209	7	150	9
Forecast	61	12	40	21
Bodyfat	252	14	200	52
Slump	103	7	50	53

TABLE 4. Results of STR on real data sets.

Data set	Error	Regression model				
		SVM	LS-SVM	WLS-SVM	MoG	MoT
Bodyfat	MAE	0.6683	0.436	6.3134e-04	0.0025	1.3600e-04
	RMSE	0.7993	0.470	0.0016	0.0034	2.7932e-04
Bodyfatlap	MAE	0.4703	0.4310	2.6968e-04	8.2381e-04	7.0283e-05
	RMSE	0.6154	0.4709	3.1759e-04	0.0011	8.9573e-05
Bodyfatmix	MAE	0.6228	0.4653	0.0031	0.0192	0.0109
	RMSE	0.6709	0.4684	0.0098	0.0273	0.0140
Waterdynamic	MAE	30.2275	11.4884	3.2967	0.4096	0.3672
	RMSE	36.1050	14.8883	4.7028	0.5801	0.5640
Hardware	MAE	250.6603	99.9503	6.6296	2.4731	2.0827
	RMSE	282.9691	130.3835	15.2411	5.2187	4.7462
Forecast	MAE	303.0867	285.8558	4.6190	2.2582	0.4947
	RMSE	319.3412	309.5571	7.6349	2.8578	0.7624
Slump	MAE	51.1854	102.8247	13.5908	12.8298	10.6901
	RMSE	53.7961	103.1623	17.0921	18.0443	13.7825

B. REAL DATA FOR SINGLE TASK

Next, we use some real data sets for comparison. The data sets include Waterdynamic, Computer Hardware, Forecast, Slump from UCI repository, and Bodyfat from Statlib collection. Descriptions of all data sets are shown in Table 3. For each data set, the input and target variables are normalized into the interval [0, 1]. Synthetic outliers are added into the training set by the same method in Section IV-A. The random noises of Bodyfatlap are generated by Laplace distribution LA(0, 0.15), whereas the noises of Bodyfatmix are generated by the mixtures of LA(0, 0.15) and N(0, 0.15^2) distributions.

Table 4 lists all the results under different models. SVM and LS-SVM are clearly ineffective to cope with contaminated data. WLS-SVM, MoG and MoT improve the results significantly, and overall MoT provides the best results.

C. SYNTHETIC DATA FOR MULTI-TASK

We firstly focus on comparing the performance of MoT and MoG in multi-task learning. We construct data under two scenarios. In the first scenario, the inputs of all samples are randomly selected from the interval $[0, 2\pi]$, the corresponding outputs in two tasks come from the following two functions

$$f_1(x) = \sin x + \cos x, \quad f_2(x) = \sin x - \cos x. \tag{38}$$

200 samples are generated for training and 20 samples for testing. In all tasks, we randomly select one tenth points and shift their outputs by 50 and contaminate all the points by mixture noises $4/5\ t(0, 0.05; 4) + 1/5\ t(0, 0.25; 4)$. In the second scenario, the outputs in three tasks come from the functions

$$f_j(x) = \sin x + \cos(3x)/3 + \tau_j(x), \quad j = 1, 2, 3, \tag{39}$$

where $\tau_j(x)$'s are generated by Gaussian process using covariance kernel

$$\kappa(x_a, x_b) = v \exp\{-w(x_a - x_b)^2/2\}, \tag{40}$$

with $v = 0.04$ and $w = 1$. Each task has 100 samples for training and 100 samples for testing which are randomly generated from the interval $[0, 2\pi]$. In the 100 training data of each task, we randomly select two samples as outliers by times their outputs by 10. We also contaminate 20 samples by mixture noises $1/3\ t(0, 0.16; 4) + 1/3\mathrm{LA}(0, 0.64) + 1/3\mathrm{N}(0, 1)$.

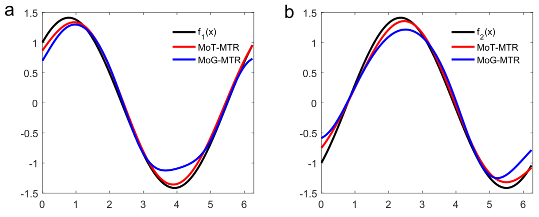


FIGURE 3. The estimated curves of MTR in Scenario 1. (a) $f_1(x)$, (b) $f_2(x)$.

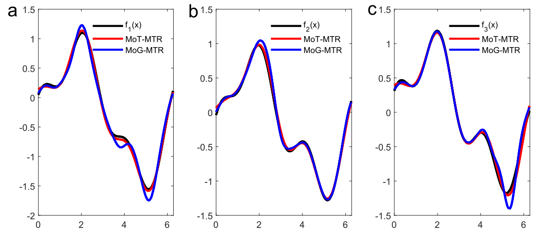


FIGURE 4. The estimated curves of MTR in Scenario 2. (a) $f_1(x)$, (b) $f_2(x)$, (c) $f_3(x)$.

The estimated curves in Scenario 1 and Scenario 2 are presented in Figures 3 and 4 respectively. Both show that the estimated curves under MoT are closer to the true curves than those under MoG. Table 5 show the numerical results on test data. Compared with MoG, MoT can reduce the estimation error by about 50% in both scenarios. It shows the robustness of MoT against outliers and data contamination.

TABLE 5. Results of MTR on synthetic data in Scenario 1 and 2.

Synthetic data	Error	Regression model	
		MoG-MTR	MoT-MTR
Scenario 1	MAE	0.0921	0.0548
	RMSE	0.1235	0.0649
Scenario 2	MAE	0.0436	0.0275
	RMSE	0.0628	0.0343

One of the major advantages of multi-task learning is its ability of sharing information across different tasks via the structural expression of the tasks and the error composition. We now investigate the performance MoT-MTR in two scenarios. In Scenario 3, the inputs of 120 training samples are selected from the interval $[-10, 10]$, where the corresponding outputs are computed by the following two functions

$$\begin{aligned} f_1(x) &= \sin x + \cos(3x)/3, \\ f_2(x) &= \sin x + \cos(3x)/3 + x/3. \end{aligned} \tag{41}$$

To investigate complementary information sharing from other tasks, 100 points of x for task 1 are randomly selected from the interval $[-10, 0]$ while the other 20 points from $[0, 10]$; while the data points for Task 2 are the opposite, i.e. 20 from $[-10, 0]$ and 100 from $[0, 10]$. The x of 100 test data are all randomly selected from $[-10, 10]$. In Scenario 4, the functions in three tasks are the same as those in Scenario 2. The training inputs of the tasks are sparsely located in $[-1, 9]$ except the first one. Specifically, task one have 120 training samples while task 2 and task 3 each have only 20 training samples.

TABLE 6. Results of MTR on synthetic data in Scenario 3 and 4.

Synthetic data	Error	Regression model	
		MoT-STR	MoT-MTR
Scenario 3	MAE	0.0981	0.0527
	RMSE	0.1203	0.0956
Scenario 4	MAE	0.1052	0.0891
	RMSE	0.1348	0.1106

The estimated curves in Scenario 3 and Scenario 4 are shown in Figure 5 and Figure 6 respectively, and the numerical results for test data are listed in Table 6. In Scenario 3, the two tasks can borrow information from each other since the sparse sample locations of the two tasks are complementary. Compared with the MoT-STR (i.e. treat each task separately), the MoT-MTR can reduce the prediction error by about 40%. In Scenario 4, the last two tasks can share information each other and borrow information from the first task. Consequently, the MoT-MTR can reduce the prediction error by about 20%. Therefore, the MoT-MTR has clear advantages over the MoT-STR through communication among tasks.

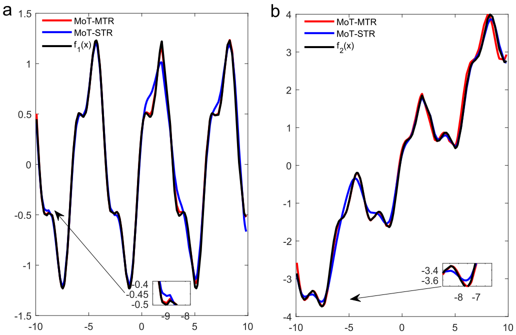


FIGURE 5. The estimated curves of MTR in Scenario 3.

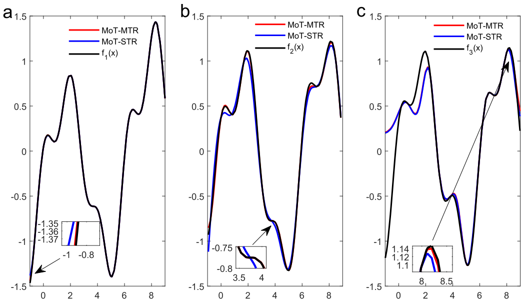


FIGURE 6. The estimated curves of MTR in Scenario 4.

TABLE 7. Descriptions of all real data sets for MTR.

Data set	Sample	Attribute	Target	Train	Test
Polymer	61	10	4	41	20
EDM	154	16	2	110	44
Slump	103	7	3	50	53
Traffic	96	1	8	50	46

D. REAL-WORLD DATA FOR MULTI-TASK

The real data for multi-task include the Polymer, EDM, Slump and Traffic data. These data are taken from ftp://ftp.cis.upenn.edu/pub/ungar/chemdata/, http://mulan.sourceforge.net/datasets-mtr.html, UCI repository, and http://tris.highwaysengland.co.uk respectively. The descriptions of all data are shown in Table 7. For the convenience of selecting penalty parameters λ and kernel parameters σ^2, both the input variables and the target variables are standardised into the interval [0, 1]. We synthesise outliers for randomly selected one fifth data points from the training set by increasing their values of outputs by 10 times. The noises added to the one tenth of training samples from LA(0, 0.15). The experiment results are shown in Table 8.

Table 8 shows clearly that the proposed model MoT-MTR performs much better than MoG-MTR in the presence of outliers and contaminations for multi-tasking data.

E. COMPUTATIONAL COMPLEXITY AND LEARNING CURVE

For single task regression, the computational complexity of SVM regression is $O(n^3)$, where n stands for the sample size. The complexity of both MoG-STR and MoT-STR is $O(n^3+Kn)$, where K is the number of mixed components. The airfoil data from UCI repository is used in this subsection. The total sample size is 1500, in which, 1000 of them are used for training and remaining 500 for testing. Figure 7 shows the running time of SVM, MOG and MoT under different training sample sizes. It shows how the running time in each iteration increase with the sample size. The MoT has no obvious disadvantage in running time compared with other models even if the sample size increases to 1000. For multi-task regression, we suppose that there are m tasks and all tasks have n samples together. The computational complexity of SVM regression is $O((m+n)^3)$. MoG will increase complexity about $O(Kn)$, while MoT will increase another $O(Kn)$. Because the number of K is far less than m and n, the total complexity of MoT-MTR is still $O((m+n)^3)$. When n or m is large, Nyström method [32], Krylov method [33] or others can help to reduce the amount of computation burden.

we now perform a study on convergence speed through learning curve. The WLS-SVM, MoG and MoT are three candidates for comparison. Figure 8(a) shows the predictive RMSEs of the three models under different sizes of

TABLE 8. Results of MTR on real data sets.

Data set	Error	Regression model	
		MoG-MTR	MoT-MTR
Polymer	MAE	3.3648	0.1798
	RMSE	3.6556	0.1120
EDM	MAE	0.0830	0.0658
	RMSE	0.0997	0.0832
Slump	MAE	0.0217	0.0189
	RMSE	0.0281	0.0245
Traffic	MAE	0.0083	0.0074
	RMSE	0.0127	0.0102

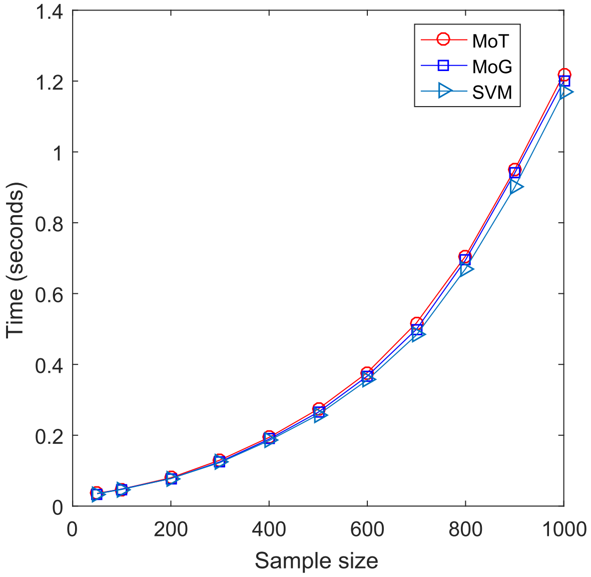


FIGURE 7. Running time of three regression models under different training sample sizes.

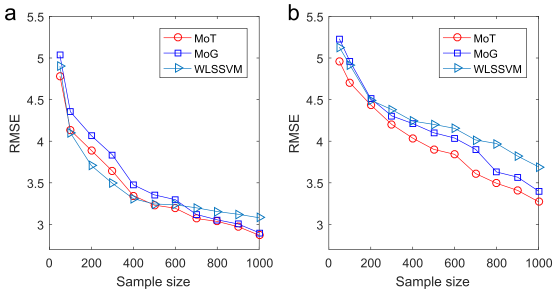


FIGURE 8. Learning curves of three regression models.

training samples. As the size of training samples increases, the RMSEs of all models decrease as expected. The RMSEs of MoT is slightly smaller than those of MoG under all sample sizes. The curves of WLSSVM and two mixed models cross at about 500, meaning that the mixed models have advantages in this case when the training sample sizes become larger. Furthermore, we randomly choose half of the training samples and contaminate them by mixtures noises $1/2\text{LA}(0, 8) + 1/2t(0, 8; 4)$. Figure 8(b) shows the learning curves for the contaminated data. Under all sample sizes, the RMSEs of WLS-SVM is clearly larger than those of the two mixed model, and MoT has the smallest RMSEs among all the candidates. Overall, MoT is very competitive particularly for contaminated data.

V. CONCLUSION AND FUTURE WORK

Finite mixture distributions or models are particularly effective in modeling multi-structures or complex noises data. In this paper, we established a loss function (via likelihood) based on the mixture of t-distributions, and developed estimation procedures using EM algorithms, for both single and multi task learning. The robustness of the regression models are greatly improved due to the heavy-tailed feature of t-distribution and the flexibility of the mixture structure. Although only the regression problem on task learning is discussed, the classification problem could be studied from a similar perspective.

We need to select the number of components in mixture models. In our work, it was handled by model selection criteria under the framework of likelihood. A penalized method [34] was proposed to simultaneously select the number of mixing components and estimate the mixing proportions with other unknown parameters of mixture regression model. Future work on relevant research in the field of task learning needs to be considered.

For statistical learning based on t-distribution, the robustness is related to the degree of freedom ν. Generally speaking, the robustness of the model and the corresponding inferences could be increased gradually as ν decreases. For real data, we need to find the appropriate value of ν to keep the balance between the goodness of fit and robustness. In our model, the degree ν can be updated with other parameters in each iteration step, or be estimated through marginal likelihood. Based on our experience, the estimate of the degree of freedom ν may be unreliable in three ways: $\hat{\nu}$ is very small, $\hat{\nu}$ is large, or $\hat{\nu}$ is unstable. Possible reasons for these three cases are respectively, (1) the data has extreme outliers, (2) the data has a near normal distribution, and (3) the sample size is not sufficiently large. For the cases (1) and (3), we may pre-fixed the values of ν (e.g. $\nu = 4$) [35], and choose it by BIC, cross validation or other criteria; for the case (2), when the estimation of ν is quite large, say larger than 100, the accuracy of ν is no longer important, since the underline distribution is very close to the normal distribution. In order to reduce the computational burden, we can limit the value of ν in a certain range, e.g., [2, 100] in practice. The parameter of interest in task learning is $\boldsymbol{w}$, and ν is a nuisance parameter. The accuracy of the nuisance parameters' estimation has limited influence to the whole learning procedure based on our experience.

For mixture model and other models with latent variables, the EM algorithm is popular on the calculation of the MLE since conditional likelihood or distribution can be simplified via the hierarchical structures of the models. When the conditional distribution of latent variables is complex, we may use Bayesian approach through Markov chain Monte Carlo (MCMC). However, MCMC usually converges slowly and is very time consuming. The hierarchical likelihood [36] and variational inference [37] could be explored on task learning, which respectively, estimates the latent variables directly and approximates the posterior densities through optimization.

REFERENCES

[1] B. S. Everitt and D. J. Hand, *Finite Mixture Distributions*. London, U.K.: Chapman & Hall, 1981.

[2] D. M. Titterington, A. F. M. Smith, and U. E. Markov, *Statistical Analysis of Finite Mixture Distributions*. New York, NY, USA: Wiley, 1985.

[3] G. J. McLachlan and K. E. Basford, *Mixture Models: Inference and Applications*. New York, NY, USA: Marcel Dekker, 1988.

[4] G. Soffritti and G. Galimberti, "Multivariate linear regression with non-normal errors: A solution based on mixture models," *Statist. Comput.*, vol. 21, no. 4, pp. 523–536, Oct. 2011.

[5] N. Dilokthanakul, P. A. M. Mediano, M. Garnelo, M. C. H. Lee, H. Salimbeni, K. Arulkumaran, and M. Shanahan, "Deep unsupervised clustering with Gaussian mixture variational autoencoders," *CoRR*, vol. abs/1611.02648, pp. 1–12, Jan. 2016.

[6] Z. Tang and Z. Miao, "Fast background subtraction and shadow elimination using improved Gaussian mixture model," in *Proc. IEEE Int. Workshop Haptic, Audio Vis. Environments Games*, Oct. 2007, pp. 541–544.

[7] Z. Ji, Y. Xia, Q. Sun, Q. Chen, D. Xia, and D. D. Feng, "Fuzzy local Gaussian mixture model for brain MR image segmentation," *IEEE Trans. Inf. Technol. Biomed.*, vol. 16, no. 3, pp. 339–347, May 2012.

[8] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted Gaussian mixture models," *Digit. Signal Process.*, vol. 10, nos. 1–3, pp. 19–41, Jan. 2000.

[9] S. Dasgupta, "Learning mixtures of Gaussians," in *Proc. 40th Annu. Symp. Found. Comput. Sci.*, 1999, pp. 634–644.

[10] J. Q. Shi and B. Wang, "Curve prediction and clustering with mixtures of Gaussian process functional regression models," *Statist. Comput.*, vol. 18, no. 3, pp. 267–283, Sep. 2008.

[11] D. Peel and G. J. McLachlan, "Robust mixture modelling using the *t* distribution," *Statist. Comput.*, vol. 10, pp. 339–348, Oct. 2000.

[12] S. Shoham, "Robust clustering by deterministic agglomeration EM of mixtures of multivariate t-distributions," *Pattern Recognit.*, vol. 35, no. 5, pp. 1127–1142, May 2002.

[13] W. Yao, Y. Wei, and C. Yu, "Robust mixture regression using the *t*-distribution," *Comput. Statist. Data Anal.*, vol. 71, no. 3, pp. 116–127, May 2014.

[14] Y. Chen, H. Zhang, Y. Zheng, B. Jeon, and Q. M. J. Wu, "An improved anisotropic hierarchical fuzzy c-means method based on multivariate student t-distribution for brain MRI segmentation," *Pattern Recognit.*, vol. 60, pp. 778–792, Dec. 2016.

[15] D. F. Andrews and C. L. Mallows, "Scale mixtures of normal distributions," *J. Roy. Stat. Soc., B (Methodol.)*, vol. 36, no. 1, pp. 99–102, Sep. 1974.

[16] C. Meza, F. Osorio, and R. De la Cruz, "Estimation in nonlinear mixed-effects models using heavy-tailed distributions," *Statist. Comput.*, vol. 22, no. 1, pp. 121–139, Jan. 2012.

[17] C.-Z. Cao, J.-G. Lin, and X.-X. Zhu, "On estimation of a heteroscedastic measurement error model under heavy-tailed distributions," *Comput. Statist. Data Anal.*, vol. 56, no. 2, pp. 438–448, Feb. 2012.

[18] F. Osorio, "Influence diagnostics for robust P-splines using scale mixture of normal distributions," *Ann. Inst. Stat. Math.*, vol. 68, no. 3, pp. 589–619, Jun. 2016.

[19] C. Cao, J. Q. Shi, and Y. Lee, "Robust functional regression model for marginal mean and subject-specific inferences," *Stat. Methods Med. Res.*, vol. 27, no. 11, pp. 3236–3254, Nov. 2018.

[20] T. Evgeniou, C. A. Micchelli, and M. Pontil, "Learning multiple tasks with kernel methods," *J. Mach. Learn. Res.*, vol. 6, pp. 615–637, Apr. 2005.

[21] H. Wang, Y. Wang, and Q. Hu, "Self-adaptive robust nonlinear regression for unknown noise via mixture of Gaussians," *Neurocomputing*, vol. 235, pp. 274–286, Apr. 2017.

[22] B. Bakker and T. Heskes, "Task clustering and gating for Bayesian multitask Learning," *J. Mach. Learn. Res.*, vol. 4, pp. 83–99, May 2003.

[23] S. Ben-David and R. Schuller, "Exploiting task relatedness for multiple task learning," in *Proc. Conf. Learn. Theory*, 2003, pp. 567–580.

[24] Y. Zhang and D. Yeung, "A convex formulation for learning task relationships in multi-task learning," in *Proc. UAI*, Toronto, ON, Canada, 2010, pp. 733–742.

[25] Y. Zhang and Q. Yang, "A survey on multi-task learning," 2017, *arXiv:1707.08114*. [Online]. Available: http://arxiv.org/abs/1707.08114

[26] V. Christianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*. Cambridge, U.K.: Cambridge Univ. Press, 2002.

[27] J. A. K. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Process. Lett.*, vol. 9, no. 3, pp. 293–300, Jun. 1999.

[28] Q. Hu, S. Zhang, Z. Xie, J. Mi, and J. Wan, "Noise model based ν-support vector regression with its application to short-term wind speed forecasting," *Neural Netw.*, vol. 57, pp. 1–11, Sep. 2014.

[29] O. Karal, "Maximum likelihood optimal and robust support vector regression with *Lncosh* loss function," *Neural Netw.*, vol. 94, pp. 1–12, Oct. 2017.

[30] C. Liu and D. B. Rubin, "The ECME algorithm: A simple extension of EM and ECM with faster monotone convergence," *Biometrika*, vol. 81, no. 4, pp. 633–648, 1994.

[31] J. A. K. Suykens, J. De Brabanter, L. Lukas, and J. Vandewalle, "Weighted least squares support vector machines: Robustness and sparse approximation," *Neurocomputing*, vol. 48, nos. 1–4, pp. 85–105, Oct. 2002.

[32] C. K. I. Williams and M. Seeger, "Using the Nyström method to speed up kernel machines," in *Advances in Neural Information Processing Systems 13*, T. K. Leen, T. G. Dietterich, V. Tresp, Eds. Cambridge, MA, USA: MIT Press, 2001, pp. 682–688.

[33] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed. Philadelphia, PA, USA: SIAM, 2003, pp. 157–258.

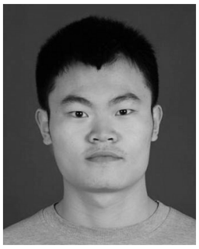
[34] P. Xu, H. Peng, and T. Huang, "Unsupervised learning of mixture regression models for longitudinal data," *Comput. Statist. Data Anal.*, vol. 125, pp. 44–56, Sep. 2018.

[35] K. L. Lange, R. J. A. Little, and J. M. G. Taylor, "Robust statistical modeling using the *t* distribution," *J. Amer. Statist. Assoc.*, vol. 84, no. 408, pp. 881–896, 1989.

[36] Y. Lee, J. A. Nelder, and Y. Pawitan, *Generalized Linear Models With Random-Effects, Unified Analysis via H-Likelihood*, 2nd ed. New York, NY, USA: Chapman & Hall, 2006.

[37] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *J. Amer. Stat. Assoc.*, vol. 112, no. 518, pp. 859–877, Apr. 2017.

CHUNZHENG CAO received the Ph.D. degree in applied mathematics from Southeast University, Nanjing, China, in 2012. He is currently a Professor in statistics with the School of Mathematics and Statistics, Nanjing University of Information Science and Technology, Nanjing. His research interests include statistical learning, functional/longitudinal data analysis, and Bayesian inference.



ZIYUE WANG received the B.S. degree from the School of Mathematics and Statistics, Nanjing University of Information Science and Technology, Nanjing, China, in 2013, where he is currently pursuing the M.S. degree with the School of Mathematics and Statistics. His current research interests include statistical learning and Bayesian inference.

JIAN QING SHI received the B.Sc. degree in computing mathematics, and the M.Sc. and Ph.D. degrees in statistics from the Chinese University of Hong Kong, in 1984, 1990, and 2006, respectively. He is currently a Reader in statistics with Newcastle University and also a Turing Fellow of the Alan Turing Institute. His research interests include nonlinear regression analysis for functional data, applications in big data with complex structure, missing data, and covariance structural equation modeling. He is a member of the Royal Statistical Society.

YUNJIE CHEN (Member, IEEE) received the B.S. and M.S. degrees from the School of Mathematics and Statistics, Nanjing University of Information Science and Technology, Nanjing, China, in 2002 and 2005, respectively, and the Ph.D. degree from the School of Computer Science and Engineering, Nanjing University of Science and Technology, in 2009. He is currently a Professor with the School of Mathematics and Statistics, Nanjing University of Information Science and Technology. His research interests include medical image processing, statistical analysis, and machine learning.