Scientific Research Publishing

Journal of Applied Mathematics and Physics, 2020, 8, 1877-1887
https://www.scirp.org/journal/jamp
ISSN Online: 2327-4379
ISSN Print: 2327-4352

A Bayesian Regression Model and Applications

Yijun Yu

Department of Mathematics, Tuskegee University, Tuskegee, AL, USA
Email: yyu@tuskegee.edu

How to cite this paper: Yu, Y.J. (2020) A Bayesian Regression Model and Applications. *Journal of Applied Mathematics and Physics*, **8**, 1877-1887.
https://doi.org/10.4236/jamp.2020.89141

Received: July 17, 2020
Accepted: September 19, 2020
Published: September 22, 2020

Abstract

A sparse vector regression model is developed. The model is established by employing Bayesian formulation and trained by using a set of data $\boldsymbol{D}=\{\boldsymbol{x}_n,\boldsymbol{t}_n\}_{n=1}^N$. The parameters needed to be determined in the algorithm are reduced by a special prior hyperparameter setting, and therefore the algorithm is simpler than similar type of Bayesian vector regression models. The examples of applications to the function approximation and inverse scattering problem are presented.

Keywords

Bayesian, Regression, Applications

1. Introduction

There has been a lot of interest in studying the Bayesian vector regression and its application on various classification and regression problems [1] [2] [3] [4]. The Bayesian approach considers probability distributions with the observed data; prior distributions are converted to posterior distribution through the use of Bayes' theorem. Let $\boldsymbol{x}$ be an input vector and $\boldsymbol{t}$ be a vector of target parameters. In a regression formulation our goal is to define a model $\mathbf{y}(\boldsymbol{x};\boldsymbol{w})$ that yields an approximation to the true target $\boldsymbol{t}$, with the model defined by the parameters $\boldsymbol{w}$. The model is typically designed using a set of "training" data $\boldsymbol{D}=\{\boldsymbol{x}_n,\boldsymbol{t}_n\}_{n=1}^N$, Although we initially consider a finite set $\boldsymbol{D}$, the goal is for the subsequent model $\mathbf{y}(\boldsymbol{x};\boldsymbol{w})$ to be applicable to arbitrary $(\boldsymbol{x},\boldsymbol{t})\notin\boldsymbol{D}$, over the anticipated range of $\boldsymbol{t}$. When developing a regression model one must address the bias-variance tradeoff. A bias is introduced by restricting the form that $\mathbf{y}(\boldsymbol{x};\boldsymbol{w})$ may take, while the variance represents the error between the model $\mathbf{y}(\boldsymbol{x};\boldsymbol{w})$ and true target parameters $\boldsymbol{t}$. Models with minimal bias typically have significant flexibility, and therefore the model parameters may vary significantly as a function of the spe-

cific training set $\boldsymbol{D}$ employed. To obtain good model generalization, which may be connected to the variation in the model parameters as a function of $\boldsymbol{D}$, one must introduce a bias. The utilization of a small number of non-zero parameters $\boldsymbol{w}$ often yields a good balance between bias and variance; such models are termed "sparse". This has led to development of the relevance vector machine [5].

The rest of this paper is organized as follows. The theory of the vector-regression formulation is presented in Section 2, with application example provided in Section 3. The work is summarized in Section 4.

2. Sparse Bayesian Vector Regression

2.1. Model Specification

Assume we have available a set of training data $\boldsymbol{D}=\{\boldsymbol{x}_n,\boldsymbol{t}_n\}_{n=1}^N$, where $\boldsymbol{x}_n=\left[x_n^{(1)}\ x_n^{(2)}\cdots x_n^{(L)}\right]^{\mathrm{T}}$ and $\boldsymbol{t}_n=\left[t_n^{(1)}\ t_n^{(2)}\cdots t_n^{(M)}\right]^{\mathrm{T}}$. Our objective is to develop a function $\mathbf{y}(\boldsymbol{x};\boldsymbol{w})$ that is dependent on the parameters $\boldsymbol{w}$. After $\mathbf{y}(\boldsymbol{x};\boldsymbol{w})$ is so designed, it may be used to map an arbitrary $\boldsymbol{x}$ to an approximation of the target parameters $\boldsymbol{t}$.

The specific vector-regression function $\mathbf{y}(\boldsymbol{x};\boldsymbol{w})=\left[y^{(1)}(\boldsymbol{x};\boldsymbol{w})\ y^{(2)}(\boldsymbol{x};\boldsymbol{w})\cdots y^{(M)}(\boldsymbol{x};\boldsymbol{w})\right]^{\mathrm{T}}$ employed here is defined as

$$\mathbf{y}(\boldsymbol{x};\boldsymbol{w})=\sum_{i=1}^{N}w_i\boldsymbol{t}_iK(\boldsymbol{x},\boldsymbol{x}_i)+\boldsymbol{w}_0 \tag{1}$$

where $\boldsymbol{w}_0=\left[w_0^{(1)}\ w_0^{(2)}\cdots w_0^{(M)}\right]^{\mathrm{T}}$, and $K(\boldsymbol{x},\boldsymbol{x}_i)$ is a kernel function that is designed such that $K(\boldsymbol{x},\boldsymbol{x}_i)$ is large if $\boldsymbol{x}_i\approx\boldsymbol{x}$ and otherwise $K(\boldsymbol{x},\boldsymbol{x}_i)$ is small. Hence in (1) only those $\boldsymbol{x}_i\approx\boldsymbol{x}$ are important in defining $\mathbf{y}(\boldsymbol{x};\boldsymbol{w})$.

Let

$$\boldsymbol{w}=\left[w_1\ w_2\cdots w_N\ w_0^{(1)}\ w_0^{(2)}\cdots w_0^{(M)}\right]^{\mathrm{T}},$$

$$\boldsymbol{\psi}_i(\boldsymbol{x})=\left[\phi_i^{(1)}\ \phi_i^{(2)}\cdots\phi_i^{(M)}\right]^{\mathrm{T}},\quad i=1,2,\cdots,N$$

with

$$\phi_i^{(k)}=t_i^{(k)}K(\boldsymbol{x},\boldsymbol{x}_i),\ i=1,2,\cdots,N;k=1,2,\cdots,M \tag{2}$$

and $M\times(N+M)$ matrix

$$\boldsymbol{\Psi}(\boldsymbol{x})=\left[\boldsymbol{\psi}_1(\boldsymbol{x})\boldsymbol{\psi}_2(\boldsymbol{x})\cdots\boldsymbol{\psi}_N(\boldsymbol{x})\boldsymbol{I}_M\right], \tag{3}$$

where $\boldsymbol{I}_M$ is $M\times M$ identity matrix, then (1) can be expressed in matrix form

$$\mathbf{y}(\boldsymbol{x};\boldsymbol{w})=\boldsymbol{\Psi}(\boldsymbol{x})\boldsymbol{w} \tag{4}$$

Assume that target is from the model with additive noise

$$\boldsymbol{t}=\mathbf{y}(\boldsymbol{x};\boldsymbol{w})+\boldsymbol{\varepsilon}=\boldsymbol{\Psi}(\boldsymbol{x})\boldsymbol{w}+\boldsymbol{\varepsilon}, \tag{5}$$

where model error $\boldsymbol{\varepsilon}=\left[\varepsilon^{(1)}\ \varepsilon^{(2)}\cdots\varepsilon^{(M)}\right]^{\mathrm{T}}$ and $\varepsilon^{(k)},k=1,2,\cdots,M$ are inde-

pendent samples from a zero-mean Gaussian process with variance α_0^{-1}

$$\boldsymbol{p}\left(\varepsilon^{(k)}\right)=N\left(\varepsilon^{(k)}\,|\,0,\alpha_0^{-1}\right),\ \ k=1,2,\cdots,M \tag{6}$$

We therefore have

$$\begin{aligned}\boldsymbol{p}\left(\boldsymbol{t}\,|\,\boldsymbol{x},\boldsymbol{w},\alpha_0\right)&=\left(\frac{2\pi}{\alpha_0}\right)^{-\frac{M}{2}}\exp\left(-\frac{\alpha_0}{2}\left\|\boldsymbol{t}-\boldsymbol{\Psi}\left(\boldsymbol{x}\right)\boldsymbol{w}\right\|_2^2\right)\\&=N\left(\boldsymbol{t}\,|\,\boldsymbol{\Psi}\left(\boldsymbol{x}\right)\boldsymbol{w},\alpha_0^{-1}\boldsymbol{I}_M\right)\end{aligned} \tag{7}$$

We wish to constrain the weights $\boldsymbol{w}$ such that a simple model is favored, this accomplished by invoking a prior distribution on $\boldsymbol{w}$ that favors most of the weights being zero. In this context, only the most relevant members of the training set $\boldsymbol{D}=\left\{\boldsymbol{x}_n,\boldsymbol{t}_n\right\}_{n=1}^N$, those with nonzero weights w_n, are ultimately used in the final regression model. This simplicity allows improved regression performance for $\left(\boldsymbol{x},\boldsymbol{t}\right)\notin\boldsymbol{D}$ [5] [6].

We employ a zero-mean Gaussian prior distribution for $\boldsymbol{w}$

$$\boldsymbol{p}\left(\boldsymbol{w}\,|\,\alpha_0,\alpha\right)=N\left(\boldsymbol{w}\,|\,\boldsymbol{0}_{N+M},\alpha_0^{-1}\alpha^{-1}\boldsymbol{I}_{N+M}\right), \tag{8}$$

where $\boldsymbol{0}_{N+M}$ is a $(N+M)$-dimensional zero vector, $\boldsymbol{I}_{N+M}$ is a $\left(N+M\right)\times\left(N+M\right)$ identity matrix, and suitable priors over hyperparameters α_0 and α are Gamma distributions [7]

$$p\left(\alpha_0\,|\,a,b\right)=\mathrm{Gamma}\left(\alpha_0\,|\,a,b\right) \tag{9}$$

$$p\left(\alpha\,|\,c,d\right)=\mathrm{Gamma}\left(\alpha\,|\,c,d\right) \tag{10}$$

where $\mathrm{Gamma}\left(\alpha_0\,|\,a,b\right)=\Gamma\left(a\right)^{-1}b^a\alpha_0^{a-1}\mathrm{e}^{-b\alpha_0}$ with $\Gamma\left(a\right)=\int_0^\infty t^{a-1}\mathrm{e}^{-t}\mathrm{d}t$.

The hierarchical prior over $\boldsymbol{w}$ favors a sparse model and the prior over α_0 will be used to favor small model error on the training data $\boldsymbol{D}$.

2.2. Inference

For training data $\boldsymbol{D}=\left\{\boldsymbol{x}_n,\boldsymbol{t}_n\right\}_{n=1}^N$ we introduce LN-dimensional vector

$$\mathbf{X}=\left[\boldsymbol{x}_1^{\mathrm{T}}\ \boldsymbol{x}_2^{\mathrm{T}}\cdots\boldsymbol{x}_N^{\mathrm{T}}\right]^{\mathrm{T}}$$

and MN-dimensional vector

$$\mathbf{T}=\left[\boldsymbol{t}_1^{\mathrm{T}}\ \boldsymbol{t}_2^{\mathrm{T}}\cdots\boldsymbol{t}_N^{\mathrm{T}}\right]^{\mathrm{T}}$$

and let $\left(MN\right)\times\left(M+N\right)$ matrix

$$\boldsymbol{\Phi}=\left[\boldsymbol{\Phi}_1^{\mathrm{T}}\ \boldsymbol{\Phi}_2^{\mathrm{T}}\cdots\boldsymbol{\Phi}_N^{\mathrm{T}}\right]^{\mathrm{T}}\ \text{ with }\ \boldsymbol{\Phi}_i=\boldsymbol{\Psi}\left(x_i\right),i=1,2,\cdots,N,$$

then by (7), we have

$$\begin{aligned}p\left(\mathbf{T}\,|\,\boldsymbol{w},\alpha_0,\mathbf{X}\right)&=\left(\frac{2\pi}{\alpha_0}\right)^{-\frac{MN}{2}}\exp\left(-\frac{\alpha_0}{2}\left\|\mathbf{T}-\boldsymbol{\Phi}\boldsymbol{w}\right\|_2^2\right)\\&=N\left(\mathbf{T}\,|\,\boldsymbol{\Phi}\boldsymbol{w},\alpha_0^{-1}\boldsymbol{I}_{MN}\right)\end{aligned} \tag{11}$$

Noting that $p\left(\mathbf{T}\,|\,\alpha_0,\alpha,\mathbf{X}\right)=\int p\left(\mathbf{T}\,|\,\boldsymbol{w},\alpha_0,\mathbf{X}\right)\boldsymbol{p}\left(\boldsymbol{w}\,|\,\alpha_0,\alpha\right)\mathrm{d}\boldsymbol{w}$ is a convolu-

tion of Gaussians, the posterior distribution over the weights $\boldsymbol{w}$ can be derived as

$$p\left(\boldsymbol{w}\,|\,\alpha_0,\alpha,\mathbf{X},\mathbf{T}\right)=\frac{p\left(\mathbf{T}\,|\,\boldsymbol{w},\alpha_0,\mathbf{X}\right)p\left(\boldsymbol{w}\,|\,\alpha_0,\alpha\right)}{p\left(\mathbf{T}\,|\,\alpha_0,\alpha,\mathbf{X}\right)}=N\left(\boldsymbol{w}\,|\,\boldsymbol{\mu},\alpha_0^{-1}\boldsymbol{\Sigma}\right) \tag{12}$$

where

$$\boldsymbol{\Sigma}=\left(\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}+\alpha\boldsymbol{I}_{M+N}\right)^{-1}=\left(\sum_{i=1}^{N}\boldsymbol{\Phi}_i^{\mathrm{T}}\boldsymbol{\Phi}_i+\alpha\boldsymbol{I}_{M+N}\right)^{-1} \tag{13}$$

$$\boldsymbol{\mu}=\boldsymbol{\Sigma}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{T}=\boldsymbol{\Sigma}\sum_{i=1}^{N}\left(\boldsymbol{\Phi}_i t_i\right) \tag{14}$$

2.3. Hyperparameter Optimization

We determine α in (13) by maximizing $p\left(\alpha\,|\,\mathbf{T},\mathbf{X}\right)\propto p\left(\mathbf{T}\,|\,\alpha,\mathbf{X}\right)p\left(\alpha\right)$ with respect to α. It is equivalent to maximize the ln of this quantity. In addition, we can choose to maximize with respect to $\ln\alpha$ as we can assume hyperpriors over a logarithmic scale.

Since

$$\begin{aligned}&\ln p\left(\mathbf{T}\,|\,\alpha,\mathbf{X}\right)\\&=\ln\int p\left(\mathbf{T}\,|\,\boldsymbol{w},\alpha_0,\mathbf{X}\right)\boldsymbol{p}\left(\boldsymbol{w}\,|\,\alpha_0,\alpha\right)p\left(\alpha_0\,|\,a,b\right)\mathrm{d}\boldsymbol{w}\mathrm{d}\alpha_0\\&=-\frac{1}{2}\left[\ln\left|\mathbf{B}\right|+\left(MN+2a\right)\ln\left(\mathbf{T}^{\mathrm{T}}\mathbf{B}^{-1}\mathbf{T}+2b\right)\right]+const\end{aligned}$$

where $\mathbf{B}=\boldsymbol{I}_{MN}+\alpha^{-1}\boldsymbol{\Phi}\boldsymbol{\Phi}^{\mathrm{T}}$, and $p\left(\ln\alpha\right)=\alpha p\left(\alpha\right)$, we obtain objective function

$$L\left(\alpha\right)=-\frac{1}{2}\left[\ln\left|\mathbf{B}\right|+\left(MN+2a\right)\ln\left(\mathbf{T}^{\mathrm{T}}\mathbf{B}^{-1}\mathbf{T}+2b\right)\right]+c\ln\alpha-d\alpha \tag{15}$$

By the determinant identity [8], we have

$$\begin{aligned}\left|\mathbf{B}\right|&=\left|I_{MN}+\alpha^{-1}\boldsymbol{\Phi}\boldsymbol{\Phi}^{\mathrm{T}}\right|\\&=\alpha^{-(M+N)}\left|\alpha I_{M+N}+\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right|\\&=\alpha^{-(M+N)}\left|\boldsymbol{\Sigma}^{-1}\right|,\end{aligned}$$

and so

$$\ln\left|\mathbf{B}\right|=-\left(M+N\right)\ln\alpha+\ln\left|\boldsymbol{\Sigma}^{-1}\right| \tag{16}$$

Using the Woodbury formula, we obtain

$$\begin{aligned}\mathbf{B}^{-1}&=\left(I_{MN}+\alpha^{-1}\boldsymbol{\Phi}\boldsymbol{\Phi}^{\mathrm{T}}\right)^{-1}\\&=I_{MN}-\boldsymbol{\Phi}\left(\alpha I_{M+N}+\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{\Phi}\right)^{-1}\boldsymbol{\Phi}^{\mathrm{T}}\\&=I_{MN}-\boldsymbol{\Phi}\boldsymbol{\Sigma}\boldsymbol{\Phi}^{\mathrm{T}},\end{aligned}$$

thus

$$\begin{aligned}\mathbf{T}^{\mathrm{T}}\mathbf{B}^{-1}\mathbf{T}&=\mathbf{T}^{\mathrm{T}}\left(\mathbf{T}-\boldsymbol{\Phi}\boldsymbol{\Sigma}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{T}\right)\\&=\mathbf{T}^{\mathrm{T}}\left(\mathbf{T}-\boldsymbol{\Phi}\boldsymbol{\mu}\right) &(17)\\&=\left\|\mathbf{T}\right\|^2-\mathbf{T}^{\mathrm{T}}\boldsymbol{\Phi}\boldsymbol{\Sigma}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{T} &(18)\end{aligned}$$

Then by (16) and Jacobi's formula, we have

$$\begin{aligned}\frac{\mathrm{d}\ln|\mathbf{B}|}{\mathrm{d}\ln\alpha} &= -(M+N)+\frac{1}{\left|\boldsymbol{\Sigma}^{-1}\right|}\frac{\mathrm{d}\left|\boldsymbol{\Sigma}^{-1}\right|}{\mathrm{d}\ln\alpha}\\ &= -(M+N)+tr\left(\boldsymbol{\Sigma}\frac{\mathrm{d}\boldsymbol{\Sigma}^{-1}}{\mathrm{d}\ln\alpha}\right)\\ &= -(M+N)+\alpha\sum_{j=1}^{M+N}\boldsymbol{\Sigma}_{jj}\end{aligned} \tag{19}$$

where $\boldsymbol{\Sigma}_{jj}$ is the j-th diagonal element of matrix $\boldsymbol{\Sigma}$.

By (18)

$$\begin{aligned}\frac{\mathrm{d}\mathbf{T}^{\mathrm{T}}\mathbf{B}^{-1}\mathbf{T}}{\mathrm{d}\ln\alpha} &= -\frac{\mathrm{d}\mathbf{T}^{\mathrm{T}}\boldsymbol{\Phi}\boldsymbol{\Sigma}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{T}}{\mathrm{d}\ln\alpha}\\ &= -\mathbf{T}^{\mathrm{T}}\boldsymbol{\Phi}\frac{\mathrm{d}\boldsymbol{\Sigma}}{\mathrm{d}\ln\alpha}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{T}\\ &= -\mathbf{T}^{\mathrm{T}}\boldsymbol{\Phi}\boldsymbol{\Sigma}\frac{\mathrm{d}\boldsymbol{\Sigma}^{-1}}{\mathrm{d}\ln\alpha}\boldsymbol{\Sigma}\boldsymbol{\Phi}^{\mathrm{T}}\mathbf{T}\\ &= \alpha\|\boldsymbol{\mu}\|^2\end{aligned} \tag{20}$$

Using (17), (19) and (20), we have

$$\begin{aligned}\frac{\mathrm{d}L(\alpha)}{\mathrm{d}\alpha} &= \frac{1}{2}\left(M+N-\alpha\sum_{j=1}^{M+N}\boldsymbol{\Sigma}_{jj}\right)-\frac{(MN+2a)}{2\left(\mathbf{T}^{\mathrm{T}}\mathbf{B}^{-1}\mathbf{T}+2b\right)}\frac{\mathrm{d}\mathbf{T}^{\mathrm{T}}\mathbf{B}^{-1}\mathbf{T}}{\mathrm{d}\ln\alpha}+c-d\alpha\\ &= \frac{1}{2}\left(M+N-\alpha\sum_{j=1}^{M+N}\boldsymbol{\Sigma}_{jj}\right)-\frac{(MN+2a)\|\boldsymbol{\mu}\|^2\alpha}{2\left[\mathbf{T}^{\mathrm{T}}(\mathbf{T}-\boldsymbol{\Phi}\boldsymbol{\mu})+2b\right]}+c-d\alpha\end{aligned} \tag{21}$$

Setting (21) to zero, followed by algebra operations, yield

$$\alpha = \frac{M+N+2c}{\sum_{j=1}^{M+N}\boldsymbol{\Sigma}_{jj}+2d+(MN+2a)\|\boldsymbol{\mu}\|^2/\left[\mathbf{T}^{\mathrm{T}}(\mathbf{T}-\boldsymbol{\Phi}\boldsymbol{\mu})+2b\right]} \tag{22}$$

The algorithm consists of (13), (14) and (22) with iteration for $\alpha, \boldsymbol{\Sigma}$ and $\boldsymbol{\mu}$.

2.4. Making Predictions

Assume α_{MP} and $\alpha_{0_{MP}}$ are maximizing values obtained by maximizing $p(\alpha\,|\,\mathbf{T},\mathbf{X})$ (Sec. 2.3) and $p(\alpha_0\,|\,\mathbf{T},\mathbf{X})$, respectively. Assume

$$p(\alpha_0,\alpha\,|\,\mathbf{X},\mathbf{T}) \approx \delta\left(\alpha_0-\alpha_{0_{MP}}\right)\delta(\alpha-\alpha_{MP})$$

then

$$\begin{aligned}p(\boldsymbol{t}\,|\,\boldsymbol{x},\mathbf{X},\mathbf{T}) &= \int p(\boldsymbol{t}\,|\,\boldsymbol{x},\boldsymbol{w},\alpha_0,\alpha)\,p(\boldsymbol{w},\alpha_0,\alpha\,|\,\mathbf{X},\mathbf{T})\,\mathrm{d}\boldsymbol{w}\mathrm{d}\alpha_0\mathrm{d}\alpha\\ &= \int p(\boldsymbol{t}\,|\,\boldsymbol{x},\boldsymbol{w},\alpha_0)\,p(\boldsymbol{w}\,|\,\alpha_0,\alpha,\mathbf{X},\mathbf{T})\,p(\alpha_0,\alpha\,|\,\mathbf{X},\mathbf{T})\,\mathrm{d}\boldsymbol{w}\mathrm{d}\alpha_0\mathrm{d}\alpha\\ &\approx \int p(\boldsymbol{t}\,|\,\boldsymbol{x},\boldsymbol{w},\alpha_0)\,p(\boldsymbol{w}\,|\,\alpha_0,\alpha,\mathbf{X},\mathbf{T})\,\delta\left(\alpha_0-\alpha_{0_{MP}}\right)\delta(\alpha-\alpha_{MP})\,\mathrm{d}\boldsymbol{w}\mathrm{d}\alpha_0\mathrm{d}\alpha\\ &= \int p\left(\boldsymbol{t}\,|\,\boldsymbol{x},\boldsymbol{w},\alpha_{0_{MP}}\right)p\left(\boldsymbol{w}\,|\,\alpha_{0_{MP}}\alpha_{MP},\mathbf{X},\mathbf{T}\right)\mathrm{d}\boldsymbol{w}\\ &= N\left(\boldsymbol{t}\,|\,\mathbf{y}(\boldsymbol{x};\boldsymbol{\mu}),\left(\alpha_{0_{MP}}\right)^{-1}\boldsymbol{\Omega}\right)\end{aligned} \tag{23}$$

with

$$\mathbf{y}(\boldsymbol{x};\boldsymbol{\mu}) = \boldsymbol{\Psi}(\boldsymbol{x})\boldsymbol{\mu} \tag{24}$$

$$\boldsymbol{\Omega} = I_M + \boldsymbol{\Psi}(x)\boldsymbol{\Sigma}\boldsymbol{\Psi}(x)^{\mathrm{T}} \tag{25}$$

3. Applications

In examples we employ a radial-basis-function kernel $K(x, x_i) = \exp\left(-\|x - x_i\|^2 / r^2\right)$, and just parameters *a*, *b*, *c* and *d* by training and testing on given training data, finally we take $a = b = c = d = 0.05$ for all examples in this section. In all figures the horizontal axis is the index of samples and the vertical axis is output.

3.1. Regression: Function Approximation

The model can be used to establish the relation between independent variables and dependent variables of a function.

Example 1 2-dimensional vector function with two variables

$$t_1 = \operatorname{sinc}\left(\frac{x_1 + x_2}{4}\right)$$

$$t_2 = -0.5\operatorname{sinc}\left(\frac{x_1 + x_2}{4}\right)\sin\left(\frac{x_1 x_2}{20}\right) - 0.4$$

in domain $\{(x_1, x_2) \mid -10 \le x_1 \le 10, 0 \le x_2 \le 20\}$, where $\operatorname{sinc}(x) = \sin(x)/x$.

Figure 1 and Figure 2 illustrate the results. Figure 1 is learning from 100 noise-free training samples. Figure 2 is based on 100 noisy training samples. The noise is generated from zero-mean Gaussian with 5% of average training data $\|\boldsymbol{t}\|$ as standard deviation. Both test on 100 examples that are not in training data.

Example 2 3-dimensional vector function with 200 variables $(x_1, x_2, \cdots, x_{200}) \to (t_1, t_2, t_3)$.

$$t_1 = \sum_{k=1}^{200} \sin\left((x_k)^{5/7}\right) + \frac{x_{50}}{100}$$

$$t_2 = \frac{x_{200}}{800} t_1 + \frac{x_{50}}{200} + \cos\left(\frac{x_{100}}{5}\right) - 10$$

$$t_3 = \operatorname{atan}\left(\frac{t_1 + t_2}{6}\right) + \frac{t_2 - t_1}{2} - 10$$

We choose samples at point $\boldsymbol{x}^n = \left(x_1^n, x_2^n, \cdots, x_{200}^n\right)$ with $x_k^n = k + (n-1)\pi/4$. 100 samples at points $\boldsymbol{x}^n$ with $n = 1, 3, 5, \cdots, 199$ used as training data, and 100 samples at points $\boldsymbol{x}^n$ with $n = 2, 4, 6, \cdots, 200$ used as testing data.

Figure 3 is from noise-free training samples. Figure 4 is based on noisy training samples. The noise is generated from zero-mean Gaussian with 5% of average training data $\|\boldsymbol{t}\|$ as standard deviation.

3.2. Regression: Inverse Scattering

The model can be used to characterize the connection between measured vector

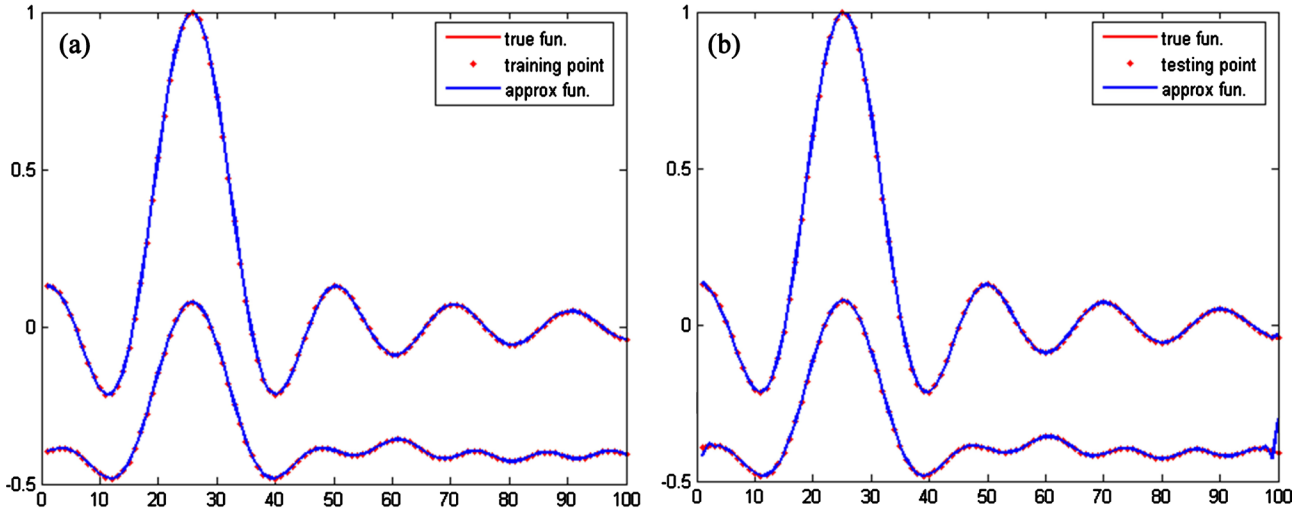

Figure 1. Results for 2-dim vector function with noise-free data: (a) predict on training points; (b) predict on testing points.

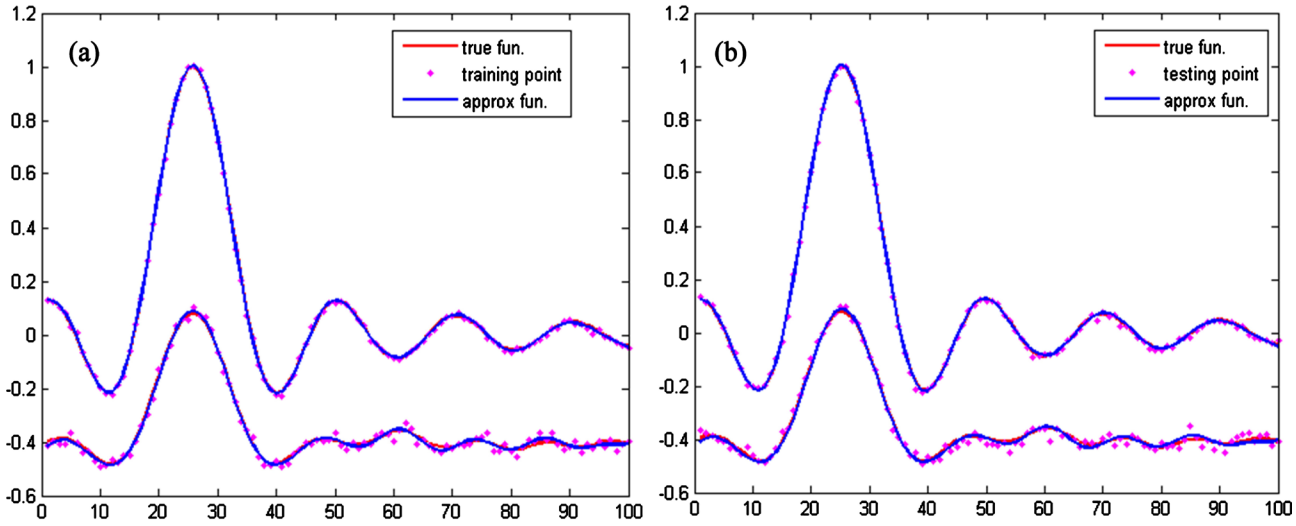

Figure 2. Results for 2-dim vector function with noisy data: (a) predict on training points; (b) predict on testing points.

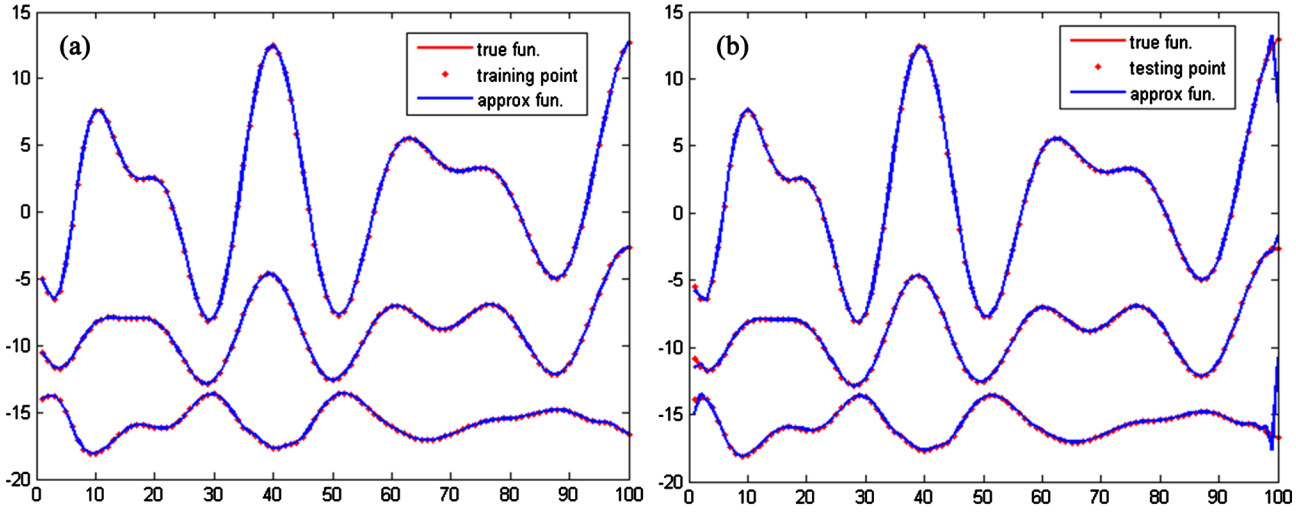

Figure 3. Results for 3-dim vector function with noise-free data: (a) predict on training points; (b) predict on testing points.

scattered-field data $\boldsymbol{x}$ and the underlying target responsible for these fields, characterized by the parameter vector $\boldsymbol{t}$. The scattering data $\boldsymbol{x}$ may be measured at multiple positions. In the examples the measure data is simulated by forward model.

We consider a homogeneous lossless dielectric target buried in a lossy dielectric half space. The objective is to invert for the parameters of the target. In the examples, the parameter vector $\boldsymbol{t}$ is composed of three real numbers: the depth of target, the size of target, and the dielectric constant of target. For each target there are 100 simulated measure data. Training data $\boldsymbol{D} = \{\boldsymbol{x}_n, \boldsymbol{t}_n\}_{n=1}^{N}$ is composed of $N = 180$ examples and testing data is composed of 125 examples that are not in $\boldsymbol{D}$.

Example 1 We consider cube target in this example. Figure 5 and figure 6 illustrate the results. Figure 5 is from noise-free data. Figure 6 is based on noisy data. The noise is generated from zero-mean Gaussian with 10% of average training data $\|\boldsymbol{x}\|$ as standard deviation. The "size" is the width of cube.

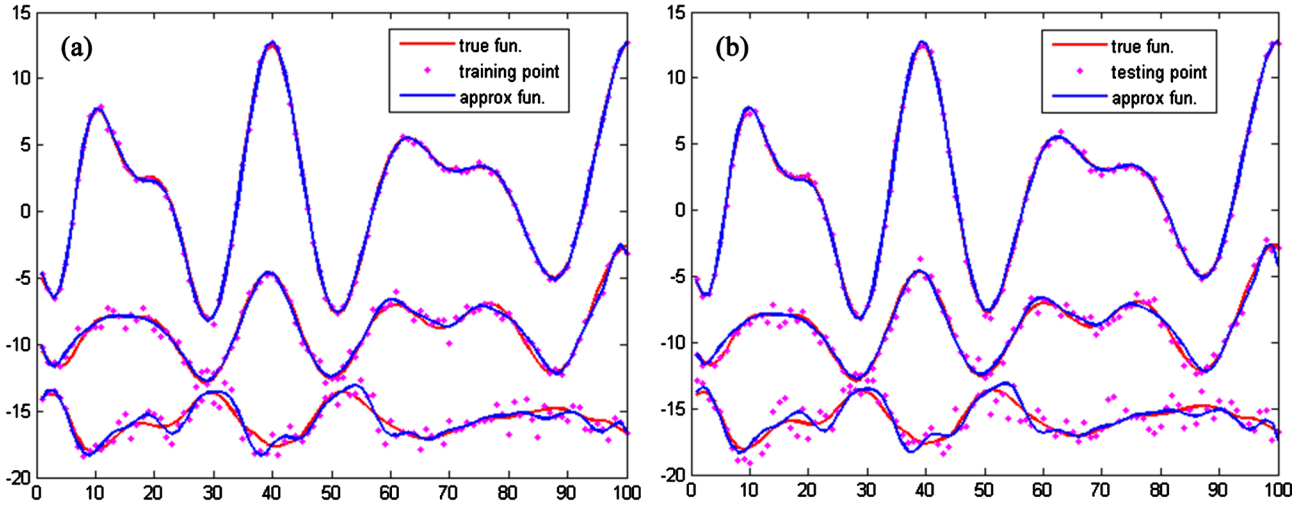

Figure 4. Results for 3-dim vector function with noisy data: (a) predict on training points; (b) predict on testing points.

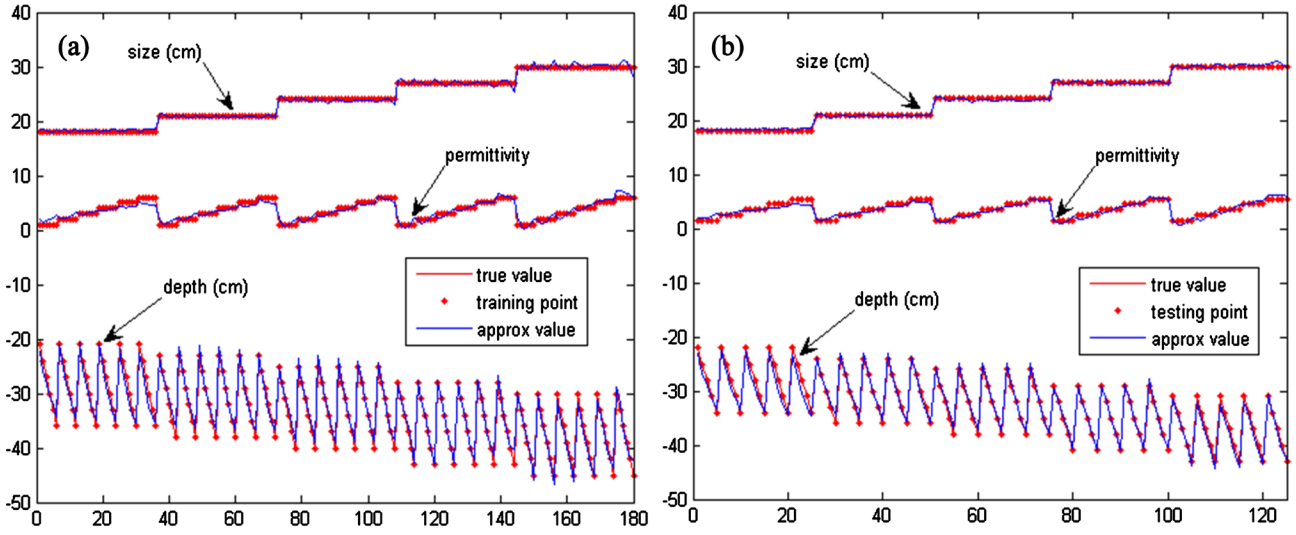

Figure 5. Results for cube target with noise-free data: (a) predict on training points; (b) predict on testing points.

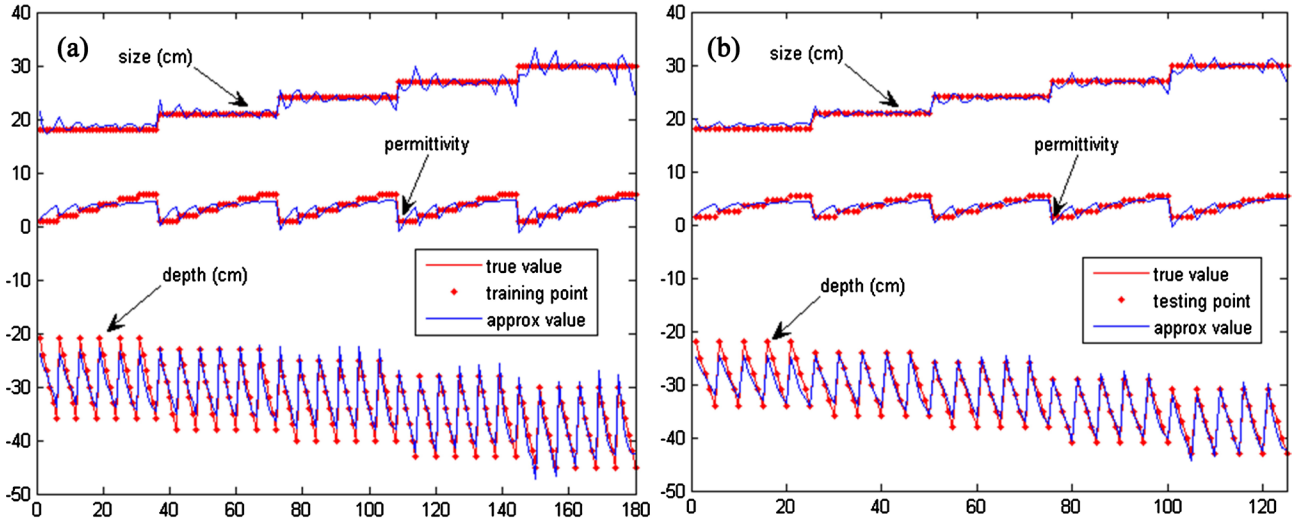

Figure 6. Results for cube target with noisy data: (a) predict on training points; (b) predict on testing points.

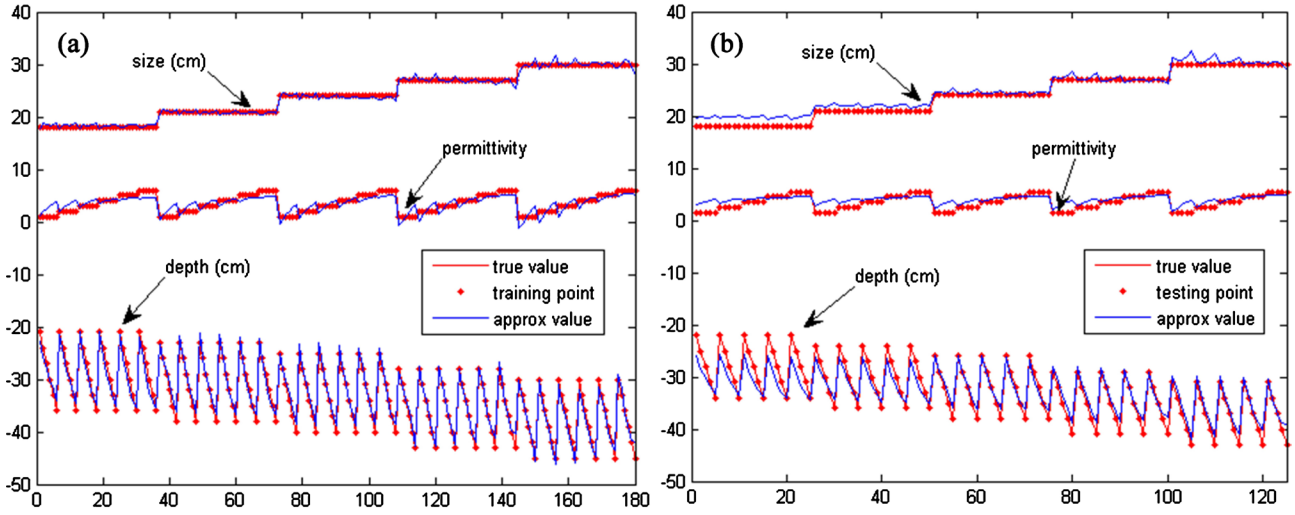

Figure 7. Results for sphere target with noise-free data: (a) predict on training points; (b) predict on testing points.

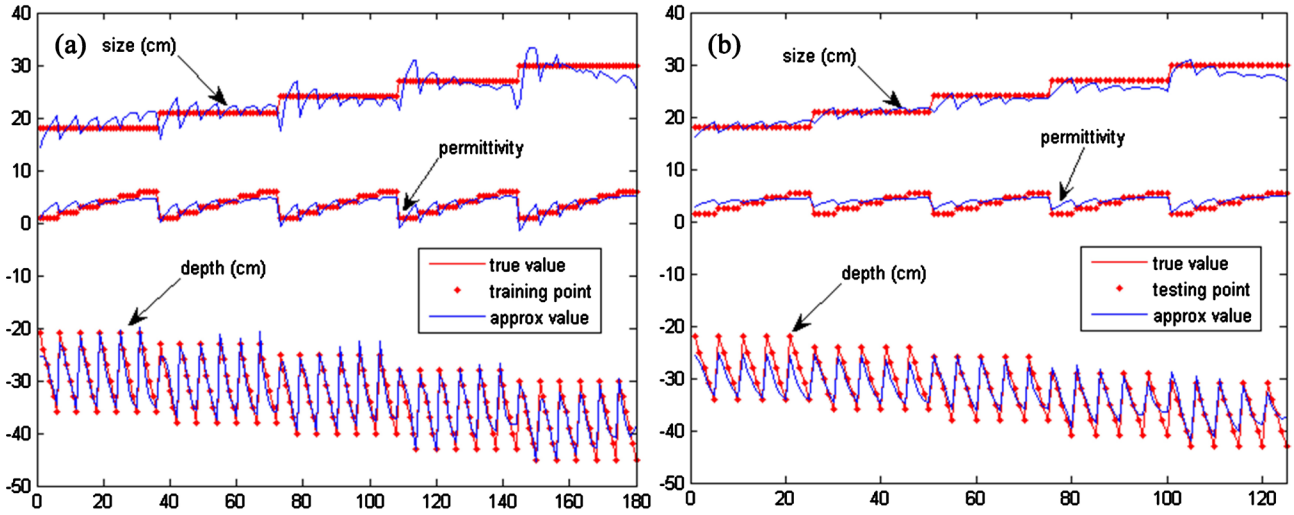

Figure 8. Results for sphere target with noisy data: (a) predict on training points; (b) predict on testing points.

Example 2 We consider sphere target in this example. Figure 7 and figure 8 illustrate the results. Figure 7 is from noise-free data. Figure 8 is based on noisy data. The noise is generated from zero-mean Gaussian with 10% of average training data $\|x\|$ as standard deviation. The "size" is the diameter of sphere.

We applied the model to two completely different types of problems, the model works well for both application. The results display this regression model can apply to various types of regression problems.

4. Conclusion

A Bayesian vector-regression algorithm has been developed. The model employs a statistical prior that favors a sparse model, for which most of its weights are zero [5]. This model improves the algorithm in [9], and reduces the number of hyperparameters, which need to be calculated in the algorithm, from two to one. The model is not established for one specific problem, and so can be applied to different regression problems. We have discussed the theoretical development of the model and have presented several example results for two different applications. One is for function approximation, and the other is for inverse scattering of dielectric targets buried in a lossy half space. It has been demonstrated that the algorithm works well for different applications.

Conflicts of Interest

The author declares no conflicts of interest regarding the publication of this paper.

References

[1] Law, T. and Shawe-Taylor, J. (2017) Practical Bayesian Support Vector Regression for Financial Time Series Prediction and Market Condition Change Detection. *Quantitative Finance*, **17**, 1403-1416. https://doi.org/10.1080/14697688.2016.1267868

[2] Yu, J. (2012) A Bayesian Inference Based Two-Stage Support Vector Regression Framework for Soft Sensor Development in Batch Bioprocesses. *Computers & Chemical Engineering*, **41**, 134-144. https://doi.org/10.1016/j.compchemeng.2012.03.004

[3] Jacobs, J.P. (2012) Bayesian Support Vector Regression with Automatic Relevance Determination Kernel for Modeling of Antenna Input Characteristics. *IEEE Transactions on Antennas and Propagation*, **60**, 2114-2118. https://doi.org/10.1109/TAP.2012.2186252

[4] Hans, C. (2009) Bayesian Lasso Regression. *Biometrika*, **96**, 835-845. https://doi.org/10.1093/biomet/asp047

[5] Tipping, M.E. (2001) Sparse Bayesian Learning and the Relevance Vector Machine. *Journal of Machine Learning Research*, **1**, 211-244.

[6] Scholkopf, B. and Smola, A.J. (2001) Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge.

[7] Berger, J.O. (1985) Statistical Decision Theory and Bayesian Analysis. 2nd Edition, Springer, Berlin. https://doi.org/10.1007/978-1-4757-4286-2

[8] Mardia, K.V., Kent, J.T. and Bibby, J.B. (1979) Multivariate Analysis. Academic Press, New York.

[9] Yu, Y., Krishnapuram, B. and Carin, L. (2004) Inverse Scattering with Sparse Bayesian Vector Regression. *Inverse Problems*, **20**, 217-231. https://doi.org/10.1088/0266-5611/20/6/S13