2014 International Joint Conference on Neural Networks (IJCNN)
July 6-11, 2014, Beijing, China

Robust LS-SVR Based on Variational Bayesian and Its Applications

Kefeng Ning[1], Min Liu*[1,2], *Senior Member, IEEE*, Mingyu Dong[1], and Zhansong Wu[3]
[1]Department of Automation, Tsinghua University, Beijing, 100084, China
[2]College of Electrical Engineering, Zhejiang University, Hangzhou, Zhejiang, 310027, China
[3]Department of Thermal Engineering, Tsinghua University, Beijing, 100084, China
lium@tsinghua.edu.cn

***Abstract*—Outliers often exist in the data for modeling in actual industrial processes. If these outliers are used as support vectors, the obtained Support Vector Regression function maybe unreliable. In this paper, we propose a new Robust Least Squares Support Vector Regression based on variational Bayesian (RB-LSSVR). The main idea of RB-LSSVR is to learn the parameters of LSSVR in Bayesian framework, but replace the Gaussian distribution with Student's t-distribution as the probability density function of residuals of the model output and real output, which makes the model more robust to outliers. In order to solve RB-LSSVR, the Student's t-distribution is written as a scale-mixture form and variational approximation is used to iteratively learn the parameters of RB-LSSVR. The hyperparameters of the Gamma distribution that can't be solved explicitly are optimized by using Newton method. And, by using variational Bayesian, the user-specified parameters selection is simplified in RB-LSSVR. The numerical results based on several benchmark regression problems and one actual industrial modeling problem show the proposed RB-LSSVR can handle outliers very well.**

I. Introduction

Considering the large computational complexity of conventional Support Vector Regression (SVR) for large-scale problems, Least Squares Support Vector Regression (LSSVR) is proposed by Suykens et al. [1], [2]. LSSVR is known as LSSVM for classification problems. In LSSVR, the inequality constraints in SVR are replaced by equality constraints. Then, the solution follows from solving a set of linear equations, instead of solving quadratic programming for classical SVR [1]. After that, different variants of LSSVR have been proposed, such as twin LSSVR [3], fuzzy LSSVR [4], LSSVR with confidence/prediction interval [5], recursive reduced LSSVR [6], weighted LSSVR [7], and so on. LSSVR-based methods have been successfully applied to many real-world applications [8], [9], [10], [11], [12].

However, compared with SVR, LSSVR is less robust because of the sensitivity of sum square error (SSE) [13]. While in real-world industrial systems, data used for modeling is always subject to outliers [14]. Outliers may be generated by many reasons, such as measurement error, noise disturbance, and wrong records. Take an actual modeling problem in the steel refining process as an example. The measured value of the steel temperature depends on not only the steel temperature itself but also the direction, angle, and depth of the temperature measuring gun inserted into the liquid steel. Also, the temperature measuring gun may be damaged sometimes and wrong values maybe recorded into the system. So, outliers often exist in the steel temperature modeling problem. When these outliers are used as training data unaware and taken as support vectors in SVR, the learning process may try to fit those unwanted data and the learned model becomes unreliable [14]. The conventional method for handling outliers is to identify them and exclude or downweight them in some way. However, outliers are not always easy to be identified, especially in industrial applications when the dimension of the input data is high and the mechanism is complex.

Considering the above situation, a robust model based on LSSVR is desired for the industrial modeling of data with outliers. In the view of Bayesian framework, the conventional LSSVR is learned under the assumption of the residuals follow Gaussian distribution, which is not robust to outliers. However, The Student's t-distribution is one of the heavy-tailed distributions and gives higher probability for extreme values, which is more robust to outliers. In this paper, we propose a new Robust Least Squares Support Vector Regression based on variational Bayesian, named as RB-LSSVR. In RB-LSSVR, we first learn the parameters by using the conventional LSSVR, then optimize the in a Bayesian framework with the Student's t-distribution as the probability density function of the residuals to replace the general used Gaussian distribution. Unfortunately, the above problem is computation intractable when the Student's t-distribution is used. In order to solve this problem, the Student's t-distribution is written as a scale-mixture of infinitely many Normal distributions and a Gamma distribution. Then, a variational approximation procedure is derived to iteratively learn the parameters. And, the hyperparameters of the Gamma distribution that can't be solved explicitly are optimized by using Newton method.

The rest of this paper is organized as follows: In Section II, LSSVR is briefly reviewed. And, Section III presents the motivation and derivation of the RB-LSSVR in details. Then, the numerical comparisons of RB-LSSVR and LSSVR on several regression problems are discussed in Section IV. Finally, Section V gives the concluding remarks.

*The corresponding author.

978-1-4799-1484-5/14/$31.00 ©2014 IEEE

II. Least Squares Support Vector Regression

This section briefly reviews LSSVR [1], [2]. For N arbitrary distinct samples $(\mathbf{x}_i, t_i) \in \mathbf{R}^n \times R$, the output of the ith sample can be written as,

$$t_i = \mathbf{w}^T \varphi(\mathbf{x}_i) + b + \epsilon_i \quad i = 1, ..., N \tag{1}$$

where $\mathbf{x}_i$ is a $n \times 1$ input vector $(x_{i,1}, x_{i,2}, ..., x_{i,n})$, t_i is one dimensional output, ϵ_i is the residual of the ith output, $\varphi(\cdot)$ is a nonlinear mapping that maps the input feature into a high-dimensional feature space.

Then, LSSVR can be formulated as the following optimization problem, with (1) as its constraints:

$$\min \quad \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\boldsymbol{\epsilon}^T\boldsymbol{\epsilon} \tag{2}$$

where $\boldsymbol{\epsilon} = [\epsilon_1, ..., \epsilon_N]^T$ and $C \in R$ is the parameter for regulation.

By introducing the Lagrangian function, we can get

$$L \quad := \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\boldsymbol{\epsilon}^T\boldsymbol{\epsilon} + \sum_{i=1}^{N} \lambda_i (t_i - \mathbf{w}^T\varphi(\mathbf{x}_i) - b - \epsilon_i) \tag{3}$$

where $\lambda_i, i = 1, ..., N$ are the Lagrange multipliers.

The KKT necessary and sufficient optimality conditions for (3) are given by

$$\begin{cases} \frac{\partial L}{\partial \mathbf{w}} = 0 & \Rightarrow \mathbf{w} = \sum_{i=1}^{N} \lambda_i \varphi(\mathbf{x}_i) \\ \frac{\partial L}{\partial b} = 0 & \Rightarrow \sum_{i=1}^{N} \lambda_i = 0 \\ \frac{\partial L}{\partial \epsilon_i} = 0 & \Rightarrow \lambda_i = C\epsilon_i \\ \frac{\partial L}{\partial \lambda_i} = 0 & \Rightarrow t_i - \mathbf{w}^T\varphi(\mathbf{x}_i) - b - \epsilon_i = 0 \end{cases} \tag{4}$$

Eliminating $\mathbf{w}$ and $\boldsymbol{\epsilon}$, we can get

$$\begin{bmatrix} \mathbf{K} + \frac{\mathbf{I}}{C} & \mathbf{e} \\ \mathbf{e}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ b \end{bmatrix} = \begin{bmatrix} \mathbf{t} \\ 0 \end{bmatrix} \tag{5}$$

where $\mathbf{K}_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i)^T\varphi(\mathbf{x}_j)$, $k(\cdot, \cdot)$ is the kernel function, and $\mathbf{e} = [1, ..., 1]^T$.

Given a new input $\mathbf{x}$, when $\boldsymbol{\lambda}$ and b obtained, we can predict its output by

$$f(\mathbf{x}) = \mathbf{w}^T\varphi(\mathbf{x}) + b = \sum_{i=1}^{N} \lambda_i k(\mathbf{x}, \mathbf{x}_i) + b \tag{6}$$

III. Robust Bayesian LSSVR

A. Problem Description

The robust Bayesian LSSVR can be simply summarized as learn its parameters in LSSVR as the initial parameters and then adjust them in the Bayesian framework with the Student's t-distribution. As $\varphi(\cdot)$ maybe infinitely dimension and unknown, $\mathbf{w}$ can't be solved directly. In order to make LSSVR can be learned in the Bayesian form, we have to make a few transforms. Let $\widetilde{K}(\mathbf{x}) = [k(\mathbf{x}, \mathbf{x}_1), ..., k(\mathbf{x}, \mathbf{x}_N), 1]^T$, $\widetilde{\mathbf{w}} = [\lambda_1, ..., \lambda_N, b]^T$, then (6) can be written as,

$$f(\mathbf{x}) = \widetilde{\mathbf{w}}^T\widetilde{K}(\mathbf{x}) \tag{7}$$

The ith residual of model output and real output can be written as,

$$\epsilon_i = t_i - f(\mathbf{x}_i) = t_i - \widetilde{\mathbf{w}}^T\widetilde{K}(\mathbf{x}_i) \tag{8}$$

Usually, ϵ_is are assumed to follow Normal distribution, which is not robust to outliers. In this paper, the Student's t-distribution (one of the heavy-tailed distributions) is used as the conditional distribution of the ith residual,

$$p(\epsilon_i|\nu, \sigma) = \frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})\sqrt{\pi\nu\sigma}} \times \left(1 + \frac{1}{\nu}\left(\frac{\epsilon_i}{\sigma}\right)^2\right)^{-\frac{\nu+1}{2}}. \tag{9}$$

(9) can be written as a scale-mixture of infinitely many Normal distributions and a Gamma distribution [15]:

$$p(\epsilon_i|c, d) = \int p(\epsilon_i|\beta_i)p(\beta_i|c, d)\mathrm{d}\beta_i \tag{10}$$

where $p(\epsilon_i|\beta_i)$ is a Normal distribution and $p(\beta_i|c, d)$ is a Gamma distribution:

$$p(\epsilon_i|\beta_i) = (\frac{1}{2\pi})^{\frac{N}{2}} \prod_{i=1}^{N} \beta_i^{\frac{1}{2}} e^{-\frac{\beta_i}{2}\epsilon_i^2} \tag{11}$$

$$p(\beta_i|c, d) = \frac{d^c}{\Gamma(c)}\beta_i^{c-1}e^{-\beta_i d}. \tag{12}$$

Gaussian distribution is usually used as the prior distribution over $\widetilde{\mathbf{w}}$,

$$p(\widetilde{\mathbf{w}}|\alpha) = \left(\frac{\alpha}{2\pi}\right)^{\frac{\widetilde{N}}{2}} \prod_{i=1}^{\widetilde{N}} e^{-\frac{\alpha}{2}w_i^2} = \left(\frac{\alpha}{2\pi}\right)^{\frac{\widetilde{N}}{2}} e^{-\frac{\alpha}{2}\widetilde{\mathbf{w}}^T\widetilde{\mathbf{w}}} \tag{13}$$

where $\widetilde{N} = N + 1$.

Based on (8) and (11), the likelihood function of the training data set can be written as

$$p(\mathbf{t}|\widetilde{\mathbf{w}}, \boldsymbol{\beta}) = (\frac{1}{2\pi})^{\frac{N}{2}} \prod_{i=1}^{N} \beta_i^{\frac{1}{2}} e^{-\frac{\beta_i}{2}\left(t_i - \widetilde{\mathbf{w}}^T\widetilde{K}(\mathbf{x}_i)\right)^2}. \tag{14}$$

In order to optimize $\widetilde{\mathbf{w}}$ and $\boldsymbol{\beta}$, the posterior distribution is constructed by Bayes' rule:

$$p(\widetilde{\mathbf{w}}, \boldsymbol{\beta}|\mathbf{t}, \alpha, c, d) = \frac{p(\mathbf{t}|\widetilde{\mathbf{w}}, \boldsymbol{\beta})p(\widetilde{\mathbf{w}}|\alpha)p(\boldsymbol{\beta}|c, d)}{p(\mathbf{t})}. \tag{15}$$

B. Parameters Learning

Unfortunately, (15) is not tractable when the Student's t-distribution is used. We introduce the variational approximation method [16] here to obtain an approximate optimization solution for the optimization of (15). Variational approximation has been applied in Gaussian Process models [17] and Relevance Vector Machine [18].

The goal is to find an approximation for the posterior distribution $p(\widetilde{\mathbf{w}}, \boldsymbol{\beta}|\mathbf{t}, \alpha, c, d)$ and the model evidence $p(\mathbf{t})$. And, the log model evidence $\ln p(\mathbf{t})$ can be reformed as the sum of a lower bound of itself and the KL divergence between two related distributions[16],

$$\begin{aligned} \ln p(\mathbf{t}) &= \int q(\widetilde{\mathbf{w}}, \boldsymbol{\beta}) \ln p(\mathbf{t}) \mathrm{d}\widetilde{\mathbf{w}}\mathrm{d}\boldsymbol{\beta} \\ &= \mathcal{L}(q(\widetilde{\mathbf{w}}, \boldsymbol{\beta}), \alpha, c, d) \\ &\quad + \mathrm{KL}(q(\widetilde{\mathbf{w}}, \boldsymbol{\beta}))||p(\widetilde{\mathbf{w}}, \boldsymbol{\beta}|\mathbf{t}, \alpha, c, d)) \end{aligned} \tag{16}$$

where

$$\mathcal{L}(q(\widetilde{\mathbf{w}},\boldsymbol{\beta}),\alpha,c,d)=\int q(\widetilde{\mathbf{w}},\boldsymbol{\beta})\ln\frac{p(\mathbf{t},\widetilde{\mathbf{w}},\boldsymbol{\beta}|\alpha,c,d)}{q(\widetilde{\mathbf{w}},\boldsymbol{\beta})}\mathrm{d}\widetilde{\mathbf{w}}\mathrm{d}\boldsymbol{\beta} \tag{17}$$

$$\mathrm{KL}(q(\widetilde{\mathbf{w}},\boldsymbol{\beta}))||p(\widetilde{\mathbf{w}},\boldsymbol{\beta}|\mathbf{t},\alpha,c,d))= -\int q(\widetilde{\mathbf{w}},\boldsymbol{\beta})\ln\frac{p(\widetilde{\mathbf{w}},\boldsymbol{\beta}|\mathbf{t},\alpha,c,d)}{q(\widetilde{\mathbf{w}},\boldsymbol{\beta})}\mathrm{d}\widetilde{\mathbf{w}}\mathrm{d}\boldsymbol{\beta}. \tag{18}$$

(18) is the KL divergence (non-negative) between $q(\widetilde{\mathbf{w}},\boldsymbol{\beta})$ and $p(\widetilde{\mathbf{w}},\boldsymbol{\beta}|\mathbf{t},\alpha,c,d)$, and it equals to zero when $q(\widetilde{\mathbf{w}},\boldsymbol{\beta})=p(\widetilde{\mathbf{w}},\boldsymbol{\beta}|\mathbf{t},\alpha,c,d)$. So $\mathcal{L}(q(\widetilde{\mathbf{w}},\boldsymbol{\beta}),\alpha,c,d)\leq\ln p(\mathbf{t})$ and it is a lower bound of $\ln p(\mathbf{t})$. The main idea of variational approximation is to choose a distribution $q(\widetilde{\mathbf{w}},\boldsymbol{\beta})$ to approximate $p(\widetilde{\mathbf{w}},\boldsymbol{\beta}|\mathbf{t},\alpha,c,d)$, and then optimize the lower bound $\mathcal{L}(q(\widetilde{\mathbf{w}},\boldsymbol{\beta}),\alpha,c,d)$ instead of $\ln p(\mathbf{t})$.

The iteration learning procedure of the variational approximation method in this paper can be summarized as follows (in the $k+1$th iteration): the E step choose a distribution $q(\widetilde{\mathbf{w}},\boldsymbol{\beta})$ to approximate $p(\widetilde{\mathbf{w}},\boldsymbol{\beta}|\mathbf{t},\alpha,c,d)$ by minimizing $\mathrm{KL}(q(\widetilde{\mathbf{w}},\boldsymbol{\beta}))||p(\widetilde{\mathbf{w}},\boldsymbol{\beta}|\mathbf{t},\alpha,c,d))$, *i.e.*

$$\begin{aligned}q(\widetilde{\mathbf{w}})^{(k+1)}&\leftarrow\arg\min_{q(\widetilde{\mathbf{w}})}\mathrm{KL}(q(\widetilde{\mathbf{w}},\boldsymbol{\beta}))||p(\widetilde{\mathbf{w}},\boldsymbol{\beta}|\mathbf{t},\alpha,c,d))\\ q(\boldsymbol{\beta})^{(k+1)}&\leftarrow\arg\min_{q(\boldsymbol{\beta})}\mathrm{KL}(q(\widetilde{\mathbf{w}},\boldsymbol{\beta}))||p(\widetilde{\mathbf{w}},\boldsymbol{\beta}|\mathbf{t},\alpha,c,d))\end{aligned} \tag{19}$$

and the M step maximizes $\mathcal{L}(q(\widetilde{\mathbf{w}},\boldsymbol{\beta}),\alpha,c,d)$ with respect to α, c and d

$$\begin{aligned}\alpha^{(k+1)}&\leftarrow\arg\max_{\alpha}\mathcal{L}(q(\widetilde{\mathbf{w}},\boldsymbol{\beta})^{(k+1)},\alpha,c,d)\\ c^{(k+1)}&\leftarrow\arg\max_{c}\mathcal{L}(q(\widetilde{\mathbf{w}},\boldsymbol{\beta})^{(k+1)},\alpha,c,d)\\ d^{(k+1)}&\leftarrow\arg\max_{d}\mathcal{L}(q(\widetilde{\mathbf{w}},\boldsymbol{\beta})^{(k+1)},\alpha,c,d).\end{aligned} \tag{20}$$

$q(\widetilde{\mathbf{w}},\boldsymbol{\beta})$ are usually assumed to be in the expression of factorized distributions[16].

$$q(\widetilde{\mathbf{w}},\boldsymbol{\beta})=q(\widetilde{\mathbf{w}})q(\boldsymbol{\beta}) \tag{21}$$

where $q(\widetilde{\mathbf{w}})$ and $q(\boldsymbol{\beta})$ are assumed as the same distribution types as $p(\widetilde{\mathbf{w}}|\alpha)$ and $p(\boldsymbol{\beta}|c,d)$,

$$q(\widetilde{\mathbf{w}})=\mathcal{N}(\widetilde{\mathbf{w}}|\widetilde{\boldsymbol{\mu}},\widetilde{\boldsymbol{\Sigma}}) \tag{22}$$

$$q(\boldsymbol{\beta})=\prod_{i=1}^{N}\mathrm{Gamma}(\beta_i|\widetilde{c}_i,\widetilde{d}_i). \tag{23}$$

Based on the above assumptions, (19) leads to the following equation [15]

$$\begin{aligned}q(\widetilde{\mathbf{w}})^{(k+1)}&\propto\exp\mathop{\mathbb{E}}_{q(\boldsymbol{\beta})}\ln p(\mathbf{t},\widetilde{\mathbf{w}},\boldsymbol{\beta},|\alpha,c,d)\\ q(\boldsymbol{\beta})^{(k+1)}&\propto\exp\mathop{\mathbb{E}}_{q(\widetilde{\mathbf{w}})}\ln p(\mathbf{t},\widetilde{\mathbf{w}},\boldsymbol{\beta},|\alpha,c,d)\end{aligned} \tag{24}$$

where $\mathbb{E}$ means the expectation.

The optimal solution of $q(\widetilde{\mathbf{w}})$ (i.e. $q^{\star}(\widetilde{\mathbf{w}})$) can be obtained from

$$\begin{aligned}\ln q^{\star}(\widetilde{\mathbf{w}})&=\mathop{\mathbb{E}}_{q(\boldsymbol{\beta})}\ln p(\mathbf{t},\widetilde{\mathbf{w}},\boldsymbol{\beta}|\alpha,c,d)\\ &=\mathop{\mathbb{E}}_{q(\boldsymbol{\beta})}\left(-\frac{\alpha}{2}\widetilde{\mathbf{w}}^T\widetilde{\mathbf{w}}-\frac{1}{2}\sum_{i=1}^{N}\beta_i(t_i-\widetilde{\mathbf{w}}^T\widetilde{K}(\mathbf{x}_i))^2\right)+\text{const}\\ &=-\frac{\alpha}{2}\widetilde{\mathbf{w}}^T\widetilde{\mathbf{w}}-\frac{1}{2}\widetilde{\mathbf{w}}^T\widetilde{\mathbf{K}}^T\mathbf{B}\widetilde{\mathbf{K}}\widetilde{\mathbf{w}}+\widetilde{\mathbf{w}}^T\widetilde{\mathbf{K}}^T\mathbf{B}\mathbf{t}+\text{const}\end{aligned} \tag{25}$$

where

$$\mathbf{B}=\mathrm{diag}\Big\{\mathbb{E}(\beta_1),\mathbb{E}(\beta_2),...,\mathbb{E}(\beta_N)\Big\}. \tag{26}$$

$$\widetilde{\mathbf{K}}=\begin{bmatrix}k(\mathbf{x}_1,\mathbf{x}_1)&\cdots&k(\mathbf{x}_1,\mathbf{x}_N)&1\\ \vdots&\vdots&\vdots&\vdots\\ k(\mathbf{x}_N,\mathbf{x}_1)&\cdots&k(\mathbf{x}_N,\mathbf{x}_N)&1\end{bmatrix} \tag{27}$$

and $\mathbb{E}(\beta_i)$ can be calculated by the expectation of the Gamma distribution

$$\mathbb{E}(\beta_i)=\widetilde{c}_i/\widetilde{d}_i\quad i=1,2,...,N. \tag{28}$$

From (22) and (25), we obtain the parameters ($\widetilde{\boldsymbol{\mu}}$ and $\widetilde{\boldsymbol{\Sigma}}$) of (22) as

$$\widetilde{\boldsymbol{\mu}}=\widetilde{\boldsymbol{\Sigma}}\widetilde{\mathbf{K}}^T\mathbf{B}\boldsymbol{t} \tag{29}$$

$$\widetilde{\boldsymbol{\Sigma}}=(\alpha\mathbf{I}+\widetilde{\mathbf{K}}^T\mathbf{B}\widetilde{\mathbf{K}})^{-1}. \tag{30}$$

Simularly, the optimal solution of $q(\boldsymbol{\beta})$ (i.e. $q^{\star}(\boldsymbol{\beta})$) is given by

$$\begin{aligned}\ln q^{\star}(\boldsymbol{\beta})&=\mathop{\mathbb{E}}_{q(\widetilde{\mathbf{w}})}\ln p(\mathbf{t},\widetilde{\mathbf{w}},\boldsymbol{\beta},|\alpha,c,d)\\ &=\mathop{\mathbb{E}}_{q(\widetilde{\mathbf{w}})}\left(-\frac{\alpha}{2}\widetilde{\mathbf{w}}^T\widetilde{\mathbf{w}}-\frac{1}{2}\sum_{i=1}^{N}\beta_i(t_i-\widetilde{\mathbf{w}}^T\widetilde{K}(\mathbf{x}_i))^2\right.\\ &\left.+\frac{1}{2}\sum_{i=1}^{N}\ln\beta_i+(c-1)\sum_{i=1}^{N}\ln\beta_i-d\sum_{i=1}^{N}\beta_i\right)+\text{const}\\ &=\sum_{i=1}^{N}(c-\frac{1}{2})\ln\beta_i-\sum_{i=1}^{N}\beta_i\left(d+\frac{1}{2}(t_i^2\right.\\ &\left.-2t_i\widetilde{K}(\mathbf{x}_i)^T\mathbb{E}(\widetilde{\mathbf{w}})+\widetilde{K}(\mathbf{x}_i)^T(\mathbb{E}(\widetilde{\mathbf{w}}\widetilde{\mathbf{w}}^T))\widetilde{K}(\mathbf{x}_i))\right).\end{aligned} \tag{31}$$

Compared (23) with (31), we have

$$\widetilde{c}_i=c+\frac{1}{2} \tag{32}$$

$$\begin{aligned}\widetilde{d}_i=d+\frac{1}{2}\Big(&t_i^2-2t_i\widetilde{K}(\mathbf{x}_i)^T\mathbb{E}(\widetilde{\mathbf{w}})\\ &+\widetilde{K}(\mathbf{x}_i)^T(\mathbb{E}(\widetilde{\mathbf{w}}\widetilde{\mathbf{w}}^T))\widetilde{K}(\mathbf{x}_i))\Big)\end{aligned} \tag{33}$$

where $i=1,2,...,N$. $\mathbb{E}(\widetilde{\mathbf{w}})$ and $\mathbb{E}(\widetilde{\mathbf{w}}\widetilde{\mathbf{w}}^T)$ can be calculated by the properties of Normal distribution,

$$\begin{aligned}\mathbb{E}(\widetilde{\mathbf{w}})&=\widetilde{\boldsymbol{\mu}}\\ \mathbb{E}(\widetilde{\mathbf{w}}\widetilde{\mathbf{w}}^T)&=\widetilde{\boldsymbol{\mu}}\widetilde{\boldsymbol{\mu}}^T+\widetilde{\boldsymbol{\Sigma}}.\end{aligned} \tag{34}$$

When (24) (30) (32) and (33) are determined, we can

fix $q(\widetilde{\mathbf{w}})$ and $q(\boldsymbol{\beta})$ and maximize $\mathcal{L}(q(\widetilde{\mathbf{w}},\boldsymbol{\beta}),\alpha,c,d)$ with respect to α, c and d respectively, let

$$\frac{\partial\mathcal{L}(q(\widetilde{\mathbf{w}},\boldsymbol{\beta}),\alpha,c,d)}{\partial\alpha}=\frac{\partial \underset{q(\widetilde{\mathbf{w}})}{\mathbb{E}}\ln p(\widetilde{\mathbf{w}}|\alpha)}{\partial\alpha}=0 \tag{35}$$

we have

$$\frac{\widetilde{N}}{2\alpha}=\frac{\alpha}{2}(\widetilde{\boldsymbol{\mu}}^T\widetilde{\boldsymbol{\mu}}+\mathrm{tr}(\widetilde{\boldsymbol{\Sigma}})) \tag{36}$$

From (36), α can be calculated by

$$\alpha=\frac{\widetilde{N}}{\widetilde{\boldsymbol{\mu}}^T\widetilde{\boldsymbol{\mu}}+\mathrm{tr}(\widetilde{\boldsymbol{\Sigma}})} \tag{37}$$

Similarly, let

$$\frac{\partial\mathcal{L}(q(\widetilde{\mathbf{w}},\boldsymbol{\beta}),\alpha,c,d)}{\partial c}=\frac{\partial \underset{q(\boldsymbol{\beta})}{\mathbb{E}}\ln p(\boldsymbol{\beta}|c,d)}{\partial c}=0 \tag{38}$$

$$\frac{\partial\mathcal{L}(q(\widetilde{\mathbf{w}},\boldsymbol{\beta}),\alpha,c,d)}{\partial d}=\frac{\partial \underset{q(\boldsymbol{\beta})}{\mathbb{E}}\ln p(\boldsymbol{\beta}|c,d)}{\partial d}=0. \tag{39}$$

Two formulations contain c and d can be gotten

$$\ln d=\psi(c)-\sum_{i=1}^{N}(\psi(\widetilde{c}_i)-\ln\widetilde{d}_i)/N \tag{40}$$

$$\frac{c}{d}=\frac{1}{N}\sum_{i=1}^{N}\frac{\widetilde{c}_i}{\widetilde{d}_i} \tag{41}$$

where $\psi()$ is the digamma function.

There is no closed-form solution for (40) and (41). However, we observed $c\propto d$ in (41), then we have

$$d=Nc/\sum_{i=1}^{N}\frac{\widetilde{c}_i}{\widetilde{d}_i}. \tag{42}$$

Eliminate d based on (42), (40) becomes

$$\ln c-\psi(c)=\delta. \tag{43}$$

where

$$\delta=\ln\sum_{i=1}^{N}\frac{\widetilde{c}_i}{\widetilde{d}_i}-\ln N-\sum_{i=1}^{N}(\psi(\widetilde{c}_i)-\ln\widetilde{d}_i)/N. \tag{44}$$

Now we observe that the right-hand side of (43) δ is a fixed value, and the left-hand side of (43) is differentiable and strictly decreases monotonically. So, we can use gradient-based optimization method to solve (43). In this paper, Newton method is used.

C. Algorithm Outline

The RB-LSSVR can be summarized as follows. Suppose the original training data sets are $\{(\mathbf{x}_i,t_i)|\mathbf{x}_i\in\mathbf{R}^n,t_i\in R,i=1,...,N\}$.

Step 1) Select the kernel function $k(\cdot,\cdot)$ (if gaussian kernel are chosen, the user-specified parameter τ should be selected too) and regulator C. Initialize the user-specified parameters α and $\boldsymbol{\beta}$. Set $k=0$.

Step 2) Calculate $\widetilde{\mathbf{w}}$ (i.e. $\boldsymbol{\lambda}$ and b) using (5).

TABLE I
SPECIFICATION OF DATA SETS

Data Sets	#Samples	#Train	#Outliers	#Test	#Inputs
Sinc ($\mathbb{A}$)	201	101	10/20	100	1
Auto-mpg ($\mathbb{B}$)	392	196	20/39	196	7
Bodyfat ($\mathbb{C}$)	252	126	13/25	126	14
LF Steel Temperature ($\mathbb{D}$)	1157	579	58/116	578	20

TABLE II
SPECIFICATION OF OUTLIERS

Case	Outlier Distribution	Parameters	Outlier Rate
1	Normal	(0, 1^2)	0.1
2	Normal	(0, 1^2)	0.2
3	Normal	(0, 2^2)	0.1
4	Normal	(0, 2^2)	0.2

Step 3) Compute the parameters of $q^{(k)}(\boldsymbol{\beta})$ and $q^{(k)}(\widetilde{\mathbf{w}})$ using (24)-(34) repeatedly until convergence.

Step 4) Compute $\alpha^{(k)}$ according to (37). Set $c^{(k)}=10^{-10}$ as the initial value, and iteratively optimize $c^{(k)}$ and $d^{(k)}$ by solving (43) and (42) with Newton method until convergence. And Let $k=k+1$.

Step 5) Run Step 3) and Step 4) iteratively until convergence.

IV. NUMERICAL COMPUTATIONS

The performance of the proposed RB-LSSVR is evaluated and compared with LSSVR by numerical computations with one synthetic data set, several benchmark problems and one actual industrial problem. The Gaussian kernel ($k(\mathbf{x},\mathbf{y})=\exp(-\tau\|\mathbf{x}-\mathbf{y}\|^2)$) is used in both LSSVR and RB-LSSVR. The user-specified parameter τ and C is selected from $[10^{-10},10^{-9},...,10^{10}]$ using cross validation. And, the basic specification of all data sets is shown in Table I. Also, the training data of all data sets are contaminated by synthetic outliers. The synthetic outliers are generated from random distributions, and the details of which can be seen in Table II. All data sets are normalized to $[-1,1]$ except the sinc function data set. The training and testing data of all data sets are reshuffled at each trial of simulation, and the results are averaged on 10 trails.

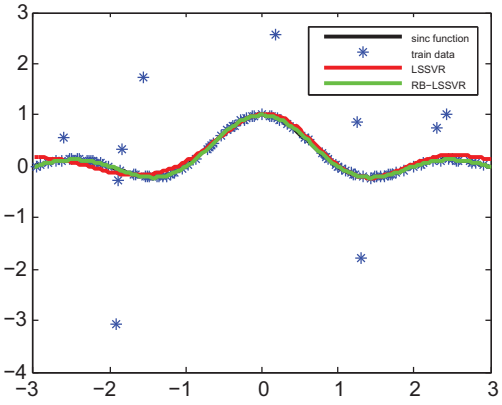
A. Modeling of sinc Function

The sinc function is usually used as a benchmark problem for evaluating regression models, and it is defined as:

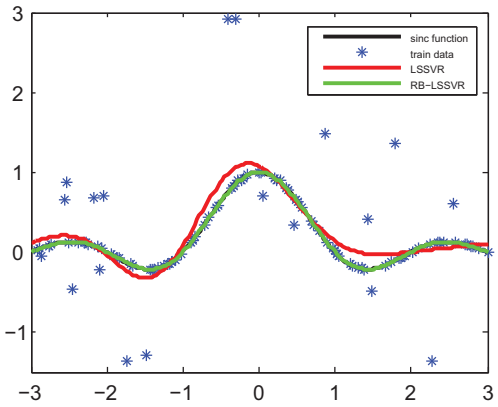
$$y=\mathrm{sinc(x)}=\begin{cases}\frac{\sin(\pi x)}{\pi x} & x\neq 0,\\ 1 & x=0.\end{cases} \tag{45}$$

Two hundred and one patterns of this function are uniformly produced with x in the range of [-3, 3]. Fig.1 plots the approximation results of the sinc function with outliers generated by normal distribution. From Fig.1, we see that LSSVR are affected by outliers in most cases, especially in

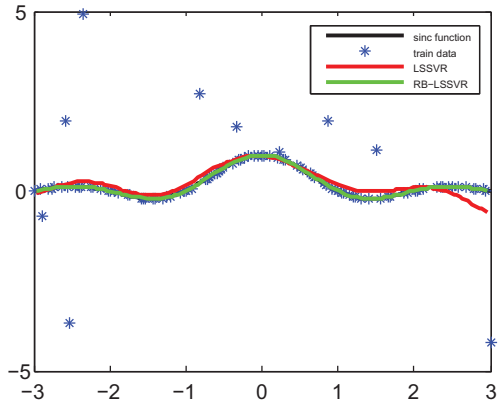
Fig. 1. Modeling of sinc function with outliers generated by normal distribution



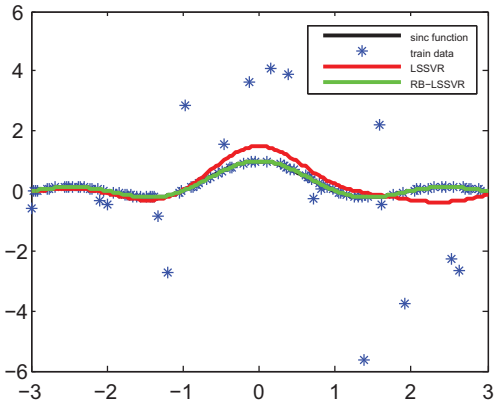
(a) Outlier Case 1



(b) Outlier Case 2



(c) Outlier Case 3



(d) Outlier Case 4

the two ends of the sinc function. While the proposed RB-LSSVR fits the sinc function very well even in the two ends of the sinc function. In Table III, RMSE is used for evaluating the model performance. And the corresponding parameters of best RMSE results and simulation time are also listed in the table. Table III shows that RB-LSSVR outperforms LSSVR in all cases.

B. Several Benchmark Regression Problems

Two data sets (Auto-MPG and Bodyfat) come from StatLib [19] are used as benchmark regression problems. The objective of the Auto-MPG problem is to predict city-cycle fuel consumption in miles per gallon with the information of cylinders, displacement, horsepower, weight, acceleration, model year, origin, and car name. The objective of the Bodyfat problem is to estimate the percentage of body fat from density, age, weight, height, and various skin-fold measurements (e.g. neck circumference).

Table IV shows the best RMSE results that each model obtains in all data sets under all outlier cases. We can see that RB-LSSVR outperforms LSSVR in all data sets. And, Fig.2 shows that the performance of LSSVR is sensitive to the user-specified parameters (C, τ). While in RB-LSSVR, the performance only depends on τ, and different C with same τ will lead to the same performance. So, only τ needs to be chosen in RB-LSSVR, which means the user-specified parameters selection of RB-LSSVR is simplified.

C. Actual Industrial Problem (data set $\mathbb{D}$)

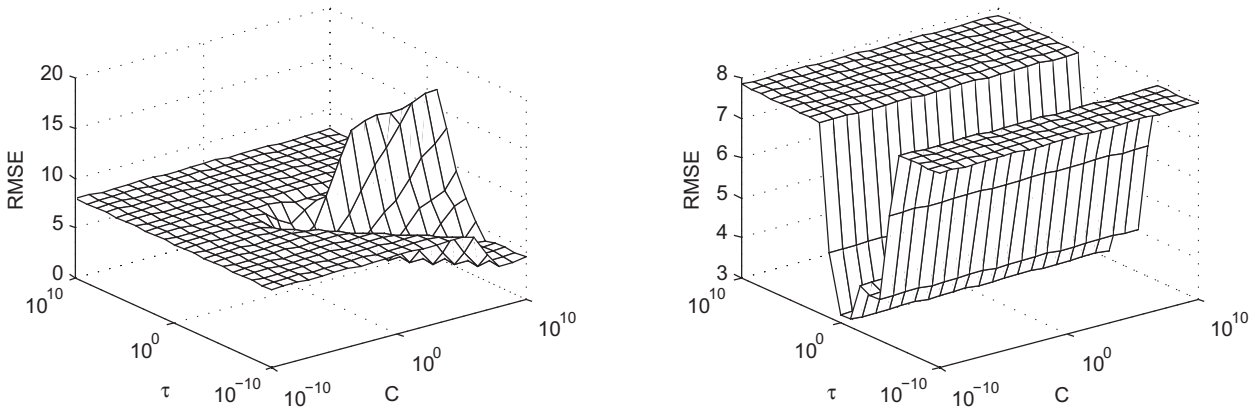
In the Iron & Steel industry, in order to meet the demands of the steel casting process, the liquid steel temperature in the ladle furnace at the end of the refining process should be controlled in an appropriate range. However, the liquid steel temperature is very high, and there are no measurements can measure the liquid steel temperature online yet. So, constructing a model to predict the liquid steel temperature becomes very important. There are many process variables affecting the liquid steel temperature. The main input features used in this paper are listed in Table V.

Table VI shows the simulation time and best RMSE of LSSVR and RB-LSSVR under different user-specified parameters. The RMSE of RB-LSSVR outperforms LSSVR in

TABLE III
SINC FUNCTION APPROXIMATION.

data set	Outlier Case	LSSVR				RB-LSSVR			
		C	τ	Simulation Time (s)	RMSE	C	τ	Simulation Time (s)	RMSE
$\mathbb{A}$	1	1	1	0.55991	0.132379	1	10	1.88876	0.000002
	2	1	1	0.52924	0.144953	1	10	1.66078	0.000003
	3	1	1	0.61429	0.140957	1	10	1.94597	0.000003
	4	1	1	0.63124	0.162382	1	10	1.68511	0.000002

Fig. 2. Performances of LSSVR and RB-LSSVR using Gaussian kernel with user-specified parameters (C, τ): Data Set $\mathbb{B}$



(a) LSSVR (b) RB-LSSVR

TABLE IV
BENCHMARK PROBLEMS.

data set	Outlier Case	LSSVR				RB-LSSVR			
		C	τ	Simulation Time (s)	RMSE	C	τ	Simulation Time (s)	RMSE
$\mathbb{B}$	1	1	0.1	0.31181	3.66166	1	0.1	8.46644	3.20528
	2	1	0.1	0.39998	4.28511	1	0.1	7.53808	2.96537
	3	10	0.01	0.38582	3.54184	1	0.1	7.90403	2.99669
	4	10^9	0	0.49805	3.85705	1	1	12.3427	3.21290
$\mathbb{C}$	1	10^9	0	0.12292	4.27318	1	0.1	1.84939	1.40780
	2	1	0.1	0.13159	4.63450	1	0.1	1.57853	1.78234
	3	10	0.001	0.11674	6.09098	1	0.1	2.09920	1.56433
	4	10^7	0	0.24464	5.76767	1	0.1	1.70724	1.00744

TABLE V
MAIN INPUT FEATURES USED IN DATA SET $\mathbb{D}$.

initial temperature of the steel	span of the refining process
thermal state of the ladle	materials added into the steel
energy from heating system	temperature of the environment
temperature of the cooling water	flow rate of the cooling water
temperature of the flue gas	flow rate of the flue gas
temperature of the ladle wall	flow rate of ar blow

all simulations. Although the simulation time of RB-LSSVR is longer than LSSVR, it doesn't mean the learning of RB-LSSVR is much slower than LSSVR. As in RB-LSSVR, the user-specified parameter C is fixed to one, so RB-LSSVR saves a lot of time on cross validation.

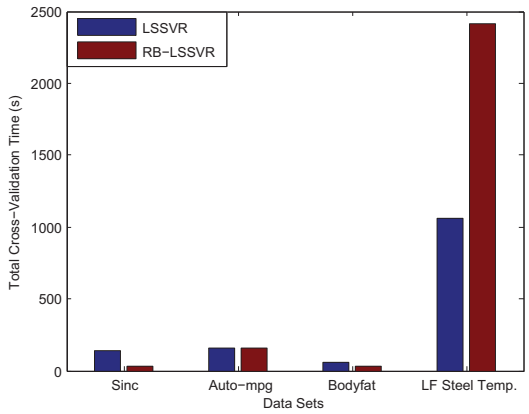
D. Cross-Validation Time Analysis

We have listed the simulation time under given user-specified parameters (C, τ) in Tables III, IV, and VI. From these tables, we know that RB-LSSVR is much slower than LSSVR for a given (C, τ) as it needs an additional step to adjust its parameters. However, the user-specified parameter C can be fixed to one in RB-LSSVR, and τ is the only user-specified parameter need to be determined by cross-validation. So, RB-LSSVR saves a lot of time for cross-validation, which makes the total cross-validation time of RB-LSSVR is comparable to LSSVR. Suppose that the user-specified parameter C and τ are selected from $[10^{-10}, 10^{-9}, ..., 10^{10}]$, the total cross-validation time of LSSVR is $21*21$ times of its single simulation time, while the total cross-validation time of RB-LSSVR is 21 times of its single simulation time. Fig. 3 shows the total cross-validation time spent by LSSVR and RB-LSSVR. From Fig. 3, we can see that the total cross-validation time of RB-LSSVR is less than LSSVR in Sinc, Auto-pmg, and Bodyfat. In data set LF steel temperature, the total cross-validation time of RB-

TABLE VI
ACTUAL INDUSTRIAL PROBLEM.

data set	Outlier Case	LSSVR				RB-LSSVR			
		C	τ	Simulation Time (s)	RMSE	C	τ	Simulation Time (s)	RMSE
$\mathbb{D}$	1	10	0.01	2.37588	8.08505	1	0.1	120.09243	7.80847
	2	1000	0.001	2.80668	9.31798	1	0.01	111.73943	7.97532
	3	10000	0.0001	2.17065	8.04410	1	0.01	98.50746	7.71068
	4	0.1	0.1	2.67462	12.36915	1	0.1	127.96642	8.14831

Fig. 3. Total cross-validation time spent in different data sets



LSSVR is more than LSSVR. In indeed, if we want to use RB-LSSVR in large data sets, the concept of using fixed size support vectors in RB-LSSVR may be a possible solution.

V. CONCLUSION

In this paper, a new Robust Bayesian LSSVR (RB-LSSVR) is proposed for the modeling problem with outliers. The Student's t-distribution is introduced into the Bayesian LSSVR as the probability density function of the model output, which makes the model more robust to outliers. In LSSVR, the user-specified parameters C and τ should be both chosen carefully, if not, the results can be very poor. However, in RB-LSSVR, C is only used in the parameters initialization phase and can be chosen randomly. The only user-specified parameter should be determined by cross validation is τ in RB-LSSVR. And the training time of RB-LSSVR is comparable to LSSVR (considering the cross validation procedure). In addition, the numerical computational results of several regression problems show that the proposed RB-LSSVR outperforms LSSVR and it is a robust model for modeling with outliers.

ACKNOWLEDGMENT

This work was supported by the National Key Basic Research and Development Program of China (2009CB320602), the National Natural Science Foundation of China (60834004, 61025018, 61021063, 61104172) and the National Science and Technology Major Project of China (2011ZX02504-008).

REFERENCES

[1] J. A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Process. Lett.*, vol. 9, no. 3, pp. 293–300, 1999.
[2] J. Suykens, L. Lukas, P. Van Dooren, B. De Moor, J. Vandewalle *et al.*, "Least squares support vector machine classifiers: a large scale algorithm," in *Proc. European Conf. Circuit Theory Design*, vol. 99, 1999, pp. 839–842.
[3] Y.-P. Zhao, J. Zhao, and M. Zhao, "Twin least squares support vector regression," *Neurocomput.*, 2013.
[4] K.-C. Hung and K.-P. Lin, "Long-term business cycle forecasting through a potential intuitionistic fuzzy least-squares support vector regression approach," *Inform. Sci.*, vol. 224, pp. 37–48, MAR 1 2013.
[5] K. De Brabanter, J. De Brabanter, J. A. Suykens, and B. De Moor, "Approximate confidence and prediction intervals for least squares support vector regression," *IEEE Trans. Neural Netw.*, vol. 22, no. 1, pp. 110–120, 2011.
[6] Y. Zhao and J. Sun, "Recursive reduced least squares support vector regression," *Pattern Recogn.*, vol. 42, no. 5, pp. 837–842, 2009.
[7] W. Wen, Z. Hao, and X. Yang, "A heuristic weight-setting strategy and iteratively updating algorithm for weighted least-squares support vector regression," *Neurocomput.*, vol. 71, no. 16, pp. 3096–3103, 2008.
[8] R. Liao, H. Zheng, S. Grzybowski, and L. Yang, "Particle swarm optimization-least squares support vector regression based forecasting model on dissolved gases in oil-filled power transformers," *Electr. Pow. Syst. Res.*, vol. 81, no. 12, pp. 2074–2080, DEC 2011.
[9] W. Cui and X. Yan, "Adaptive weighted least square support vector machine regression integrated with outlier detection and its application in QSAR," *Chemom. Intell. Lab. Syst.*, vol. 98, no. 2, pp. 130–135, OCT 15 2009.
[10] G. Zhiwei and B. Guangchen, "Application of Least Squares Support Vector Machine for Regression to Reliability Analysis," *Chinese J. Aeronaut.*, vol. 22, no. 2, pp. 160–166, APR 2009.
[11] R. Cogdill and P. Dardenne, "Least-squares support vector machines for chemometrics: an introduction and evaluation," *J. Near Infrared Spec.*, vol. 12, no. 2, pp. 93–100, 2004.
[12] K.-P. Lin, P.-F. Pai, Y.-M. Lu, and P.-T. Chang, "Revenue forecasting using a least-squares support vector regression model in a fuzzy environment," *Inform. Sci.*, vol. 220, pp. 196–209, JAN 20 2013.
[13] X. Peng, "Tsvr: an efficient twin support vector machine for regression," *Neural Netw.*, vol. 23, no. 3, pp. 365–372, 2010.
[14] C.-C. Chuang, S.-F. Su, J.-T. Jeng, and C.-C. Hsiao, "Robust support vector regression networks for function approximation with outliers," *IEEE Trans. Neural Netw.*, vol. 13, no. 6, pp. 1322–1330, 2002.
[15] M. Kuss, "Gaussian process models for robust regression, classification, and reinforcement learning," Ph.D. dissertation, TU Darmstadt, 2006.
[16] C. M. Bishop *et al.*, *Pattern recognition and machine learning*. springer New York, 2006, vol. 4, no. 4.
[17] M. J. Beal, "Variational algorithms for approximate bayesian inference," Ph.D. dissertation, University of London, 2003.
[18] M. E. Tipping and N. D. Lawrence, "Variational inference for student-t models: Robust bayesian interpolation and generalised component analysis," *Neurocomput.*, vol. 69, no. 1, pp. 123–141, 2005.
[19] M. Mike, "Statistical datasets," 1989. [Online]. Available: http://lib.stat.cmu.edu/datasets/