Simple Probabilistic Predictions for Support Vector Regression

Chih-Jen Lin[*] and Ruby C. Weng[†]

[*]Department of Computer Science

National Taiwan University, Taipei 106, Taiwan

(cjlin@csie.ntu.edu.tw)

[†]Department of Statistics

National Chengchi University, Taipei 116, Taiwan

Abstract

Support vector regression (SVR) has been popular in the past decade, but it provides only an estimated target value instead of predictive probability intervals. Many work have addressed this issue but sometimes the SVR formula must be modified. This paper presents a rather simple and direct approach to construct such intervals. We assume that the conditional distribution of the target value depends on its input only through the predicted value, and propose to model this distribution by simple functions. Experiments show that the proposed approach gives predictive intervals with competitive coverages with Bayesian SVR methods.

I. Introduction

In the past decade support vector regression (SVR) [15], [12] has been popular for regression problems. SVR provides only an estimated target value; however, the statement that the future value falls in an interval with a specified probability is more informative. This paper aims to construct predictive intervals for the future values.

For conventional linear regression, the prediction interval has been well developed; for example, see [16] for Gaussian noise case and [3], [14] for non-Gaussian case. SVR differs from conventional regression in that it maps input data into a high dimensional reproducing kernel Hilbert space and uses an ϵ-insensitive loss function. As a result, SVR has a sparse representation of solutions, and hence is relatively fast in training/testing. However, due to these differences, the existing methods for constructing prediction intervals can not be applied. Recently Bayesian interpretations of SVR have been developed [6], [4], [2] along the ways of Bayesian techniques for Neural Networks [8] and for SVM classification [13], [11]. Using a Bayesian framework, one can determine parameters in SVR by maximizing an evidence function, and at the same time derive an error bar for prediction.

Some of these Bayesian approaches perform well, but in several situations they cannot be applied. For example, they may modify the SVR formulation, so it is more difficult to use existing SVR software. In addition, some may prefer using other methods (e.g., cross validation) for parameter selection. As the best parameters are not from minimizing the Bayesian evidence function, the Bayesian error bar is not applicable. In this article, we propose a rather simple approach to construct predictive intervals under given parameters. The key ideas are assuming that the conditional distribution of the target value depends on its input only through the predicted value, and modeling this distribution by some simple functions. To begin, we employ cross validation (CV) to obtain a set of out-of-sample regression residuals from the training data. These residuals are supposed to provide information regarding the distribution of prediction errors. Then, as the prediction errors are usually symmetric and concentrated around zero, we fit the residuals with zero-mean Gaussian and Laplace families. The most powerful scale-invariant test is conducted to select between Gaussian and Laplace families. After selecting the family, the final model is determined by using the maximum likelihood estimate for the scale parameter.

The assumption that the distribution of the target value depends on its input only through the predicted value is somewhat restricted. However, it often works well in practice or can provide a crude estimation for initial analysis. For data whose distribution strongly depends on variables, we can cluster data into different groups and apply the proposed technique on each group.

Though an error bar for prediction is a natural by-product under the Bayesian framework, the performance of such an error bar estimation has not been fully investigated in the literature. In this paper, we evaluate the Bayesian approach and our proposed by measuring the difference between the counted and the expected numbers of future data points lying in the interval with pre-specified probabilities. This paper is organized as follows. Section II introduces the methods and justifies their validity. Section III briefly reviews SVR and its Bayesian interpretation. Experiments and analysis on real-world sets are in Sections IV and V, respectively. Section VI gives concluding remarks.

II. The Proposed Approach

In regression problems, we are given a set of training data $\mathcal{D} = \{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in R^n, y_i \in R, i = 1, ..., l\}$. We suppose that the data are collected from the model:

$$y_i = f(\mathbf{x}_i) + \delta_i, \tag{1}$$

where $f(\mathbf{x})$ is the underlying function and δ_i are independent and identically distributed random noises.

Given a test data $\mathbf{x}$, the distribution of y given $\mathbf{x}$ and $\mathcal{D}$, $P(y \mid \mathbf{x}, \mathcal{D})$, allows one to draw probabilistic inferences about y; for example, one can construct a predictive interval $\mathcal{I} = \mathcal{I}(\mathbf{x})$ such that $y \in \mathcal{I}$ with a pre-specified probability. Denoting $\hat{f}$ as the estimated function based on $\mathcal{D}$ (using SVR or other methods by training on $\mathcal{D}$), then $\zeta = \zeta(\mathbf{x}) \equiv y - \hat{f}(\mathbf{x})$ is the out-of-sample residual (or prediction error), and $y \in \mathcal{I}$ is equivalent to $\zeta \in \mathcal{I} - \hat{f}(\mathbf{x})$. We propose to model the distribution of ζ based on a set of out-of-sample residuals $\{\zeta_i\}_{i=1}^{l}$ using training data $\mathcal{D}$. The ζ_i's are generated by first conducting a k-fold cross validation to get $\hat{f}_j$, $j = 1, \ldots, k$, and then setting $\zeta_i \equiv y_i - \hat{f}_j(\mathbf{x}_i)$ for $(\mathbf{x}_i, y_i)$ in the jth fold. It is conceptually clear that the distribution of ζ_i's may resemble that of the prediction error ζ.

To further illustrate this approach, in Figure 1 we investigate ζ_i's from a real data set (cpusmall). Basically, a discretized distribution like histogram can be used to model the data; however, it may be more complex because all

ζ_i's must be retained. On the contrary, distributions like Gaussian and Laplace, commonly used as noise models, require only location and scale parameters. In Figure 1 we plot the fitted curves using these two families and the histogram of ζ_i's. The figure shows that the distribution of ζ_i's seems symmetric about zero and that both Gaussian and Laplace reasonably capture the shape of ζ_i's. Thus, we propose to model ζ_i by zero-mean Gaussian and Laplace, or equivalently, model the conditional distribution of y given $\hat{f}(\mathbf{x})$ by Gaussian and Laplace with mean $\hat{f}(\mathbf{x})$.

To obtain the fitted curves using Laplace and Gaussian distributions, we first express the density functions of zero-mean Laplace and Gaussian with scale parameter σ,

$$\text{Laplace: } p(z) = \frac{1}{2\sigma} e^{-\frac{|z|}{\sigma}}; \tag{2}$$

and

$$\text{Gaussian: } p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-z^2}{2\sigma^2}}. \tag{3}$$

Next, assuming that ζ_i are independent, we can estimate the scale parameter by maximizing the likelihood. For Laplace, the maximum likelihood estimate is

$$\sigma = \frac{\sum_{i=1}^{l} |\zeta_i|}{l}, \tag{4}$$

and for Gaussian,

$$\sigma^2 = \frac{\sum_{i=1}^{l} \zeta_i^2}{l}. \tag{5}$$

Then we obtain the fitted curves by plugging these estimates into (2) and (3), respectively. In the rests of the paper we refer to the two methods as “Lap” and “Gau.” As we conduct CV to obtain ζ_i, (4) is essentially the mean absolute error (MAE) of CV, and (5) the mean squared error (MSE).

In theory, the distribution of ζ may depend on the input $\mathbf{x}$, and accordingly the length of the predictive interval for ζ with a pre-specified coverage probability may vary from case to case, reflecting the fact that the prediction variances vary with different input values. Though our interval for ζ is free of $\mathbf{x}$, and hence does not reflect this property, it can be justified if we consider the probability to be taken over all possible input values. It also worths noting that our modeling shares some similarities with that in [10]. In the context of classification, [10] proposes to model the probability output, $P(y = 1 \mid \hat{f}(\mathbf{x}))$, by a sigmoid function of $\hat{f}$. Both [10] and our approach assume that the conditional distribution of y given $\mathbf{x}$ depends on $\mathbf{x}$ only through $\hat{f}(\mathbf{x})$. Both propose to model this conditional distribution by simple parametric functions, and then estimate the corresponding parameters by the maximum likelihood principle.

Regarding the selection of either Gaussian or Laplace, Figure 1 shows that Laplace seemingly outperforms Gaussian for problem cpusmall. Though a graph like Figure 1 does provide information as to which family better captures ζ_i's, such a visual detection is not efficient and can be subjective. In fact, one can select between Laplace and Gaussian without even fitting the two models. The following theorem [7, chapter 6] gives the most powerful test among all tests which are invariant under scale transformation.

Theorem 1: Suppose that $Z_1, \ldots, Z_l$ are a random sample from a distribution with density

$$\frac{1}{\sigma^l} p(\frac{z_1}{\sigma}) \cdots p(\frac{z_l}{\sigma}),$$

where $p(z)$ is either zero for $z < 0$ or symmetric about zero. The most powerful scale-invariant test for testing $H_0 : p = p_0$ against $H_1 : p = p_1$ rejects H_0 when

$$\frac{\int_0^\infty \tau^{l-1} p_1(\tau z_1) \cdots p_1(\tau z_l) d\tau}{\int_0^\infty \tau^{l-1} p_0(\tau z_1) \cdots p_0(\tau z_l) d\tau} > c.$$

Here "most powerful" means that when H_1 is true, the test has the highest probability of rejecting H_0. The Gaussian versus Laplace [5] is a special case of the theorem.

Corollary 2: (Gaussian vs. Laplace) For $p_0(z) = e^{-z^2/2}/\sqrt{2\pi}$ and $p_1(z) = e^{-|z|}/2$, the test of Lemma 1 reduces to rejecting H_0 when $\sqrt{\sum Z_i^2}/\sum |Z_i| > c$.

At significant level α, the constant c satisfies

$$P_0\left(\frac{\sqrt{\sum Z_i^2}}{\sum |Z_i|} > c\right) = \alpha, \tag{6}$$

where P_0 is the probability under H_0; that is, c is determined so that the probability of rejecting H_0 is α when H_0 is actually true. Typically α is chosen to be 0.05.

Now we briefly summarize the proposed procedure:

1) Generate predicted errors $\zeta_1, \ldots, \zeta_l$ by cross-validation using training data.
2) Use Corollary 2 to test Gaussian against Laplace. Once the decision is made, we determine the scale parameter σ using the maximum likelihood estimate ((4) or (5)).

One should notice that the above procedure may fit for other regression techniques as well, though this paper mainly focuses on applying it to SVR.

III. SVR and its Bayesian Interpretation: a Review

The classical SVR considers the ϵ-insensitive loss function

$$\ell_\epsilon(\delta) = \begin{cases} -\delta - \epsilon & \text{if } \delta < -\epsilon, \\ 0 & \text{if } \delta \in [-\epsilon, \epsilon], \\ \delta - \epsilon & \text{if } \delta > \epsilon, \end{cases} \tag{7}$$

and solves

$$\begin{aligned} \min_{\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\xi}^*} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) \\ \text{subject to} \quad & y_i - f(\mathbf{x}_i) \le \epsilon + \xi_i, \\ & f(\mathbf{x}_i) - y_i \le \epsilon + \xi_i^*, \\ & \xi_i, \xi_i^* \ge 0, i = 1, \ldots, l. \end{aligned} \tag{8}$$

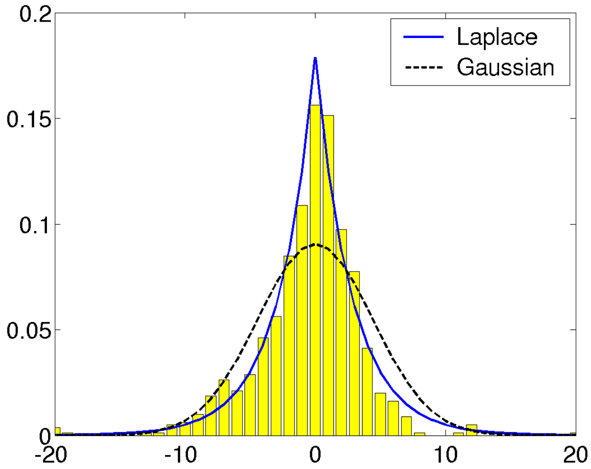

Fig. 1. Histogram of ζ_i's from the problem cpusmall (using parameters (C, γ, ϵ) listed in the last row of Table II(a)). The x-axis is ζ_i using five-fold CV and the y-axis is the normalized number of data in each bin of width 1. The Laplace distribution (4) uses the parameter $\sigma = 2.7948$, which is the cross-validation mean absolute error. The Gaussian distribution (5) uses the parameter $\sigma^2 = 19.4106$, which is the cross-validation mean squared error. Note that there are four of the $|\zeta_i|$'s exceeding 20, with the maximum 51.5, but the x-axis is cut at ±20 for visual concern.

Here $f(\mathbf{x}_i) = \mathbf{w}^T\phi(\mathbf{x}_i) + b$ and data are mapped to a higher dimensional space by the function ϕ. Similar to support vector classification, as $\mathbf{w}$ may be a huge vector variable, we solve the dual problem:

$$\begin{aligned} \min_{\boldsymbol{\alpha},\boldsymbol{\alpha}^*} \quad & \frac{1}{2}(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*)^T K(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*) + \epsilon\sum_{i=1}^{l}(\alpha_i+\alpha_i^*) \\ & +\sum_{i=1}^{l} y_i(\alpha_i-\alpha_i^*) \\ \text{subject to} \quad & \sum_{i=1}^{l}(\alpha_i-\alpha_i^*) = 0, \\ & 0 \le \alpha_i, \alpha_i^* \le C, i = 1,\ldots,l, \end{aligned} \tag{9}$$

where K is the kernel matrix with $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$. For example, the RBF kernel takes the form

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2). \tag{10}$$

In the literature of Bayesian SVR, $\mathbf{f} = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_l)]^T$ is regarded as a random vector whose prior is assumed to be a zero mean Gaussian process with covariance matrix Σ, and the likelihood of the data given $\mathbf{f}$ is assumed to be

$$p(\mathcal{D}|\mathbf{f}) = \Pi_{i=1}^{l} p(\delta_i) \propto \exp(-C \cdot \sum_{i=1}^{l} \ell(\delta_i)), \tag{11}$$

where $\delta_i = y_i - f(\mathbf{x}_i)$, C is a positive parameter, and $\ell(\cdot)$ is the loss function. The parameters in the prior and the likelihood are called $hyperparameters$, denoted as θ, which can be optimized by maximizing the evidence function

$$p(\mathcal{D}|\theta) = \int p(\mathcal{D}|\mathbf{f},\theta)p(\mathbf{f}|\theta)d\mathbf{f}.$$

If we take $\Sigma_{ij} = K_{i,j}$ as in (10) and ℓ as the ϵ-insensitive loss function, then the hyperparameter is $\theta = (\gamma, C, \epsilon)$, where γ comes from the prior of $\mathbf{f}$ and (C, ϵ) from the likelihood of the data given $\mathbf{f}$.

[4] gives a Bayesian interpretation to the classical SVR formulation but without the presence of the constant term b in the underlying function f. Then they derive an approximation to the logarithm of the evidence function:

$$\begin{aligned} \text{ln} p(\mathcal{D}|\theta) \quad &\approx \quad -(\text{optimal objective value of } (9)) \\ & -\frac{1}{2}\text{ln det}(2\pi K_{F,F}) + l\text{ln}\frac{C}{2(\epsilon C+1)} \\ & \sum_{i\in F}\text{ln}\frac{C}{|\alpha_i+\alpha_i^*|(C-|\alpha_i+\alpha_i^*|)}, \end{aligned} \tag{12}$$

where $\boldsymbol{\alpha}$, $\boldsymbol{\alpha}^*$ is the optimal solution of the dual SVR (9) under a given θ, F is the set of their free components:

$$F \equiv \{i \mid 0 < \alpha_i < C \text{ or } 0 < \alpha_i^* < C\}, \text{ and} \tag{13}$$

$K_{F,F}$ is the sub-matrix of the kernel matrix corresponding to F.

Suppose that a test case $\mathbf{x}$ is given for which the target value y corrupted with noise δ is unknown. Applying the ϵ-insensitive loss to (11), one has the density of δ,

$$p(\delta) = \frac{C}{2(\epsilon C+1)}\text{exp}(-C\ell_\epsilon(\delta)), \tag{14}$$

from which we see that δ has mean zero and variance

$$\sigma_\delta^2 = \frac{2}{C^2} + \frac{\epsilon^2(\epsilon C+3)}{3(\epsilon C+1)}. \tag{15}$$

[4] shows that the conditional probability distribution of $f(\mathbf{x})$ given $\mathcal{D}$ is

$$p(f(\mathbf{x})|\mathcal{D}) = \frac{1}{\sqrt{2\pi}\sigma_t}\text{exp}\biggl(-\frac{(f(\mathbf{x})-\hat{f}(\mathbf{x}))^2}{2\sigma_t^2}\biggr), \tag{16}$$

where

$$\sigma_t^2 = K(\mathbf{x},\mathbf{x}) - K_{F,\mathbf{x}}^T K_{F,F}^{-1} K_{F,\mathbf{x}}$$

with $K_{F,\mathbf{x}}$ being the vector containing all $K(\mathbf{x}_i, \mathbf{x}), i \in F$, and

$$\hat{f}(\mathbf{x}) = (\boldsymbol{\alpha}_F - \boldsymbol{\alpha}_F^*)^T \cdot K_{F,\mathbf{x}}$$

is the decision function. Consequently, the prediction variance is

$$\text{var}(y - \hat{f}(\mathbf{x})) = \sigma_\delta^2 + \sigma_t^2,$$

which is the square of the so called “error bar for prediction.”

The main advantages of Bayesian approaches are

1) parameter and feature selection can be done simultaneously by maximizing the evidence function, and
2) the error bar for prediction can be formulated.

TABLE I

DATA SET STATISTICS: FOR space_ga, abalone, add10, AND cpusmall, RANDOM SUBSETS OF 1,000 INSTANCES ARE USED.

Problem	#data	#features
pyrim	74	27
triazines	186	60
bodyfat	252	14
mpg	392	7
housing	506	13
add10	1000	10
cpusmall	1000	12
space_ga	1000	6
abalone	1000	8

The performance depends on the quality of the evidence function. To evaluate the performance of this error bar estimation, the distribution of $\zeta = y - \hat{f}(\mathbf{x})$ is required. By decomposing ζ into two independent components,

$$y - \hat{f}(\mathbf{x}) = (y - f(\mathbf{x})) + (f(\mathbf{x}) - \hat{f}(\mathbf{x})), \tag{17}$$

we can obtain the distribution of ζ by convolution of the two densities (14) and (16).

[2] thinks that the lack of smoothness of the ϵ-insensitive loss function may cause inaccuracy in the approximation of the evaluation function, and hence the inference about θ. Thus, they propose a soft insensitive loss function by solving a modified SVR:

$$\begin{aligned}
\min_{\mathbf{w},\boldsymbol{\xi},\boldsymbol{\xi}^*} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{l}(\psi(\xi_i) + \psi(\xi_i^*)) \\
\text{subject to} \quad & y_i - \mathbf{w}^T\phi(\mathbf{x}_i) \leq (1-\beta)\epsilon + \xi_i, \\
& \mathbf{w}^T\phi(\mathbf{x}_i) - y_i \leq (1-\beta)\epsilon + \xi_i^*, \\
& \xi_i, \xi_i^* \geq 0, i = 1, \ldots, l,
\end{aligned} \tag{18}$$

where

$$\psi(\pi) = \begin{cases} \frac{\pi^2}{4\beta\epsilon} & \text{if } \pi \in [0, 2\beta\epsilon), \\ \pi - \beta\epsilon & \text{if } \pi \in [2\beta\epsilon, \infty). \end{cases}$$

They derive an approximation to the logarithm of the evidence function:

$$\begin{aligned}
\ln p(\mathcal{D}|\theta) \quad \approx \quad & -(\text{optimal objective value of } (18)) \\
& -\frac{1}{2}\ln\,\det\big(I + \frac{C}{2\beta\epsilon}K_{F,F}\big) \\
& -l\ln Z_s,
\end{aligned} \tag{19}$$

TABLE II

DETAILED INFORMATION REGARDING FIVE TRAINING/TESTING SPLITS OF cpusmall. THE EXPECTED NUMBER OF COVERAGE IS (# TEST SET)×(80%)=160. THE METHOD LAP* IS PROPOSED AND DESCRIBED IN SECTION IV.

C,γ,ϵ	Gau	Lap	Lap*	BSVR1
64.0,0.25,0.500	178	174	170	23
64.0,0.25,0.500	176	170	164	24
64.0,0.25,0.004	178	166	161	1
64.0,0.25,0.500	167	159	158	17
64.0,0.25,0.250	181	169	166	12

(a) Best parameters based on CV and the numbers of test instances covered.

C,γ,ϵ	Gau	Lap	Lap*	BSVR1
0.5,0.06,0.500	193	187	169	81
0.5,0.06,0.500	195	183	168	89
0.5,0.06,0.500	193	181	167	84
0.5,0.06,0.500	188	178	161	90
0.5,0.06,0.500	191	180	160	77

(b) Best parameters based on maximizing (12) and the numbers of test instances covered.

$C,\kappa,\kappa_0,\kappa_b,\epsilon$	Gau	Lap	Lap*	BSVR2
0.50,0.68,335.9,102.2,0.057	175	167	162	164
0.43,0.66,338.2,102.0,0.055	181	170	168	165
0.49,0.70,329.7,102.2,0.057	178	164	162	177
0.45,0.68,274.0,103.0,0.056	174	168	167	162
0.54,0.67,314.3,101.6,0.054	182	165	163	171

(c) Best parameters based on maximizing (19) and the numbers of test instances covered.

where I is the identity matrix, F has the same form as (13) but with $\boldsymbol{\alpha},\boldsymbol{\alpha}^*$ replaced by the optimal solution of the dual of (18), and

$$Z_s = 2(1-\beta)\epsilon + 2\sqrt{\frac{\pi\beta\epsilon}{C}}\mathrm{erf}(\sqrt{C\beta\epsilon}) + \frac{2}{C}e^{-C\beta\epsilon}$$

with

$$\mathrm{erf}(z) = \frac{2}{\sqrt{\pi}}\int_0^z e^{-t^2}dt.$$

Their conditional distribution of $f(\mathbf{x})$ given data has the same form as (16), but with

$$\sigma_t^2 = K(\mathbf{x},\mathbf{x}) - K_{F,\mathbf{x}}^T\big(\frac{2\beta\epsilon}{C}I + K_{F,F}\big)^{-1}K_{F,\mathbf{x}}. \tag{20}$$

TABLE III

DETAILED INFORMATION REGARDING FIVE TRAINING/TESTING SPLITS OF housing. THE EXPECTED NUMBER OF COVERAGE IS (# TEST SET)×(80%)=81.

C, γ, ϵ	Gau	Lap	Lap*	BSVR1
32.0,0.12,0.031	87	81	78	50
8.0,0.25,0.062	90	86	84	97
4.0,0.25,0.062	93	91	87	100
64.0,0.25,0.125	84	80	79	78
8.0,0.25,0.062	93	88	86	100

(a) Best parameters based on CV and the numbers of test instances covered.

C, γ, ϵ	Gau	Lap	Lap*	BSVR1
32.0,0.50,0.004	87	82	80	70
32.0,0.50,0.008	88	86	86	74
16.0,0.50,0.016	92	90	90	86
32.0,0.50,0.004	91	89	89	71
16.0,0.50,0.004	86	82	82	90

(b) Best parameters based on maximizing (12) and the numbers of test instances covered.

$C, \kappa, \kappa_0, \kappa_b, \epsilon$	Gau	Lap	Lap*	BSVR2
13.91,0.72,0.2,73.2,0.050	91	83	78	88
11.03,0.54,0.2,81.7,0.045	92	89	85	91
14.08,0.48,0.2,60.1,0.077	92	90	88	92
10.76,0.41,0.2,81.4,0.041	93	86	79	88
13.73,0.43,0.2,66.5,0.058	95	86	83	85

(c) Best parameters based on maximizing (19) and the numbers of test instances covered.

In contrast to (7), the loss function becomes

$$l_{\epsilon,\beta}(\delta) = \begin{cases} -\delta - \epsilon & \text{if } \delta \in (-\infty, -(1+\beta)\epsilon) \\ \frac{(\delta+(1-\beta)\epsilon)^2}{4\beta\epsilon} & \text{if } \delta \in [-(1+\beta)\epsilon, -(1-\beta)\epsilon] \\ 0 & \text{if } \delta \in (-(1-\beta)\epsilon, (1-\beta)\epsilon) \\ \frac{(\delta-(1-\beta)\epsilon)^2}{4\beta\epsilon} & \text{if } \delta \in [(1-\beta)\epsilon, (1+\beta)\epsilon] \\ -\delta - \epsilon & \text{if } \delta \in ((1+\beta)\epsilon, \infty). \end{cases} \tag{21}$$

TABLE IV

AVERAGE ABSOLUTE DIFFERENCE ON NUMBER OF COVERAGES: USING CV FOR PARAMETER SELECTION.

Problem	#80%	Gau	Lap	Lap*	Hist	BSVR1
pyrim	11.8	1.4	1.2	1.8	2.0	2.8
triazines	29.8	2.7	2.1	2.1	1.5	5.2
bodyfat	40.3	9.3	7.9	3.7	2.0	9.1
mpg	62.7	4.3	2.4	2.8	2.3	14.5
housing	81.0	8.4	4.6	3.7	5.0	17.5
add10	160.0	7.8	6.6	6.6	6.8	121.2
cpusmall	160.0	16.0	8.0	4.6	5.8	144.6
space_ga	160.0	6.0	6.8	6.8	5.8	43.4
abalone	160.0	13.2	6.4	7.2	8.2	135.6

(a) Pre-specified probability = 80%.

Problem	#95%	Gau	Lap	Lap*	Hist	BSVR1
pyrim	14.1	0.7	0.5	0.7	0.7	0.7
triazines	35.3	1.1	0.9	0.9	0.9	2.2
bodyfat	47.9	1.7	1.3	0.9	1.1	1.7
mpg	74.5	0.7	0.6	0.6	0.7	3.7
housing	96.1	2.2	2.2	2.2	2.2	7.8
add10	190.0	3.6	7.8	7.8	3.8	138.8
cpusmall	190.0	4.0	3.4	2.8	1.0	168.4
space_ga	190.0	3.4	3.2	3.2	3.2	40.6
abalone	190.0	3.8	2.6	2.8	4.2	157.8

(b) Pre-specified probability = 95%.

Thus the density function of δ is

$$p(\delta) = \frac{1}{Z_D}\exp(-C\ell_{\epsilon,\beta}(\delta)), \tag{22}$$

where $Z_D = \int \exp(-C\ell_{\epsilon,\beta}(\delta))d\delta$. Using (11) and (21), σ_δ^2 is

$$\begin{aligned} \frac{2}{Z_s}\Bigg\{ & \frac{(1-\beta)^3\epsilon^3}{3} + \sqrt{\frac{\pi\beta\epsilon}{C}}\big(\frac{2\beta\epsilon}{C} + (1-\beta)^2\epsilon^2\big)\mathrm{erf}(\sqrt{C\beta\epsilon}) \\ & + \frac{4(1-\beta)\beta\epsilon^2}{C} + \big(\frac{\epsilon^2(1-\beta)^2}{C} \\ & + \frac{2\epsilon(1+\beta)}{C^2} + \frac{2}{C^3}\big)\exp(-C\beta\epsilon)\Bigg\} \end{aligned} \tag{23}$$

and the prediction variance is $\sigma_\delta^2 + \sigma_t^2$.

An important difference in [2] is the use of the kernel

$$K(\mathbf{x}_i, \mathbf{x}_j) = \kappa_0 \exp\big(-\frac{\kappa}{2}\|\mathbf{x}_i - \mathbf{x}_j\|^2\big) + \kappa_b. \tag{24}$$

TABLE V

AVERAGE ABSOLUTE DIFFERENCE ON NUMBER OF COVERAGES: MAXIMIZING BSVR1 EVIDENCE FUNCTION FOR PARAMETER SELECTION.

Problem	#80%	Gau	Lap	Lap*	Hist	BSVR1
pyrim	11.8	1.4	0.6	0.8	1.0	2.4
triazines	29.8	1.8	1.5	1.5	1.7	1.7
bodyfat	40.3	6.1	3.7	2.4	2.9	8.9
mpg	62.7	5.9	3.7	4.3	4.5	5.0
housing	81.0	7.8	4.8	4.8	5.4	8.2
add10	160.0	9.0	8.8	8.8	9.6	13.6
cpusmall	160.0	32.0	21.8	5.0	8.4	75.8
space_ga	160.0	7.6	4.6	4.6	5.4	17.4
abalone	160.0	14.4	7.0	5.8	5.8	10.4

(a) Pre-specified probability = 80%.

Problem	#95%	Gau	Lap	Lap*	Hist	BSVR1
pyrim	14.1	0.8	0.8	0.7	0.5	0.5
triazines	35.3	1.8	1.6	1.6	2.4	2.5
bodyfat	47.9	1.9	1.3	1.9	2.7	2.1
mpg	74.5	1.4	1.2	0.8	1.4	4.3
housing	96.1	2.4	2.6	2.2	1.0	5.8
add10	190.0	4.6	8.0	8.0	5.8	4.8
cpusmall	190.0	4.4	2.8	4.0	2.6	44.8
space_ga	190.0	2.8	3.2	3.2	3.2	5.0
abalone	190.0	3.4	3.4	3.6	3.2	1.4

(b) Pre-specified probability = 95%.

Thus, instead of one parameter γ in the RBF kernel, here three have to be decided, and the hyperparameter is $\theta = (\kappa_0, \kappa, \kappa_b, C, \epsilon)$.

In the rest of this paper we refer to the Bayesian methods in [4] and [2] as BSVR1 and BSVR2, respectively.

IV. EXPERIMENTS

We compare the proposed approach with the two Bayesian methods reviewed in Section III. Several regression problems are considered: Problems housing, abalone, mpg, pyrim, and triazines are from the Statlog collection [9]; bodyfat and space_ga are from StatLib (`http://lib.stat.cmu.edu/datasets`); Problems add10 and cpusmall are from the Delve archive (`http://www.cs.toronto.edu/~delve`). For these problems, some data entries have missing attributes so we remove them before conducting experiments. Note that the attribute values of these problems are scaled to $[-1, +1]$, but target values are kept the same. To save the computational time, for problems with more than 1,000 instances, only a random subset of 1,000 points are used. The numbers

TABLE VI

ERROR ON COVERAGE: MAXIMIZING BSVR2 EVIDENCE FUNCTION FOR PARAMETER SELECTION.

Problem	#80%	Gau	Lap	Lap*	Hist	BSVR2
pyrim	11.8	2.2	2.0	1.6	2.2	1.6
triazines	29.8	3.3	2.3	2.3	1.9	2.3
bodyfat	40.3	9.1	7.7	3.2	3.0	9.1
mpg	62.7	5.5	3.3	2.7	2.3	3.7
housing	81.0	11.6	5.8	3.5	2.7	7.8
add10	160.0	12.6	9.6	9.6	9.6	10.6
cpusmall	160.0	18.0	6.8	4.4	2.6	7.8
space_ga	160.0	9.0	5.4	4.8	4.8	4.8
abalone	160.0	13.6	5.0	4.0	9.2	11.4

(a) Difference to 80% coverage.

Problem	#95%	Gau	Lap	Lap*	Hist	BSVR2
pyrim	14.1	0.7	0.6	0.5	1.0	0.7
triazines	35.3	1.0	1.0	1.0	1.4	1.3
bodyfat	47.9	1.7	1.3	0.9	0.9	1.5
mpg	74.5	0.7	0.6	0.6	1.4	1.8
housing	96.1	1.4	1.0	1.6	2.0	1.5
add10	190.0	4.0	3.6	3.6	4.6	5.0
cpusmall	190.0	3.6	2.8	3.0	2.2	2.4
space_ga	190.0	3.4	2.8	2.2	2.0	2.8
abalone	190.0	2.8	2.8	3.2	2.0	3.8

(b) Difference to 95% coverage.

of data instances and features are reported in Table I.

In the experiment, each data set is separated to five folds and sequentially one fold is used for testing and the remaining are for training. To have a good model, parameter selection is conducted on the training set. We consider the following methods:

1) Cross validation: $(C, \gamma, \epsilon) = [2^{-1}, 2^{0}, \ldots, 2^{6}] \times [2^{-8}, 2^{-7}, \ldots, 2^{1}] \times [2^{-8}, 2^{-7}, \ldots, 2^{1}]$ are tried and the one with the highest five-fold CV accuracy is used to train the model for testing. For this setting, BSVR2 is not compared as its implementation uses (24), a kernel with more parameters.
2) Maximization of the evidence function $P(\mathcal{D} \mid \theta)$ of BSVR1: We search the same space of (C, γ, ϵ) used in 1) and choose the one which gives the maximal value of the evidence function. Similar to using CV for parameter selection, BSVR2 is not compared.
3) Maximization of the evidence function $P(\mathcal{D} \mid \theta)$ of BSVR2: Now there are five parameters $C, \kappa, \kappa_0, \kappa_b$, and

ϵ. $P(\mathcal{D} \mid \theta)$ is maximized by a gradient-based implementation used in [2]. For this setting we did not compare BSVR1 as its kernel implementation must be changed. On the contrary, it is still easy to use the proposed approaches as their implementations are independent of parameters.

Implementation details and experimental results are given in the following subsections.

A. Implementation Details

Given a pre-specified probability $1-2s$, the performance of various approaches is evaluated by comparing the number of testing data lying in their prediction intervals with the expected number, $(1-2s) \times (\#$ test set$)$. For each $(\mathbf{x}, y)$ in the test set, the prediction interval for y is $(\hat{f}(\mathbf{x}) - p_s, \hat{f}(\mathbf{x}) + p_s)$, where p_s is the upper sth percentile of the corresponding probability distribution of $\zeta (= y - \hat{f}(\mathbf{x}))$. Therefore, we simply count the number of ζ in the test set lying in $[-p_s, p_s]$, and compare this number with its expected value.

For a zero-mean symmetric variable Z with density $p(z)$, p_s can be determined by solving

$$\int_{-\infty}^{p_s} p(z)dz = 1 - s.$$

For example, a Gaussian with $p(z)$ defined in (3) has $p_s = \sigma^{-1}\Phi^{-1}(1-s)$, where $\Phi(x) \equiv \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-z^2/2} dz$, and hence the prediction interval for ζ is

$$(-\sigma^{-1}\Phi^{-1}(1-s), \sigma^{-1}\Phi^{-1}(1-s)); \tag{25}$$

a Laplace with $p(z)$ as in (2) has $p_s = -\sigma \ln(2s)$, and the resulting interval is

$$(\sigma \ln(2s), -\sigma \ln(2s)). \tag{26}$$

For the variable ζ in (17) under the BSVR1 setting, we denote p_ζ as the density of ζ and write

$$\begin{aligned} 1 - s &= \int_{-\infty}^{p_s} p_\zeta(z)dz \\ &= \int_{-\infty}^{\infty} \int_{-\infty}^{p_s - f} p(\delta)d\delta p_{f|\mathcal{D}}(f)df, \end{aligned}$$

where $p(\delta)$ and $p(\delta)d\delta p_{f|\mathcal{D}}(f)$ are as in (14) and (16). Since the integral of (14) over a certain range can be derived explicitly, the convolution here is reduced to a one-dimensional integration. Then this percentile problem is resolved by numerical integration. Similar treatment can be applied to BSVR2, with $p(\delta)$ in (14) replaced by (22). Note that for Bayesian approach the distribution of ζ depends on the input value $\mathbf{x}$, and so does p_s. Therefore, the numerical integration needs be carried out for each test instance.

For non-symmetric distributions like the discretized histogram, we simply sort ζ_i's into the form $\zeta_{(1)} < \zeta_{(2)} < \cdots < \zeta_{(l)}$, and then set the $(1-2s)100\%$ prediction interval for ζ as $(\zeta_{s \cdot (\# \text{ test set})}, \zeta_{(1-s) \cdot (\# \text{ test set})})$. Below we refer to this method as "Hist," which will be compared with other approaches as well.

Here are more implementation details. For searching the best parameters by CV and then calculating the coverages, we use the software LIBSVM [1], which solves the standard SVR (9). As BSVR1 uses the formulation without the bias term b, we modify LIBSVM for such a form and use it to evaluate the evidence function. For BSVR2, the implementation from [2] is adopted.

TABLE VII

COMPARISON BETWEEN CV AND BAYES FOR PARAMETER SELECTION: MSE AND AVERAGE NUMBER OF SUPPORT VECTORS.

	CV		BSVR1		BSVR2	
Problem	MSE	#SV	MSE	#SV	MSE	#SV
pyrim	0.0467	44.2	0.0864	58.6	0.0490	38.0
triazines	0.1395	83.4	0.1734	147.0	0.1291	99.0
bodyfat	0.0027	62.6	0.0087	174.2	0.0029	82.0
mpg	0.0196	149.4	0.0237	300.0	0.0193	204.0
housing	0.0214	212.6	0.0239	382.0	0.0230	272.0
add10	1.9466	634.8	8.0780	743.0	2.8555	782.0
cpusmall	16.5119	722.2	180.0748	755.0	15.9123	786.0
space_ga	0.0131	465.8	0.0128	623.6	0.0149	614.0
abalone	5.3678	684.2	7.8067	757.2	5.5827	779.0

B. *Results and a New Method "Lap*"*

We first use the dataset cpusmall to describe some experimental findings. Table II reports the results of parameter selection and the number of test instances lying in the predictive intervals for each of the five training/testing splits. Here the parameters are selected by the different strategies described earlier. In this experiment, the test size is 200 for each split and s is set as 0.1, so the coverage probability is 0.8 and the expected number of instances being covered is 160. Recall the description under Figure 1 that there are a couple of extreme values of ζ_i's, with the maximum as large as 51.5. Consequently, the estimate of the scale parameter σ is quite large and the resulting prediction interval ((25) or (26)) is too wide. This justifies why "Gau" and "Lap" tend to over-cover the test instances. Therefore, we propose to re-estimate the scale parameter by discarding the "very extreme" ζ_i's. Here ζ_i's are called "very extreme" if they exceed $\pm 5 \times$ (standard deviation of distribution). The resulting coverages are then shown in the fourth column, entitled as "Lap*." We study another problem housing in Table III. Results are similar.

Tables IV-VI present the results for all the datasets using different parameter selection strategies. Here we simply report the average absolute difference over the five splits. For each split, the absolute difference is

$$|\# \text{ of } \zeta \text{ in } [-p_s, p_s] - (1-2s) \times (\# \text{ test set})|$$

with $s = 0.1$ and 0.025, corresponding to coverage probabilities $1 - 2s = 80\%$ and 95%. For example, the absolute differences for cpusmall using CV and "Lap." are 14, 10, 6, 1, and 9, as given in Table II(a), and hence the averaged value is 8.0. A by-product of this experiment is the comparison between standard SVR and the two Bayesian SVRs. Table VII presents the averages of MSEs and the numbers of support vectors over the five training/testing splits.

V. Analysis

We first consider the results of cpusmall in Table II. Table II(a) shows that, with the model parameters selected by CV, BSVR1 severely under-covers the test instances (the expected number is 160). This phenomenon can be explained by the influence of the parameters on σ_δ^2 in (15). The parameter $(C, \epsilon) = (64, 0.5)$ leads to $\sigma_\delta^2 = 4.8 \times 10^{-4} + 8 \times 10^{-2}$, which is rather small and causes the prediction interval to be too narrow to cover most test instances. For the third training/testing split, $(C, \epsilon) = (64, 0.004)$ results in an even smaller σ_δ^2, and hence an even worse coverage. Though the situation has been substantially improved when choosing the model parameter as the maximizer of the Bayesian evidence function (12), the results are still way below the expected value 160. This result indicates that the evidence function of BSVR1 is not accurate. MSE shown in Table VII further confirms such a conclusion. On the other hand, BSVR2 gives good results in Table II(c). Its MSE in Table VI is competitive with that by CV for parameter selection. The performance of BSVR2 indicates that its Bayesian evidence function (19) is accurate, and the use of a more general kernel function may also help. Regarding proposed methods, "Gau," "Lap," and "Lap*" all produced reasonable coverages no matter using which method for parameter selection. In general, "Lap*" improves upon "Lap," and for all the five training/testing splits "Lap" outperforms "Gau." The results in Table III can be explained similarly.

For other datasets using CV as the parameter selection strategy, Table IV shows that, like the previous two tables, BSVR1 is worse than others. The overall performance of "Lap" is better than that of "Gau." Indeed the results of the most powerful test (6) are in favor of Laplace for all the nine datasets. This seems to indicate that, for these problems, $y - \hat{f}(\mathbf{x})$ is more like a Laplace rather than a Gaussian. The results of "Lap" and "Lap*" are satisfactory. The advantage of using "Lap*" over "Lap" is apparent on datasets bodyfat and cpusmall in Table IV(a), where the averaged absolute errors have been cut to nearly half. The "Hist" produces nice results as well. The main reason is that "Hist" directly makes use of information from ζ_i's, instead of assuming a symmetric model with zero. However, as mentioned in Section II, it is more complex as all ζ_i's must be retained.

Tables V and VI report results with model parameters maximizing the Bayesian evidence functions (12) and (19), respectively. For BSVR1, the coverages are much better than those in Table IV; however, in general it is still the worst among all. Again, like Table II(b), this can be explained by the choice of the parameter set and its effect on σ_δ^2. For the proposed methods, as before, "Lap" outperforms "Gau" in almost all cases, "Lap*" further improves "Lap," and "Hist" is quite competitive.

In summary, the experimental results indicate that the Bayesian error depends on different Bayesian evidence functions. As our proposed methods are not related to parameters, they are quite stable for different parameter selection methods.

Regarding the MSEs of CV and two Bayesian methods, Table VII shows that CV and BSVR2 are better. This result is consistent with those in previous tables. Better target value prediction also leads to better probability intervals.

VI. Conclusions

In this paper, we propose a simple approach for probabilistic prediction suitable for the standard SVR. Our approach starts with generating out-of-sample residuals by cross validation, and then fits the residuals by simple parametric models like Gaussian and Laplace. The most powerful scale-invariant test is applied to effectively test Gaussian against Laplace. We then compare it with the Bayesian SVR methods by evaluating the performance of the prediction intervals. The experiments on real-world problems show that our easy approach works fairly well and is robust to parameter selection strategies. Moreover, in certain cases we can further improve upon our approach by re-estimate the scale parameter of the Laplace family. In summary, though we assume that the distribution of the target value depends on its input only through the predicted value, the proposed approach easily provides some useful probability information for SVR analysis.

References

[1] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[2] W. Chu, S. Keerthi, and C. J. Ong. Bayesian support vector regression using a unified loss function. *IEEE Transactions on Neural Networks*, 15:29–44, 2004.

[3] B. Efron. Estimating the error rate of a prediction rule: Improvement on cross-validation. *Journal of the American Statistical Association*, 78(382):316–331, 1983.

[4] J. B. Gao, S. R. Gunn, C. J. Harris, and M. Brown. A probabilistic framework for SVM regression and error bar estimation. *Machine Learning*, 46:71–89, 2002.

[5] R. V. Hogg. More light on the kurtosis and related statistics. *Journal of the American Statistical Association*, 67:422–424, 1972.

[6] M. H. Law and J. T. Kwok. Bayesian support vector regression. In *Proceedings of the Eighth International Workshop on Artificial Intelligence and Statistics (AISTATS)*, pages 239–244, Key West, Florida, USA, 2001.

[7] E. L. Lehmann. *Testing statistical hypotheses*. Wiley, New York, 2nd edition, 1986.

[8] D. J. C. MacKay. A practical Bayesian framework for backpropagation networks. *Neural Computation*, 4(3):448–472, 1992.

[9] D. Michie, D. J. Spiegelhalter, and C. C. Taylor. *Machine Learning, Neural and Statistical Classification*. Prentice Hall, Englewood Cliffs, N.J., 1994. Data available at `http://www.ncc.up.pt/liacc/ML/statlog/datasets.html`.

[10] J. Platt. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, editors, *Advances in Large Margin Classifiers*, Cambridge, MA, 2000. MIT Press.

[11] M. Seeger. Bayesian model selection for support vector machines. In S. A. Solla, T. K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems*, volume 12, pages 603–609, 2000.

[12] A. J. Smola and B. Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004.

[13] P. Sollich. Bayesian methods for support vector machines: Evidence and predictive class probabilities. *Machine learning*, 46:21–52, 2002.

[14] R. A. Stine. Bootstrap prediction intervals for regression. *Journal of the American Statistical Association*, 80(392):1026–1029, 1985.

[15] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, NY, 1995.

[16] S. Weisberg. *Applied Linear Regression*. John Wiley and Sons, 1985.