mathematics

MDPI

Article

Multi-Output Bayesian Support Vector Regression Considering Dependent Outputs

Yanlin Wang, Zhijun Cheng * and Zichen Wang

College of Systems Engineering, National University of Defense Technology, Changsha 410073, China; wangyanlin@nudt.edu.cn (Y.W.); wangzichen18@nudt.edu.cn (Z.W.)
* Correspondence: chengzhijun@nudt.edu.cn

Abstract: Multi-output regression aims to utilize the correlation between outputs to achieve information transfer between dependent outputs, thus improving the accuracy of predictive models. Although the Bayesian support vector machine (BSVR) can provide both the mean and the predicted variance distribution of the data to be labeled, which has a large potential application value, its standard form is unable to handle multiple outputs at the same time. To solve this problem, this paper proposes a multi-output Bayesian support vector machine model (MBSVR), which uses a covariance matrix to describe the relationship between outputs and outputs and outputs and inputs simultaneously by introducing a semiparametric latent factor model (SLFM) in BSVR, realizing knowledge transfer between outputs and improving the accuracy of the model. MBSVR integrates and optimizes the parameters in BSVR and those in SLFM through Bayesian derivation to effectively deal with the multi-output problem on the basis of inheriting the advantages of BSVR. The effectiveness of the method is verified using two function cases and four high-dimensional real-world data with multi-output.

Keywords: multiple dependent outputs; support vector regression; Bayesian inference; semiparametric latent factor model; multi-output Bayesian support vector regression (MBSVR)

MSC: 62J02; 68Txx

check for updates

Citation: Wang, Y.; Cheng, Z.; Wang, Z. Multi-Output Bayesian Support Vector Regression Considering Dependent Outputs. *Mathematics* **2024**, *12*, 2923. https://doi.org/10.3390/math12182923

Academic Editor: Manuel Alberto M. Ferreira

Received: 24 August 2024
Revised: 17 September 2024
Accepted: 18 September 2024
Published: 20 September 2024

1. Introduction

Regression models, also known as response models, can accurately predict the output of other features by establishing a mapping relationship between data features and outputs [1]. The commonly used regression models are single-output models, i.e., models with one or more inputs but only one output. However, in real engineering problems, there are often multiple outputs. For example, in novel battery material discovery, simultaneous and comprehensive prediction of the multidimensional properties of battery electrode materials is needed to help accelerate material discovery and design [2]. In environmental forecasting, there is a need to simultaneously predict particulate matter concentrations at different air quality monitoring stations, which often have potentially nonlinear spatial correlations. Reliable and accurate predictions help in crisis response and can reduce health risks [3]. Ultra-high-performance fiber-reinforced concrete (UHPFRC) is used in a variety of civil engineering applications, and its structural behavior is closer to that of steel. To investigate the effect of component dosage on its strain and energy absorption capacity under peak tension and to optimize the material dosage, both outputs need to be predicted simultaneously [4]. Multiple outputs can be processed separately, but this method ignores the potential correlation between the outputs and results in information loss. Therefore, the correlation between the outputs can be used to build a multi-output model, which is also known as a multi-response or multi-task model.

Support vector machine (SVR) was first proposed by Vapnik based on the principle of structural risk minimization [5]. SVR uses quadratic programming to obtain predictions for a single output. Compared to other models, SVR has superior performance due to its structural risk minimization principle, which allows it to avoid overfitting and achieve better output approximation [6,7]. The Bayesian support vector machine (BSVR) introduces Gaussian process assumptions and Bayesian inference on the basis of SVR to obtain the predicted values and their distributions. The BSVR model not only obtains an estimate of the unknown sample points but also has the advantages of adaptivity and prediction error distributions of Bayesian methods [8]. Meanwhile, SVR has shown superior performance in dealing with nonlinear problems and avoiding overfitting with good generalization ability [9]. Therefore, Bayesian support vector machines have received a lot of attention in the past time, such as [9–11] and references therein.

Multiple output regression aims to establish a mapping from multivariate inputs to multiple outputs [12]. Despite the potential utility of BSVR, its standard form cannot handle multiple outputs. The simplest way to deal with the multiple-output problem is to model multiple outputs individually. For each output indicator, a model can be built independently. This treatment is simpler but does not take into account the correlation between outputs and is suitable for scenarios where no correlation exists between outputs [13]. Another method is chain modeling [14]. This method predicts an output, followed by predicting the next output using the predicted output as input, and so on. However, the use of chain modeling requires determining the order of the outputs and the dependencies between them.

Considering that multiple outputs are correlated and being modeled individually can lead to information loss, more and more multi-output modeling approaches have been proposed. Multi-output modeling takes advantage of the correlation between outputs so that a single output can utilize information from other outputs to obtain more accurate predictions [13]. Methods have been developed to extend the support vector machine model so that it can handle multiple outputs simultaneously. Pérez-Cruz et al. [14] transformed the pipeline in a pipeline-based model $\varepsilon - SVR$ into a hyper-square pipeline by equalizing the output values of the data points located outside the pipeline. This hyperspherical insensitive zone is designed to be more effective than modeling it individually. Zhang et al. [15] proposed an extended LSSVR (ELS-SVR), which extends the original feature space using vector virtualization to represent the multi-output case as an equivalent single-output case in the extended feature space and solved using a least-squares support vector machine. Inspired by multi-task learning, Xu et al. [16] changed the weight vector of a least-squares support vector machine from one to two. One carries generic information and the other carries specific information, thus characterizing the correlation between the two outputs, which is referred to as multi-output LS-SVR (MLS-SVR). Literature [17] gives an overview of the correlation methods and also analyzes the disadvantages of the above methods: the hyperspherical ϵ-tube of M-SVR does not exhibit an advantage over a hypercubic one, ELSSVR cannot handle the negative correlations, MLSSVR does not handle well the case of only partial correlations, and the above methods do not consistently outperform single-output support vector machines.

The above methods distribute modifications to $\varepsilon - SVR$ and LSSVR so that they can solve the multi-output problem. Among them, the support vectors in $\varepsilon - SVR$ are sparse and only some of the samples are involved in the model construction. LSSVR transforms convex quadratic optimization problems into linear systems of equations problems, in which all the samples are involved in the model construction. These methods better utilize the correlation between the outputs and improve the model accuracy to some extent. However, these methods cannot obtain a prediction distribution similar to BSVR, which can quantify uncertainty and has good application prospects. In addition, BSVR is based on Bayesian theory, which can systematically and effectively infer the optimal hyperparameters [8]. In terms of describing the correlation of multiple outputs, the method based on $\varepsilon - SVR$ does not have an accurate structure to describe the correlation of outputs. ELS-SVR only describes the correlation through a parameter greater than 0, so it cannot describe the negative correlation. MLS-SVR describes the shared information through the disassembly of the weight vector. However, as the weight describes the correlation of multiple outputs in a unified manner, it cannot describe the partial correlation of the outputs.

Therefore, this paper introduces the multi-output Gaussian process assumption based on the Bayesian support vector machine model (BSVR) while considering the variability of multiple outputs in terms of SVR trade-off parameters. A Bayesian framework is used to systematically and comprehensively optimize the original BSVR hyperparameters and the hyperparameters of the Gaussian process, which in turn provides the predicted values and probability distributions of multiple outputs. The difference between the multi-output Bayesian support vector machine (MBSVR) and single-output Bayesian support vector machine (BSVR) mainly lies in the kernel function. MBSVR uses the semiparametric latent factor model in the new kernel function, which describes the relationship between the inputs and outputs, and the outputs and outputs at the same time through the linear combinations of implicit functions so that information between them can be transferred to improve the accuracy of the model. The main contributions of our work are as listed:

- The method inherits the advantages of support vector machines in nonlinear, high-dimensional problems by introducing Bayesian derivation in support vector machines.
- Compared to other SVR-based multi-output regression methods. Based on Bayesian theory, the predicted mean and its probability distribution (uncertainty) can be obtained, and the hyperparameter optimization can be performed systematically and effectively.
- Compared with BSVR, the method combines the SLFM structure with BSVR for comprehensive optimization of parameters through information transfer between outputs and uses the shared information to improve model accuracy.
- The use of a trade-off parameter makes the method sensitive to outliers and allows for more robust performance on real datasets than multi-output Gaussian process.

The rest of the paper is structured as follows: in Section 2, the Bayesian support vector machine model and the semiparametric latent factor model are introduced. In Section 3, the new multi-output Bayesian support vector machine model is introduced. In Section 4, the model evaluation is carried out using function arithmetic and real datasets, and Section 5 concludes.

2. Related Description and Basic Theories

2.1. Single-Output Bayesian Support Vector Machine

Single-output Bayesian support vector machine (BSVR) introduces Gaussian process assumptions based on SVR to optimize the hyperparameters by Bayesian derivation [8]. BSVR is widely used because it can obtain the mean and the prediction variance of the prediction through rigorous derivation to measure the uncertainty of prediction while maintaining the advantages of SVR. In BSVR, the mapping relationship between outputs and factors can be expressed as [8]:

$$y_j = g(\mathbf{x}_j) + \delta_j \tag{1}$$

where δ_j is an independently and identically distributed random error. $g(\mathbf{x}_j)$ is the support vector machine regression model. It is a zero-mean Gaussian stochastic process whose covariance between the two outputs of different $\mathbf{x}$ can be expressed as:

$$k(\widetilde{g}(\mathbf{x}), \widetilde{g}(\mathbf{x'}) = k(\mathbf{x}, \mathbf{x'}) = \prod_{j=1}^{n} \exp(-\theta_j (\mathbf{x}_j - \mathbf{x}_j^{'})^2) \tag{2}$$

where $\theta = (\theta_1, \theta_2 \cdots \theta_n)$ is the hyperparameter to be adjusted and the covariance between the two outputs is equal to the value of the corresponding kernel function. Under the assumptions of the Gaussian process, the a priori of the outputs $\mathbf{G} = \{\tilde{g}(\mathbf{x_1}), \tilde{g}(\mathbf{x_2}), \cdots, \tilde{g}(\mathbf{x_N})\}^T$ can be described as:

$$p(\mathbf{G}|\boldsymbol{\gamma}) = \frac{1}{Z_G}\exp(-\frac{1}{2}(\mathbf{G}-\mathbf{b})^T\mathbf{K}^{-1}(\mathbf{G}-\mathbf{b})) \tag{3}$$

$$Z_G = (2\pi)^{N/2}\left|\mathbf{K}\right|^{1/2} \tag{4}$$

where $\mathbf{K} \in \mathbb{R}^{N\times N}$ is the input covariance matrix in the sample where training was performed, $\mathbf{b} = [b, b, \cdots, b] \in \mathbb{R}^N$. $\boldsymbol{\gamma}$ denotes all hyperparameters. In this problem, the hyperparameters include the hyperparameters $\boldsymbol{\theta} = (\theta_1, \theta_2, \cdots, \theta_n)$ in the kernel function and the trade-off parameters C. Since the noise is assumed to be an independent and identically distributed random variable, the likelihood function of the sample output for the training set of samples can be expressed as [8]:

$$p(\mathbf{Y}|\mathbf{G},\boldsymbol{\gamma}) = \prod_{j=1}^{N} p(y_j - g(\mathbf{x}_j)|\mathbf{G},\boldsymbol{\gamma}) = \prod_{j=1}^{N} p(\delta_j) \tag{5}$$

where $p(\delta_j)$ is the probability distribution of δ_j with the expression:

$$p(\delta) = \frac{1}{Z_\delta}\exp(-Cl(\delta)) \tag{6}$$

$$Z_\delta = \int \exp(-Cl(\delta))d\delta = \sqrt{2\pi/C} \tag{7}$$

where $l(\delta)$ is the loss function of the model and C is a trade-off constant. According to Bayesian theory, the posterior distribution is obtained by synthesizing the information of the existing samples with the prior distribution. In BSVR, the likelihood function characterizes the training set sample information, while $p(\mathbf{G}|\boldsymbol{\gamma})$ is the prior distribution of the samples. Since the prior distribution satisfies the Gaussian process assumption, it can be represented by Equation (3). In summary, the posterior distribution of satisfies [18].

$$p(\mathbf{G}|\mathbf{Y},\boldsymbol{\gamma}) = \frac{p(\mathbf{Y}|\mathbf{G},\boldsymbol{\gamma})p(\mathbf{G}|\boldsymbol{\gamma})}{p(\mathbf{Y}|\boldsymbol{\gamma})} \tag{8}$$

$p(\mathbf{Y}|\boldsymbol{\gamma})$ is a normalization constant. Bringing (3) and (5) into (8) yields:

$$p(\mathbf{G}|\mathbf{Y},\boldsymbol{\gamma}) = \frac{1}{Z}\exp(-C\sum_{i=1}^{N} l(y_i - \tilde{g}(\mathbf{x})) - \frac{1}{2}(\mathbf{G}-\mathbf{b})^T\mathbf{K}^{-1}(\mathbf{G}-\mathbf{b})) \tag{9}$$

$$Z = \int \exp(-S(\mathbf{G}))d\mathbf{G} \tag{10}$$

$$S(\mathbf{G}) = \sum_{j=1}^{N} Cl(y_j - \tilde{g}(\mathbf{x}_j)) + \frac{1}{2}(\mathbf{G}-\mathbf{b})^T\mathbf{K}^{-1}(\mathbf{G}-\mathbf{b}) \tag{11}$$

Thus, according to the principle of the great likelihood method of solution, maximizing the posterior distribution in (11) can be equated to:

$$\min_G \sum_{i=1}^{N} Cl(y_i - \tilde{g}(\mathbf{x_i})) + \frac{1}{2}(\mathbf{G}-\mathbf{b})^T\mathbf{K}^{-1}(\mathbf{G}-\mathbf{b}) \tag{12}$$

where C is the equilibrium parameter. The loss function of a support vector machine can be represented in a variety of ways, and one is the squared loss function:

$$l(\delta) = \frac{1}{2}\delta^2 \tag{13}$$

Minimizing the squared loss function is essentially equivalent to great likelihood estimation under the assumption that the error follows a Gaussian distribution. Bringing the loss function expression into (12) yields the new objective function as:

$$\min_{\mathbf{G}} \sum_{j=1}^{N} \frac{C}{2} e_j^2 + \frac{1}{2}(\mathbf{G}-\mathbf{b})^T \mathbf{K}^{-1} (\mathbf{G}-\mathbf{b}) \tag{14}$$

where $y_j = g(x_j) + e_j$. Solving yields an estimate of $\mathbf{G}$ as:

$$\hat{\mathbf{G}} = \mathbf{K}(\mathbf{K}+\mathbf{I}/C)^{-1}\mathbf{Y} = \mathbf{K}\boldsymbol{\beta} + \mathbf{b} \tag{15}$$

where $\mathbf{I}$ is the unit matrix, $\boldsymbol{\beta} = [\beta_1, \beta_2, \cdots \beta_N] = (\mathbf{K}+\mathbf{I}/C)^{-1}\mathbf{Y}$.

For the output $g(\mathbf{x})$ to be predicted, its joint distribution with the training set satisfies:

$$\begin{pmatrix} g(\mathbf{x}) \\ \mathbf{G} \end{pmatrix} \sim N\left(\begin{Bmatrix} \mathbf{0} \\ 0 \end{Bmatrix}, \begin{Bmatrix} k(\mathbf{x},\mathbf{x}) & \mathbf{k}(\mathbf{x},\mathbf{X}) \\ \mathbf{k}(\mathbf{X},\mathbf{x}) & \mathbf{k}(\mathbf{X},\mathbf{X}) \end{Bmatrix} \right) \tag{16}$$

$$\mathbf{k}(\mathbf{X},\mathbf{x}) = \mathbf{k}(\mathbf{x},\mathbf{X})^T = [k(\mathbf{x_1},\mathbf{x}), \cdots, k(\mathbf{x_N},\mathbf{x})]^T \tag{17}$$

(17) denotes the variance between $\mathbf{G}$ and $g(\mathbf{x})$, $k(\mathbf{X},\mathbf{X}) = \mathbf{K}$. The prior of $g(\mathbf{x})$ still obeys Gaussian distribution:

$$\begin{aligned} p(g(\mathbf{x})|\mathbf{Y}) &= \int p(\tilde{g}(\mathbf{x})|\mathbf{Y}) p(\mathbf{G}|\mathbf{Y}) d\mathbf{G} \\ &= N(\mu(\mathbf{x}), \Sigma^2(\mathbf{x})) \end{aligned} \tag{18}$$

$$\begin{aligned} \mu(x) &= k(\mathbf{x},\mathbf{X}) k(\mathbf{X},\mathbf{X})^{-1} \hat{\mathbf{G}} \\ &= k(\mathbf{x},\mathbf{X})(\mathbf{K}+\mathbf{I}/C)^{-1}\mathbf{Y} \\ &= \sum_{j=1}^{N} \beta_j k(\mathbf{x},\mathbf{x_j}) \end{aligned} \tag{19}$$

$$\Sigma^2(x) = k(\mathbf{x},\mathbf{x}) - \mathbf{k}(\mathbf{x},\mathbf{X})\mathbf{k}(\mathbf{X},\mathbf{X})^{-1}\mathbf{k}(\mathbf{X},\mathbf{x}) \tag{20}$$

For the Bayesian support vector machine with a squared loss function, the parameters to be optimized include θ and C, and the optimal values of these hyperparameters are determined by the maximum posteriori probability. The specific formula derivation and calculation methods are described in [10].

2.2. Semiparametric Latent Factor Model (SLFM)

The most used multi-output covariance structure in multi-output Gaussian processes is the linear model of coregionalization (LMC) [19]. The semiparametric latent factor model (SLFM) is a special form of LMC. The model assumes that there are Q shared potential Gaussian processes, and generally, the number of potential Gaussian processes is smaller than the number of outputs. SLFM represents the output as a linear combination of Q Gaussian processes. Taking two outputs as an example, two Gaussian implicit functions are chosen, $N_f = Q = 2$. Assuming that $u_1(\mathbf{x})$ and $u_2(\mathbf{x})$ are obtained by sampling from the two Gaussian processes respectively, where $u_1(\mathbf{x}) \sim \mathcal{GP}(0, k_1(\mathbf{x},\mathbf{x}'))$, $u_2(\mathbf{x}) \sim \mathcal{GP}(0, k_2(\mathbf{x},\mathbf{x}'))$, $g_1(\mathbf{x})$ and $g_2(\mathbf{x})$ are obtained by linearly transforming $u_1(\mathbf{x})$ and $u_2(\mathbf{x})$:

$$g_1(\mathbf{x}) = a_{1,1}u_1(\mathbf{x}) + a_{2,2}u_2(\mathbf{x}) \tag{21}$$

$$g_2(\mathbf{x}) = a_{2,1}u_1(\mathbf{x}) + a_{2,2}u_2(\mathbf{x}) \tag{22}$$

where $u_1(\mathbf{x})$ and $u_2(\mathbf{x})$ obey a Gaussian process distribution with different covariance functions and are independent of each other, i.e., in the case of $q \neq q'$, $u_q(\mathbf{x}) \perp u_{q'}(\mathbf{x}')$,

$\mathrm{cov}(u_q(\mathbf{x}), u_{q'}(\mathbf{x}')) = 0$. Since a new covariance function can be obtained by linearly combining several covariance functions. $g_1(\mathbf{x})$ and $g_2(\mathbf{x})$ can be written as:

$$\mathbf{g}(\mathbf{x}) = \mathbf{a}_1 u_1(\mathbf{x}) + \mathbf{a}_2 u_2(\mathbf{x}) \tag{23}$$

where $\mathbf{g}(\mathbf{x}) = \begin{bmatrix} g_1(\mathbf{x}) & g_2(\mathbf{x}) \end{bmatrix}^{\mathrm{T}}$, $\mathbf{a}_1 = \begin{bmatrix} a_{1,1} & a_{2,1} \end{bmatrix}^{\mathrm{T}}$, $\mathbf{a}_2 = \begin{bmatrix} a_{1,2} & a_{2,2} \end{bmatrix}^{\mathrm{T}}$, the variances of $\mathbf{g}(\mathbf{x})$ and $\mathbf{g}(\mathbf{x}')$ can be calculated:

$$\begin{aligned} \mathbf{k}_M(\mathbf{x}, \mathbf{x}') &= \mathrm{cov}(\mathbf{g}(\mathbf{x}), \mathbf{g}(\mathbf{x}^{'})) \\ &= \mathbf{a}_1(\mathbf{a}_1)^{\mathrm{T}} \mathrm{cov}(u_1(\mathbf{x}), u_1(\mathbf{x}')) + \mathbf{a}_2(\mathbf{a}_2)^{\mathrm{T}} \mathrm{cov}(u_2(\mathbf{x}), u_2(\mathbf{x}')) \\ &= \mathbf{a}_1(\mathbf{a}_1)^{\mathrm{T}} \mathbf{k}_1(\mathbf{x}, \mathbf{x}') + \mathbf{a}_2(\mathbf{a}_2)^{\mathrm{T}} \mathbf{k}_2(\mathbf{x}, \mathbf{x}') \end{aligned} \tag{24}$$

Defining $\mathbf{B}_1 = \mathbf{a}_1(\mathbf{a}_1)^{\mathrm{T}}$, $\mathbf{B}_2 = \mathbf{a}_2(\mathbf{a}_2)^{\mathrm{T}}$, we can obtain

$$k_M(\mathbf{x}, \mathbf{x}') = \mathbf{B}_1 \mathbf{k}_1(\mathbf{x}, \mathbf{x}') + \mathbf{B}_2 \mathbf{k}_2(\mathbf{x}, \mathbf{x}') \tag{25}$$

$$\begin{bmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \end{bmatrix} = \begin{bmatrix} g_1(\mathbf{x}_1) \\ \vdots \\ g_1(\mathbf{x}_N) \\ g_2(\mathbf{x}_1) \\ \vdots \\ g_2(\mathbf{x}_N) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \mathbf{B}_1 \otimes \mathbf{K}_1 + \mathbf{B}_2 \otimes \mathbf{K}_2 \right) \tag{26}$$

Furthermore, consider a plurality of outputs $\{g_i(\mathbf{x})\}_{i=1}^{N_f}$ in a more generalized form:

$$g_i(\mathbf{x}) = \sum_{q=1}^{Q} a_{i,q} u_q(\mathbf{x}) \tag{27}$$

(27) can be described using a matrix as:

$$\mathbf{g}(\mathbf{x}) = \mathbf{A}\mathbf{u}(\mathbf{x}) \tag{28}$$

$$\mathbf{g}(\mathbf{x}) = \left\{ \mathrm{g}_1(\mathbf{x}), g_2(\mathbf{x}), \cdots g_{N_f}(\mathbf{x}) \right\}^{\mathrm{T}} \tag{29}$$

$$\mathbf{u}(\mathbf{x}) = \left[u_1(\mathbf{x}), u_2(\mathbf{x}), \cdots, u_Q(\mathbf{x}) \right] \tag{30}$$

$$\mathbf{A} = \begin{bmatrix} a_{1,1} & \cdots & a_{1,Q} \\ \vdots & \ddots & \vdots \\ a_{N_f,1} & \cdots & a_{N_f,Q} \end{bmatrix} \in \mathbb{R}^{N_f \times Q} \tag{31}$$

Similarly, since multiple Gaussian processes are independent for multiple outputs $\mathbf{g}(\mathbf{x}) = [g_1(\mathbf{x}), \cdots, g_{N_f}(\mathbf{x})]^{\mathrm{T}}$, the covariance of multiple outputs can be expressed as:

$$k_M(\mathbf{x}, \mathbf{x}') = \sum_{q=1}^{Q} \mathbf{A}_q \mathbf{A}_q^{\mathrm{T}} \mathbf{k}_q(\mathbf{x}, \mathbf{x}') \tag{32}$$

$$\begin{bmatrix} \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_{N_f} \end{bmatrix} = \begin{bmatrix} g_1(\mathbf{x}_1) \\ \vdots \\ g_1(\mathbf{x}_N) \\ g_2(\mathbf{x}_1) \\ \vdots \\ g_2(\mathbf{x}_N) \\ \vdots \\ g_{N_f}(\mathbf{x}_1) \\ \vdots \\ g_{N_f}(\mathbf{x}_N) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}, \sum_{q=1}^{Q} \mathbf{A}_q \mathbf{A}_q^{\mathrm{T}} \otimes \mathbf{K}_q \right) \tag{33}$$

where in $\mathbf{A}_q \in \mathbb{R}^{N_f \times N_f}$, the elements corresponding to the i output is $A_{ii'}^{q} = a_{i,q}a_{i',q}$. To further characterize the multi-output Gaussian process, it is necessary to determine the number of implicit functions. It has been found that the larger Q is, the more flexible the model is and the more variability can be described. Some scholars have determined Q to be two or the number of outputs. The increase of Q will also bring about a further increase in computational cost. In order to balance the flexibility and accuracy of the model and the computational overhead, this paper will take the value of Q as N_f. Then, the correlation can be further expressed as:

$$\mathbf{k}_M(\mathbf{x}, \mathbf{x}') = \mathbf{R}diag[\mathbf{k}_1(\mathbf{x}, \mathbf{x}'), \cdots, \mathbf{k}_Q(\mathbf{x}, \mathbf{x}')]\mathbf{R}^T \tag{34}$$

$$\mathbf{A}_q = \mathbf{r}_q {\mathbf{r}_q}^{\mathrm{T}} \tag{35}$$

$$\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, \cdots, \mathbf{r}_Q] \tag{36}$$

$$\mathbf{A} = \begin{bmatrix} a_{1,1} & \cdots & a_{1,Q} \\ \vdots & \ddots & \vdots \\ a_{N_f,1} & \cdots & a_{N_f,Q} \end{bmatrix} = \mathbf{R}\mathbf{R}^{\mathrm{T}} \tag{37}$$

and $Q = N_f$, $\mathbf{k}_q(\mathbf{x}, \mathbf{x}') = \prod_{d=1}^{D} \exp(-\theta_d (x_d - x_d^{'})^2)$. The kernel function of a Gaussian process has a set of parameters $\boldsymbol{\theta} \in \mathbb{R}^D$ to be optimized, D is the sample dimension, then for each of the Q implicit functions, there is a set of $\boldsymbol{\theta}$, which is used to measure the importance of the inputs as equivalent to the specified outputs. Then the parameters of the kernel function to be optimized include $\boldsymbol{\theta}_M = [\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \cdots, \boldsymbol{\theta}_{N_f}] \in \mathbb{R}^{N_f \times N}$. $\sum_0 = \mathbf{A}\mathbf{A}^T \in \mathbb{R}^{N_f \times N_f}$ is used to describe the covariance between multiple outputs. $\mathbf{A}$ is the upper triangular matrix and also a set of unknown hyperparameters to be optimized.

3. Multi-Output Bayesian Support Vector Regression Model

The structure of the MBSVR model is shown in Figure 1, where the left side is the SLFM structure and $g(\mathbf{x})$ combines linear combination of Q implicit functions, according to which the variance between $g(\mathbf{x})$ can be quantitatively described. The right side represents the trade-off parameters in the support vector machine; for each output, there is a corresponding trade-off parameter, which is used to trade off the complexity and error of the model. The *ith* output can be expressed as (27), where $a_{i,q}$ is the parameter to be optimized, $u_q(\mathbf{x})$ is Gaussian process implicit function. Based on the expression of $g_i(\mathbf{x})$, the model can be expressed as:

$$\mathbf{Y}_i = \mathbf{g}(\mathbf{x}_i) + \boldsymbol{\delta}_i \tag{38}$$

where $\mathbf{x}_j$ is the *jth* sample and $\boldsymbol{\delta}$ is an independently and identically distributed random error. C is a number greater than 0, which determines the degree of tolerance for error in the model. When C is large, the model will not allow for errors, the complexity is high, and it may be overfitted with poor generalization ability. When C is small, the model does not focus on the presence of errors, the model is simpler, and it is easy to be underfitted.

MSVR combines the SLFM structure with the support vector machine model through the Bayesian assumptions. Through the Gaussian process assumption and Bayesian derivation, the correlation between the outputs is effectively delineated, and finally, the predicted mean and probability distribution of multiple outputs are obtained.

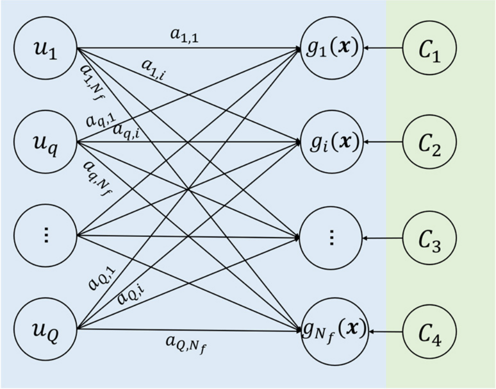


Figure 1. Model structure of MBSVR.

3.1. Bayesian Assumptions for MBSVR

Assume that a multi-output modeling problem consists of N_f outputs and N samples. Define a vector $\mathbf{Y}$, which characterizes the outputs of the sample points and contains $N_f \times N$ elements in the vector. Multi-output Bayesian support vectors aim to approximate the N_f outputs $\{g_i(\mathbf{x})\}_{1 \le i \le N_f}$ simultaneously. A more accurate model is built by considering the correlation between the outputs. In a multi-output Bayesian support vector machine, for a certain $\mathbf{x}_j \in \mathbb{R}^d$, the relationship between the outputs and the factors can be expressed as (38). where $\boldsymbol{\delta}_j \in \mathbb{R}^{N_f}$ is an independently and identically distributed random error whose distribution form is usually unknown. $\mathbf{Y}_j$ is the values of multiple outputs. $\mathbf{g}(\mathbf{x})$ is a support vector machine, which is a multi-output Gaussian process. Since there are multiple outputs, the multiple outputs of all samples $\mathbf{g}(\mathbf{x}) = \left\{ \mathrm{g}_1(\mathbf{x}), g_2(\mathbf{x}), \cdots, g_{N_f}(\mathbf{x}) \right\}^{\mathrm{T}}$ are satisfied:

$$\mathbf{g}(\mathbf{x}) \sim \mathcal{GP}(\mathbf{0}, \sum_0) \tag{39}$$

According to the SLFM principle, $\sum_0 = \mathbf{A}\mathbf{A}^T$ is the parameter to be optimized. N_f Gaussian process outputs can be expressed as:

$$\left\{ \begin{array}{c} \widetilde{g}_1(\mathbf{x}_1), \widetilde{g}_1(\mathbf{x}_2), \cdots, \widetilde{g}_1(\mathbf{x}_N) \\ \widetilde{g}_2(\mathbf{x}_1), \widetilde{g}_2(\mathbf{x}_2), \cdots, \widetilde{g}_2(\mathbf{x}_N) \\ \vdots \\ \widetilde{g}_{N_f}(\mathbf{x}_1), \widetilde{g}_{N_f}(\mathbf{x}_2), \cdots, \widetilde{g}_{N_f}(\mathbf{x}_N) \end{array} \right\}^{\mathrm{T}} \tag{40}$$

In order to make the model satisfy the Gaussian process assumptions and to facilitate the solution, the Gaussian process output is stored using a stack as: $\mathbf{G} = \{\widetilde{g}_1(\mathbf{x}_1), \widetilde{g}_2(\mathbf{x}_1), \cdots, \widetilde{g}_{N_f}(\mathbf{x}_1) \cdots, \widetilde{g}_1(\mathbf{x}_N), \widetilde{g}_2(\mathbf{x}_N), \cdots, \widetilde{g}_{N_f}(\mathbf{x}_N)\}^{\mathrm{T}} \in \mathbb{R}^{N_f \times N}$. To facilitate the derivation of the formula, it is denoted as $\mathbf{G} = \left\{ g_1, g_2, \cdots, g_{N_f \times N} \right\}^{\mathrm{T}}$. Then, the likelihood function of $\mathbf{G}$ can be expressed as:

$$p(\mathbf{G}|\boldsymbol{\gamma}) = \frac{1}{(2\pi)^{N_f \times N/2} (|\mathbf{K}_M|)^{1/2}} \exp[-\frac{1}{2}(\mathbf{G} - \mathbf{b})^T \mathbf{K}_M{}^{-1}(\mathbf{G} - \mathbf{b})] \tag{41}$$

where $\mathbf{b} = [b, b, \cdots, b] \in \mathbb{R}^{N_f N}$ denotes the mean vector of the $N_f \times N$ elements and $\mathbf{K}_M \in \mathbb{R}^{N_f N \times N_f N}$ denotes the covariance matrix of $\mathbf{Y}$. $\boldsymbol{\gamma}$ denotes the parameter vector to be optimized. The definition of the covariance matrix $\mathbf{K}_M$ is the key element that distinguishes the multi-output Gaussian process from the multi-output correlation, and the description of the multi-output correlation is also included in the covariance matrix.

Since the noise is assumed to be an independent and identically distributed random variable, the likelihood function of the sample output for a given training set of samples can be expressed as:

$$p(\mathbf{Y}|\mathbf{G}, \boldsymbol{\gamma}) = \prod_{r=1}^{r=N_f N} p(y_r - g_r|\mathbf{G}, \boldsymbol{\gamma}) = \prod_{r=1}^{r=N_f N} p(\delta_r) = \prod_{i=1}^{i=N_f} \prod_{j=1}^{j=N} p(\delta_{ij}) \tag{42}$$

where $p(\delta)$ is the probability distribution of δ and the expression is:

$$p(\delta_i) = \frac{1}{Z_{\delta_i}} \exp(-\mathbf{C}_i l(\delta_i)) \tag{43}$$

$$Z_{\delta_i} = \int \exp(-\mathbf{C}_i l(\delta_i)) d\delta = \sqrt{2\pi / \mathbf{C}_i} \tag{44}$$

where $l(\delta_i), i = 1, 2, \cdots, N_f$ is the loss function of the support vector machine. $\mathbf{C} = [C_1, C_2, \cdots, C_{N_f}]^{\mathrm{T}}$ is the trade-off constant. For each output, there is a corresponding trade-off constant. According to Bayesian theory, the posterior distribution of $\mathbf{G}$ satisfies:

$$p(\mathbf{G}|\mathbf{Y}, \boldsymbol{\gamma}) = \frac{p(\mathbf{Y}|\mathbf{G}, \boldsymbol{\gamma}) p(\mathbf{G}|\boldsymbol{\gamma})}{p(\mathbf{Y}|\boldsymbol{\gamma})} \tag{45}$$

$p(\mathbf{Y}|\boldsymbol{\gamma})$ is a normalization constant, and further, $p(\mathbf{G}|\mathbf{Y}, \boldsymbol{\gamma})$ can be expressed as (See Appendix A for more details):

$$p(\mathbf{G}|\mathbf{Y}, \boldsymbol{\gamma}) = \frac{1}{Z} \exp(-\mathbf{C}_M l(\mathbf{y} - \mathbf{g})^T - \frac{1}{2}(\mathbf{G} - \mathbf{b})^T \mathbf{K}_M{}^{-1} (\mathbf{G} - \mathbf{b})) \tag{46}$$

where $\mathbf{C} = [C_1, C_2, \cdots, C_{N_f}]^{\mathrm{T}}$, $\mathbf{C}_M = \mathbf{1D} \otimes \mathbf{C}$, $\mathbf{1D} = [1, 1, \cdots, 1] \in \mathbb{R}^{N_f}$, $\otimes$ is Kronecker products.

$$Z = \int \exp(-S(\mathbf{G})) d\mathbf{G} \tag{47}$$

$$S(\mathbf{G}) = \mathbf{C}_M l(\mathbf{y} - \mathbf{g})^{\mathrm{T}} + \frac{1}{2}(\mathbf{G} - \mathbf{b})^T \mathbf{K}_M{}^{-1} (\mathbf{G} - \mathbf{b}) \tag{48}$$

Therefore, maximizing the posterior distribution according to the principle of the great likelihood method of solution can be equated to:

$$\min_{\mathbf{G}} \mathbf{C}_M l(\mathbf{y} - \mathbf{g})^{\mathrm{T}} + \frac{1}{2}(\mathbf{G} - \mathbf{b})^T \mathbf{K}_M{}^{-1} (\mathbf{G} - \mathbf{b}) \tag{49}$$

Similar to the original support vector machine, the first term of the objective function denotes the empirical risk, and the second term, which denotes the smoothness of the function, $\mathbf{C}_M$ is an expansion of the trade-off parameters.

3.2. Model Construction for MBSVR

As with single-output Bayesian support vector machines, MBSVR still uses the squared loss function:

$$l(\delta) = \frac{1}{2}\delta^2 \tag{50}$$

The squared loss function actually obeys a Gaussian probability density function [20]. Bringing in the loss function expression yields the new objective function as

$$\begin{aligned} &\min_{\mathbf{G}} \tfrac{1}{2}\mathbf{C}_M\mathbf{e}^2 + \tfrac{1}{2}(\mathbf{G}-\mathbf{b})^T {K_M}^{-1}(\mathbf{G}-\mathbf{b}) \\ &s.t. y_r = g_r + e_r \end{aligned} \tag{51}$$

where $\mathbf{e} = \left\{e_1, e_2, \cdots, e_{N_f N}\right\}$. $e_r = y_r - g_r, r = 1, 2, \cdots, N_f N$. $\mathbf{C}_M = \mathbf{1D} \otimes \mathbf{C}$. The estimate of $\mathbf{G}$ is (See Appendix B for more details):

$$\hat{\mathbf{G}} = \mathbf{K}_M(\mathbf{K}_M + \mathbf{C}_h)^{-1}\mathbf{Y} + \mathbf{b} = \mathbf{K}_M\boldsymbol{\beta} + \mathbf{b} \tag{52}$$

where $\mathbf{C}_h$ satisfies $\mathbf{C}_h \odot diag(\mathbf{C}_M) = \mathbf{I}$, $diag(\mathbf{C}_M)$ is in diagonal form of $\mathbf{C}_M$, $\mathbf{I} \in \mathbb{R}^{N_f N \times N_f N}$ is a unit matrix, $\boldsymbol{\beta} = [\beta_1, \beta_2, \cdots \beta_{N_f \times N}] = (\mathbf{K}_M + \mathbf{C}_h)^{-1}\mathbf{Y}$. $\odot$ is the Hadamard product, denoting the element-by-element multiplication of the matrix. For the output to be predicted $\mathbf{g}(\mathbf{x})$, its joint distribution with the training set is satisfied:

$$\begin{pmatrix} \mathbf{g}(\mathbf{x}) \\ \mathbf{G} \end{pmatrix} \sim \mathcal{N}\left(\begin{Bmatrix} \mathbf{b}_0 \\ \mathbf{b} \end{Bmatrix}, \begin{Bmatrix} \mathbf{k}_M(\mathbf{x},\mathbf{x}) & \mathbf{k}_M(\mathbf{x},\mathbf{X}) \\ \mathbf{k}_M(\mathbf{X},\mathbf{x}) & \mathbf{k}_M(\mathbf{X},\mathbf{X}) \end{Bmatrix} \right) \tag{53}$$

$$\mathbf{k}_M(\mathbf{X},\mathbf{x}) = \mathbf{k}_M(\mathbf{x},\mathbf{X})^T = [\mathbf{k}_M(\mathbf{x}_1,\mathbf{x}), \cdots, \mathbf{k}_M(\mathbf{x}_N,\mathbf{x})]^T \tag{54}$$

$$\mathbf{k}_M(\mathbf{X},\mathbf{X}) = \begin{bmatrix} \mathbf{k}_{11}(\mathbf{X},\mathbf{X}) & \cdots & \mathbf{k}_{1N_f}(\mathbf{X},\mathbf{X}) \\ \vdots & \ddots & \vdots \\ \mathbf{k}_{N_f 1}(\mathbf{X},\mathbf{X}) & \cdots & \mathbf{k}_{N_f N_f}(\mathbf{X},\mathbf{X}) \end{bmatrix} \tag{55}$$

where $\mathbf{k}_M(\mathbf{X},\mathbf{X}) = \mathbf{K}_M$, $\mathbf{b}_0 = [b, b, \cdots, b] \in \mathbb{R}^{N_f}$.The prior of $\mathbf{g}(\mathbf{x})$ still obeys a multi-output Gaussian distribution:

$$p(\mathbf{g}(\mathbf{x})|\mathbf{G}) = \mathcal{N}(\mu(\mathbf{x}), \Sigma^2(\mathbf{x})) \tag{56}$$

$$\mu(\mathbf{x}) = \mathbf{b}_0 + \mathbf{k}_M(\mathbf{x},\mathbf{X})(\mathbf{k}_M(\mathbf{X},\mathbf{X}) + \mathbf{C}_h)^{-1}(\mathbf{G}-\mathbf{b}) \tag{57}$$

$$\Sigma^2(\mathbf{x}) = \mathbf{k}_M(\mathbf{x},\mathbf{x}) - \mathbf{k}_M(\mathbf{x},\mathbf{X})(\mathbf{k}_M(\mathbf{X},\mathbf{X}) + \mathbf{C}_h))^{-1}\mathbf{k}_M(\mathbf{X},\mathbf{x}) \tag{58}$$

where $\mathbf{b}_0 = [b, b, \cdots, b] \in \mathbb{R}^{N_f}$, $\mathbf{k}_M(\mathbf{x},\mathbf{X}) \in \mathbb{R}^{N_f \times N_f N}$,the expression is:

$$\mathbf{k}_M(\mathbf{x},\mathbf{X}) = \begin{bmatrix} \mathbf{k}_{11}(\mathbf{x},\mathbf{X}) & \cdots & \mathbf{k}_{1N_f}(\mathbf{x},\mathbf{X}) \\ \vdots & \ddots & \vdots \\ \mathbf{k}_{N_f 1}(\mathbf{x},\mathbf{X}) & \cdots & \mathbf{k}_{N_f N_f}(\mathbf{x},\mathbf{X}) \end{bmatrix} \tag{59}$$

$$\mathbf{k}_{ii'}(\mathbf{x},\mathbf{X}) = [\mathbf{k}_{ii'}(\mathbf{x},\mathbf{x}_1), \mathbf{k}_{ii'}(\mathbf{x},\mathbf{x}_2), \cdots, \mathbf{k}_{ii'}(\mathbf{x},\mathbf{x}_N)], i = 1, 2, \cdots N_f \tag{60}$$

$$\mathbf{k}_M(\mathbf{x},\mathbf{x}) = \begin{bmatrix} \mathbf{k}_{11}(\mathbf{x},\mathbf{x}) & \cdots & \mathbf{k}_{1N_f}(\mathbf{x},\mathbf{x}) \\ \vdots & \ddots & \vdots \\ \mathbf{k}_{N_f 1}(\mathbf{x},\mathbf{x}) & \cdots & \mathbf{k}_{N_f N_f}(\mathbf{x},\mathbf{x}) \end{bmatrix} \in \mathbb{R}^{N_f \times N_f} \tag{61}$$

The variance of the *ith* diagonal element of $\Sigma^2(\mathbf{x})$ corresponds to the variance of *ith* output of $\mathbf{x}$.

3.3. Optimized Solution of Parameters

In MBSVR, the parameters to be optimized include the kernel function parameter $\boldsymbol{\theta}_M$; the trade-off parameter $\mathbf{C}$; and the matrix $\mathbf{A}$, which describes the correlations between outputs. For computational convenience, the specific implementation is decomposed by

Cholesky $\mathbf{A}$ into $\mathbf{RR}^T$. The optimal values of these hyperparameters are determined by the maximum a posteriori probability:

$$p(\boldsymbol{\gamma}) = \frac{p(\mathbf{Y}|\boldsymbol{\gamma})p(\boldsymbol{\gamma})}{p(\mathbf{Y})} \tag{62}$$

where $\boldsymbol{\gamma} = \{\boldsymbol{\theta}_M, \mathbf{C}, \mathbf{R}\}$, $p(\boldsymbol{\gamma})$ is the prior distribution of the hyperparameters, and $p(\mathbf{Y})$ is a regularization constant that in general specifies a uniform probability distribution for the hyperparameters. Therefore, its prior distribution $p(\boldsymbol{\gamma})$ is a constant. Therefore, it is only necessary to maximize $p(\mathbf{Y}|\boldsymbol{\gamma})$ to achieve the purpose of great likelihood estimation of the parameters:

$$p(\mathbf{Y}|\boldsymbol{\gamma}) = \int p(\mathbf{Y}|\mathbf{G}, \gamma)p(\mathbf{G}|\boldsymbol{\gamma})d\mathbf{G} = \frac{1}{Z_G \prod\limits_{i=1}^{N_f} Z_{\delta_i}^N} \int \exp(-S(\mathbf{G}))d\mathbf{G} \tag{63}$$

where $Z_G = (2\pi)^{N_f \times N/2}(|\mathbf{K}_M|)^{1/2}$, $Z_{\delta_i} = \int \exp(-C_i l(\delta_i))d\delta_i$. $S(\mathbf{G})$ can be expressed as

$$S(\mathbf{G}) = S(\hat{\mathbf{G}}) + \frac{1}{2}(\mathbf{G} - \hat{\mathbf{G}})^T(\mathbf{K}_M{}^{-1} + diag(\mathbf{C}_M) \odot \mathbf{I})(\mathbf{G} - \hat{\mathbf{G}}) \tag{64}$$

Bringing (64) into the probability distribution in (63) yields the following equation:

$$-\ln(p(\mathbf{Y}|\boldsymbol{\gamma})) = \frac{1}{2}\mathbf{C}_M\mathbf{e}^2 + \frac{1}{2}\boldsymbol{\beta}^T\mathbf{K}_M\boldsymbol{\beta} + \frac{1}{2}\ln|\mathbf{I} + diag(\mathbf{C}_M) \odot \mathbf{K}_M| + N\sum_{i=1}^{N_f}\ln Z_{\delta_i} \tag{65}$$

The hyperparameters are obtained by solving according to the minimized likelihood function. The nonlinear programming problem is solved using the “fmincon” function in MATLAB2022b. Given the initial solution, iterative optimization is performed to obtain the optimal hyperparameters. In general, the method can find the global optimal solution of the objective function, and the initial value of the parameters on the optimal solution of the parameters has less influence.

4. Numerical Experiments

4.1. Performance Metrics and Experiment Settings

The single-response Bayesian support vector machine [10] and a multi-output Gaussian process model [21] are used for comparison with the multi-output Bayesian support vector machine model (MBSVR). The three methods are denoted by (independent regression) IND, (multi-output Gaussian process) MGP, and MBSVR, respectively. The lhsdesign function in MATLAB is used to generate the training set and test set, and for the model parameters, the kernel function parameter in MBSVR is initialized to 1, the trade-off parameter C is initialized to 1000, and the range of optimality search is $[0.01, 100]$ and $[1, 10^6]$. The kernel function parameter of the multi-output Gaussian process is initialized to 1. In the multi-output Bayesian support vector machine, $\mathbf{A}_q$ is initialized to a unit matrix. For the same hyperparameters in all three models, the same initial values and optimization ranges are assigned to allow for a fairer comparison.

To measure the generalization effect of the model, the error criterion normalized root mean square error (NRMSE) is used, and its expression is:

$$NRMSE_i = \frac{\sqrt{\frac{1}{Tn}\sum\limits_{t=1}^{Tn}(y_{it} - \widetilde{y}_{it})}}{\max(\mathbf{y}_i) - \min(\mathbf{y}_i)} \tag{66}$$

where Tn is the size of the test set, i denotes ith output, y_{it} is the true value of the t test set for the i first output, and $\widetilde{y}_{it}$ is the corresponding predicted value. The smaller the value of the error metric, the more accurate the model is. The performance of the model is affected not only by the training set size but also by the specific sample points. Therefore, in order to ensure the diversity of the experiments and fully reflect the model performance, repeat the experiments 100 times; 100 training sets are generated with the same number of points and statistically characterize the results. In this paper, we use the Pearson correlation coefficient for correlation analysis of the output before proceeding with the model construction, which is the most used [13].

4.2. Datasets Description

In order to evaluate the modeling effect of the three models, two function cases and four real data are selected to verify the effect of the proposed method. The functions are based on the original cases with some changes so that their outputs have a certain degree of relevance, and the selected function cases include the one-dimensional case of Forrester and the two-dimensional case of Branin, and all data details are shown in Table 1.

Table 1. Information about the test dataset.

Type	Name	D	N_f	Dataset Size	Train Size	Test Size
Numerical function	Forrester	1	2		4:1:8	100
	Branin	2	2		5:3:20	5000
Real-world datasets	Energy_efficiency	8	2	768	10:5:40	500
	Polymer	10	4	61	10:5:35	20
	Broomcorn	19	3	128	30:10:90	30
	Sarcos	21	7	44,484	10:5:50	4449

The specific expressions of Forrester and Branin are given in (Appendix C), and the information of Energy_efficiency, Polymer, Broomcorn, and Sarcos are as follows:

Energy_efficiency [22]. This dataset contains a total of two outputs for heat load and cooling load demand for building energy efficiency. It includes eight factors, such as lighting area, roof area, and overall height. There is a total of 768 sample points in this dataset.

Polymer [16]. The polymer dataset contains ten inputs, such as temperature and feed rate, and contains four outputs of measurements. The dataset contains a total of 61 samples.

Broomcorn [22]. This dataset is a sorghum sample from the Institute of Crop Seed Resources, Chinese Academy of Agricultural Sciences, containing a total of 128 sample points. It contains 19 inputs and 3 outputs. The outputs are the protein, lysine, and starch fractional content of the sorghum samples, respectively.

Sarcos [13,23]. This dataset is a high-dimensional, large-scale dataset. This dataset is an inverse dynamics modeling problem for a 7-degree-of-freedom anthropomorphic robotic arm that has 21 inputs (7 joint positions, 7 joint velocities, and 7 joint accelerations) and corresponding moments at 7 joints as outputs. Only the modeling results for six responses are shown.

4.3. Results and Discussions

4.3.1. Numerical Functions

From Forrester's expression, it can be found that the output f1 is a nonlinear variation of f2. As shown in Figure 2, the actual output of the two functions is computationally highly correlated, with a Pearson correlation coefficient of 0.95.

Figure 3 shows the modeling results for the two outputs of Forrester. The bottom and top of the boxplot indicate the lower and upper quartiles, respectively, while the center depression indicates the median of the metrics. The maximum length of the vertical line at the end of the boxplot is 1.5 times the interquartile spacing. A red plus sign indicates an outlier that is outside the boundaries of the vertical line. It can be seen that when the number of sample points is small (4), the three modeling methods are roughly equivalent.

As the number of training sets increases, the two modeling methods MBSVR and MGP, which consider output correlation. This phenomenon may be due to the fact that when the number of samples is small, the current information cannot support the accurate solution of relevant parameters such as correlation. Moreover, MBSVR has the best results and the most significant advantage when the size of the training set is 5 or 6, after which the modeling results of MBSVR are gradually comparable to the modeling results of MGP. This may be related to the small-sample modeling capability of SVR.

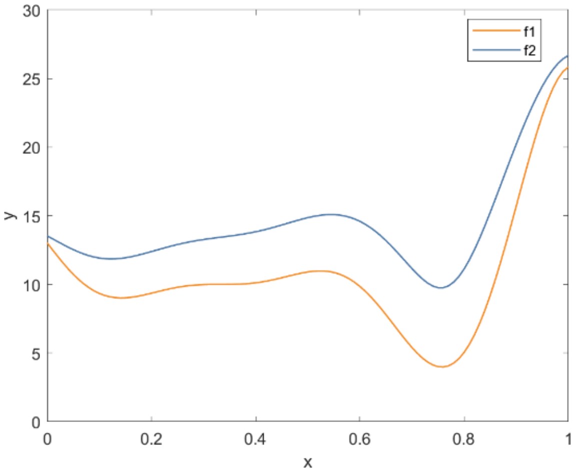


Figure 2. Two outputs for Forrester.

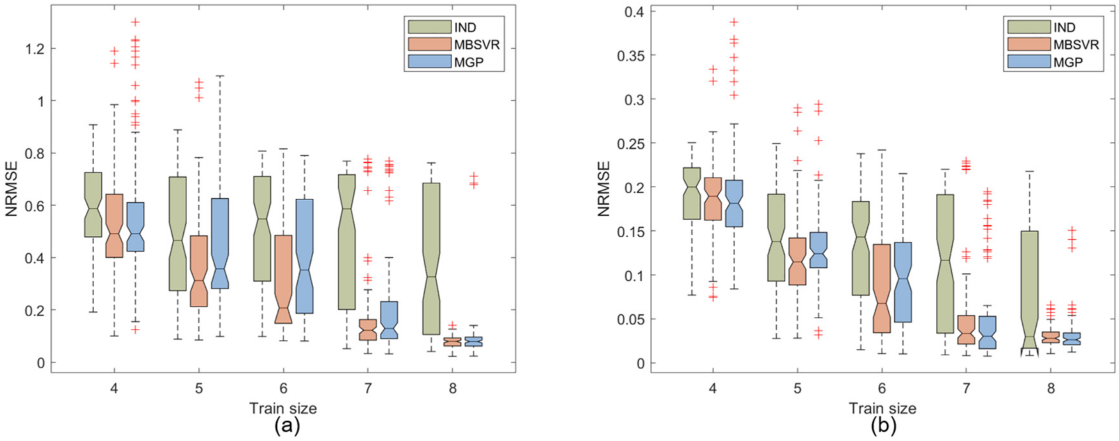


Figure 3. Modeling results for Forrester: (**a**) Results for Output 1; (**b**) results for Output 2.

Next, a two-dimensional function Branin with two outputs is used to validate method validity. Output 2 is the original Branin function, and Output 1 is a fine-tuning of the original Branin with a linear translation added, as shown in Figure 4. These two outputs are also strongly correlated, with a Pearson correlation coefficient of 0.69.

Figure 5 shows the modeling results of Branin. Overall, the modeling method considering correlation outperforms the independent modeling method, and for both Output 1 and Output 2, the accuracy of the MBSVR is higher than that of the MGP for moderate training set sizes (11, 14, 17). For training set size 20, the accuracy of the MBSVR model is roughly comparable to that of the MGP model, and even lower than that of the MGP in some sample point cases are even lower than MGP. This phenomenon, which is the modeling advantage of MBSVR becoming progressively less significant as the training set increases, is consistent with the one-dimensional arithmetic case.

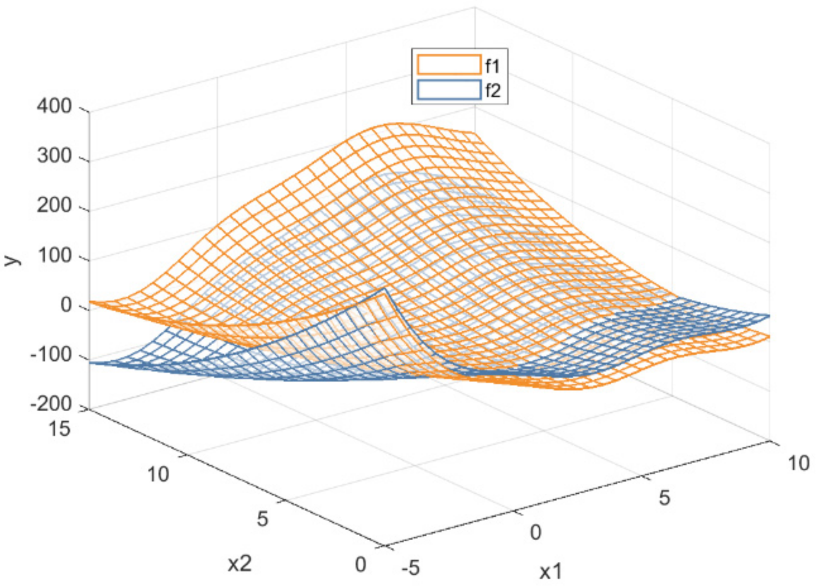


Figure 4. Two outputs for Branin.

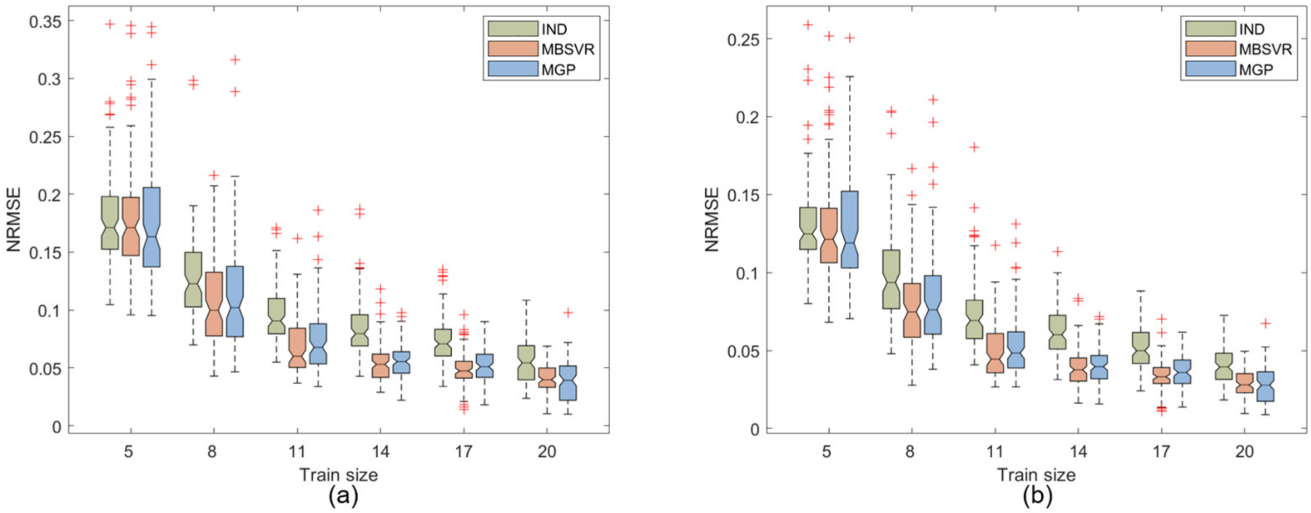
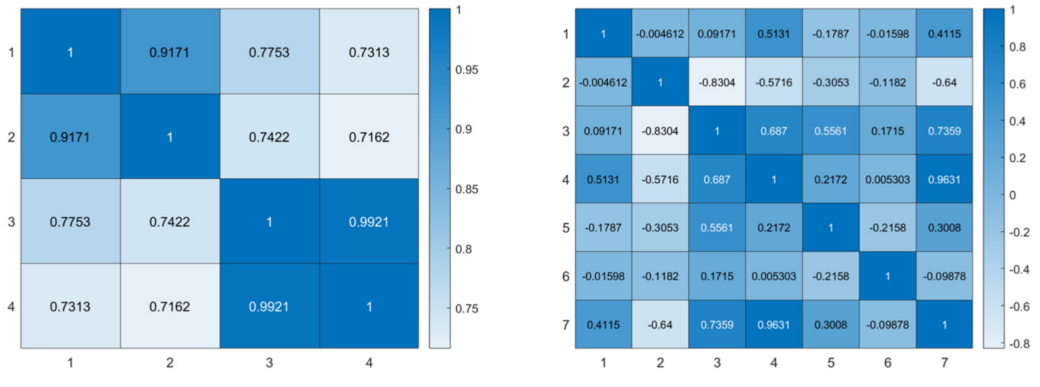


Figure 5. Modeling results for Branin: (**a**) Results for Output 1; (**b**) results for Output 2.

4.3.2. Real-World Dataset

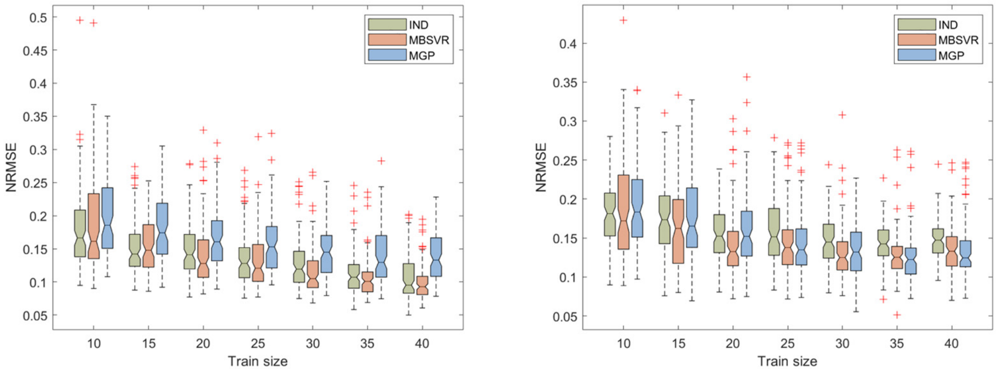
According to the results of the correlation analysis, the Pearson correlation coefficient of Energy_efficiency is greater than 0.97. The Pearson correlation coefficients of the four outputs of Polymer are large, as shown in Figure 6a, and there is a strong correlation between Output 1 and Output 2 as well as between Output 3 and Output 4. As for Broomcorn, the Pearson correlation coefficient of Output 1 and Output 2 is 0.39, the correlation coefficient of Output 1 and Output 3 is -0.67, and the correlation coefficient of Output 2 and Output 3 is -0.37. The results of the output correlation analysis of Sarcos are shown in Figure 6b. The correlation of Output 4 and Output 7 is the highest, and the correlation of Output 1 and Output 2 is the lowest. Output 6 has low correlation with all other outputs.



(a) (b)

Figure 6. Pearson correlation coefficient for polymer and Sarcos: (**a**) Pearson correlation coefficient of polymer. (**b**) Pearson correlation coefficient of Sarcos.

Figures 7–10 shows the modeling results for each dataset. As the number of sample points increases, the model accuracy gradually improves but the rate of improvement gradually decreases, as can be seen from the modeling results:



(a) (b)

Figure 7. Modeling results for Energy_efficiency: (**a**) Results for Output 1; (**b**) results for Output 2.

In most cases, modeling methods that consider correlation outperform independent modeling methods, and in aggregate, MBSVR works better. For example, MGP and MBSVR outperform the metric modeling approach BSVR in Output 1 for Energy_efficiency, Output 2 and Output 3 for Broomcorn, and most of the outputs for Sarcos. Except for Output 1 for Broomcorn, where the accuracy of MBSVR is significantly lower than that of the independent modeling approach (IND). In conclusion, MBSVR is more accurate in many problems than other modeling methods.

In some cases, independent modeling methods will outperform methods that consider correlations, such as polymer's output 3 and Broomcorn's output 1. For Broomcorn's output 1, the independent modeling approach significantly outperforms the other. Observing the correlation coefficients, it can be found that the Pearson correlation coefficient between output 1 and output 2 is 0.39, and the correlation coefficient between output 1 and output 3 is -0.67. This disadvantage in accuracy may be due to the lack of obvious correlation, or it may be due to the model's inability to accurately approximate the real shared information.

As the training size increases, the model accuracy advantage shows two different trends. In Output 1 and Output 4 for polymer and Output 2 and Output 3 for Broomcorn, the advantage of MBSVR over other modeling methods is more pronounced as the training size increases. However, there are also cases where the advantage of MBSVR is not obvious as the training size increases, e.g., in the Sacros dataset, the modeling accuracy of MBSVR gradually converges to the modeling accuracy of MGP as the training set increases. Theoretically, as the training set increases, the hyperparameter estimates should get closer and closer to the true values, and if the model assumptions are correct, the hyperparameter estimates are accurate, and the model accuracy improves as the training set increases.

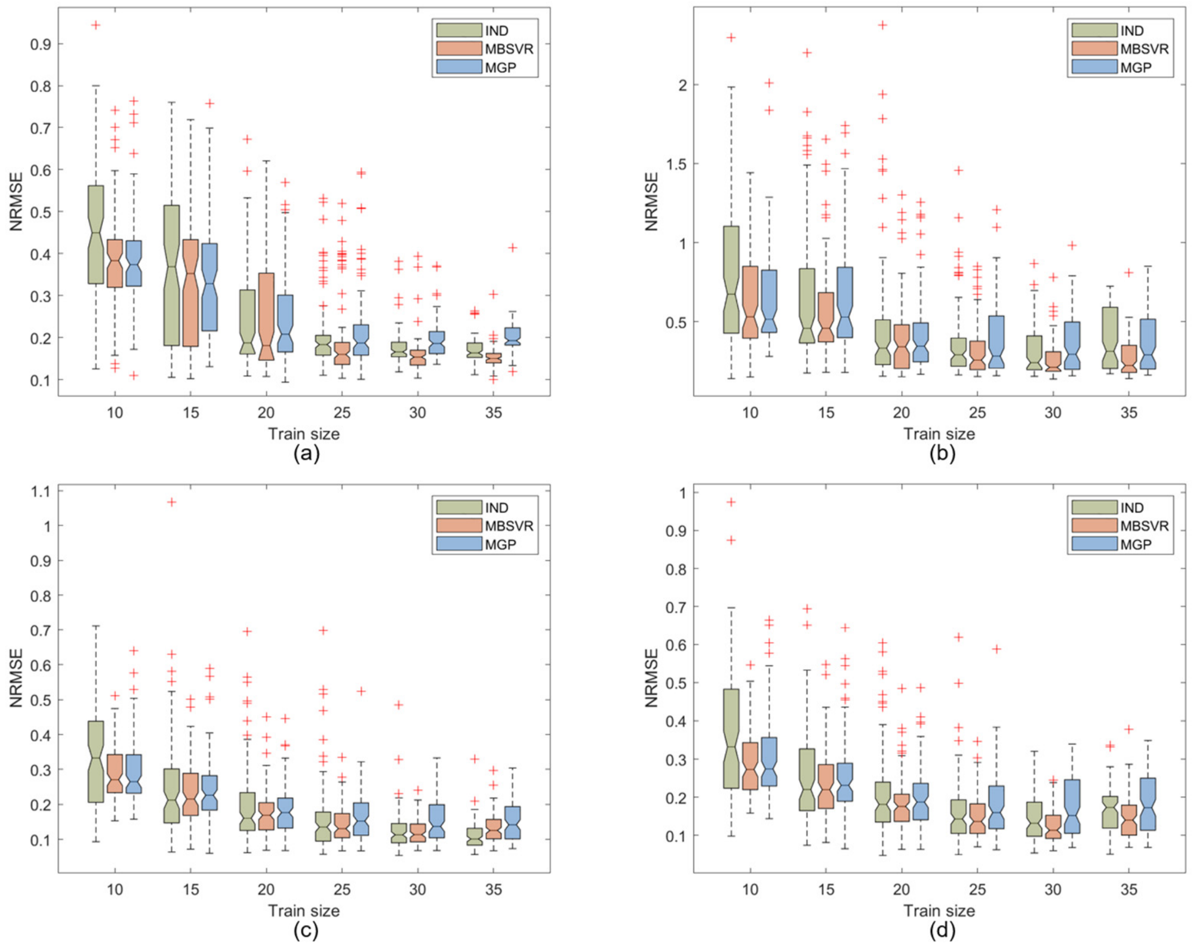


Figure 8. Modeling results for polymer: (**a**) results for Output 1; (**b**) results for Output 2; (**c**) results for Output 3; (**d**) results for Output 4.

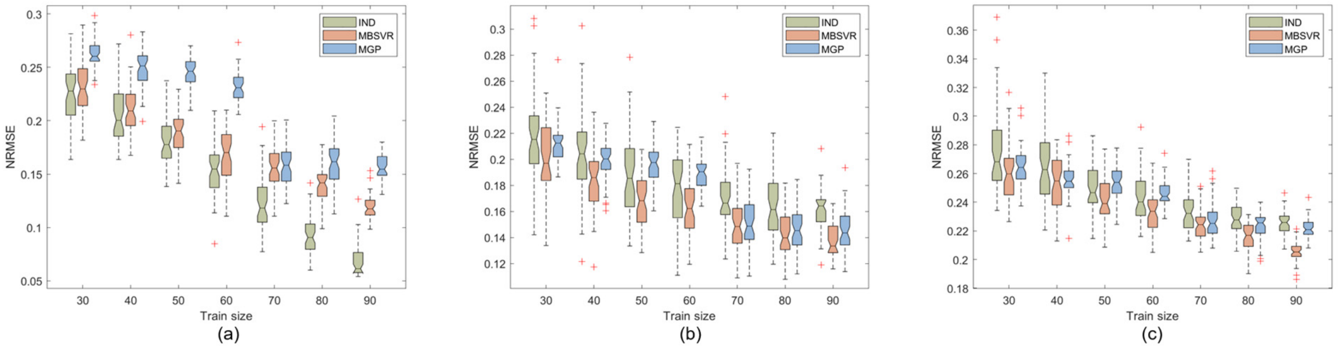


Figure 9. Modeling results for Broomcorn: (**a**) results for Output 1; (**b**) results for Output 2. (**c**) results for Output 3.

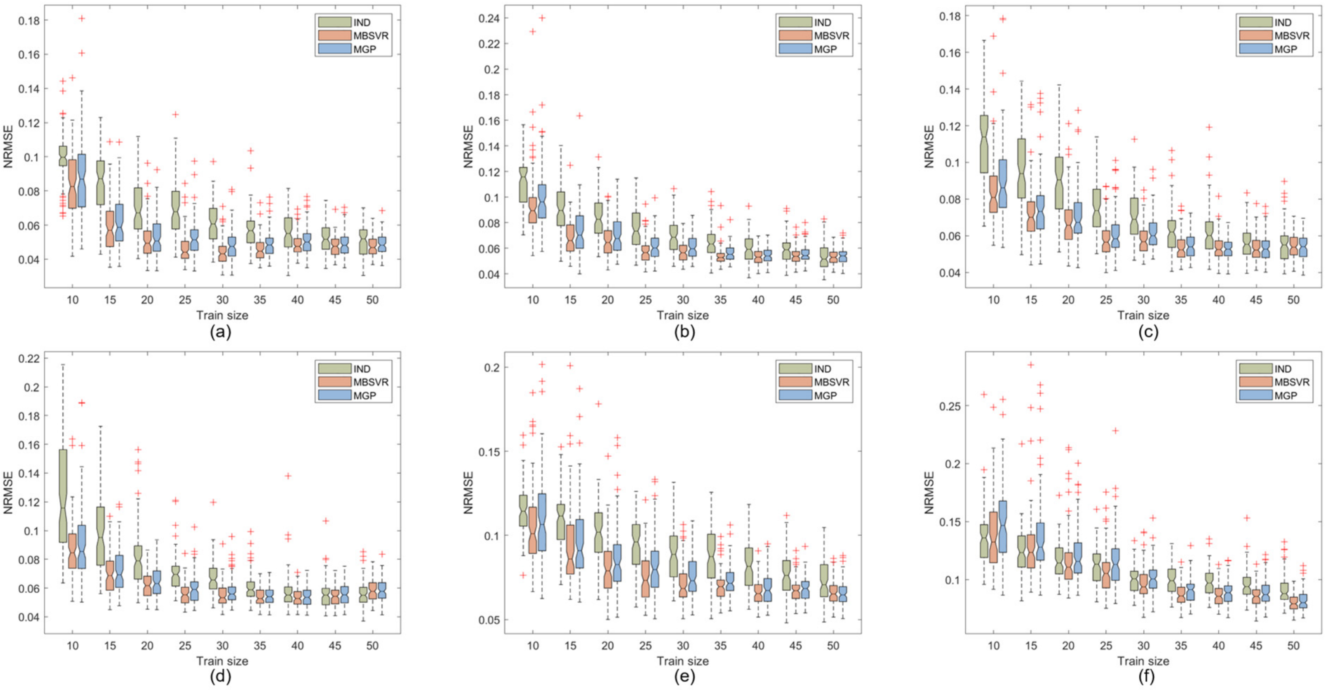


Figure 10. Modeling results for Sarcos (f1–f6): (**a**) results for Output 1; (**b**) results for Output 2. (**c**) Results for Output 3; (**d**) results for Output 4; (**e**) results for Output 5. (**f**) results for Output 6.

From the results of the numerical analysis, it can be seen that in some arithmetic cases, due to the large number of hyperparameters that need to be optimized by MBSVR, the discrepancy between the shared information obtained from modeling and the actual accurate information leads to limited room for improvement in model accuracy. The final modeling accuracy is comparable to that of MGP. This discrepancy also leads to the fact that in some outputs, the accuracy of modeling methods that consider correlation may be lower than that of independent modeling methods. Additionally, MBSVR requires more optimized hyperparameters, resulting in lower time efficiency.

5. Conclusions

This paper investigates the multi-output modeling problem, aiming to improve the model accuracy by quantitatively describing the correlation between the outputs and using the information between the outputs to construct the model for multiple outputs simultaneously. To inherit the advantages of a single-output Bayesian support vector machine, based on it, the SLFM model is introduced, combined with Bayesian derivation, and the hyperparameters are optimized comprehensively to get the multi-output model that can predict multiple output means and probability distributions at the same time. Model validation is carried out on simple function arithmetic cases and real datasets, and overall, the MBSVR accuracy is higher due to the single-output modeling and the multi-output Gaussian process model.

Due to the large number of hyperparameters that need to be optimized in MBSVR, the efficiency of the algorithm is low. In addition, inaccurate hyperparameters make a difference between the shared information and the actual accurate information, resulting in limited room for improvement in model accuracy. Therefore, achieving efficient and accurate parameter optimization is the main problem that needs to be solved in the future. In addition, how to simplify the correlation description structure and further improve the applicability and optimization efficiency of the model is also the direction of further research.

Author Contributions: Conceptualization, Y.W.; methodology, Y.W.; validation, Y.W. and Z.W.; writing—original draft preparation, Y.W.; writing—review and editing, Y.W., Z.C. and Z.W.; visualization, Y.W. and Z.W.; supervision, Z.C. and Z.W.; funding acquisition, Z.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Natural Science Foundation of China (No. 72171231).

Data Availability Statement: All data generated or analyzed during this study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

The posterior distribution in (48) considering multiple outputs can be described as:

$$
\begin{aligned}
p(\mathbf{G}|\mathbf{Y},\boldsymbol{\gamma}) &= \frac{p(\mathbf{Y}|\mathbf{G},\boldsymbol{\gamma})p(\mathbf{G}|\boldsymbol{\gamma})}{p(\mathbf{Y}|\boldsymbol{\gamma})} \\
&= \prod_{i=1}^{i=N_f}\prod_{j=1}^{j=N} p(\delta_{ij}) \times \frac{1}{(2\pi)^{N_f\times N/2}(|\mathbf{K}_M|)^{1/2}} \exp[-\frac{1}{2}(\mathbf{G}-\mathbf{b})^T\mathbf{K}_M{}^{-1}(\mathbf{G}-\mathbf{b})] \\
&= \prod_{i=1}^{N_f}\frac{1}{Z_{\delta_i}}\exp(-\mathbf{C}_i l(\delta_i)) \times \frac{1}{(2\pi)^{N_f\times N/2}(|\mathbf{K}_M|)^{1/2}} \exp[-\frac{1}{2}(\mathbf{G}-\mathbf{b})^T\mathbf{K}_M{}^{-1}(\mathbf{G}-\mathbf{b})] \\
&= \frac{\exp(\sum\limits_{i=1}^{N_f} -\mathbf{C}_i l(\delta_i) - \frac{1}{2}(\mathbf{G}-\mathbf{b})^T\mathbf{K}_M{}^{-1}(\mathbf{G}-\mathbf{b}))}{\prod\limits_{i=1}^{N_f} Z_{\delta_i} \times (2\pi)^{N_f\times N/2}(|\mathbf{K}_M|)^{1/2}} \\
&= \frac{\exp(\sum\limits_{i=1}^{N_f} -\mathbf{C}_i l(\mathbf{y}_i-\mathbf{g}_i) - \frac{1}{2}(\mathbf{G}-\mathbf{b})^T\mathbf{K}_M{}^{-1}(\mathbf{G}-\mathbf{b}))}{\prod\limits_{i=1}^{N_f}\int\exp(-\mathbf{C}_i l(\mathbf{y}_i-\mathbf{g}_i))dG \times \int\exp(-\frac{1}{2}(\mathbf{G}-\mathbf{b})^T\mathbf{K}_M{}^{-1}(\mathbf{G}-\mathbf{b}))dG} \\
&= \frac{\exp(\sum\limits_{i=1}^{N_f} -\mathbf{C}_i l(\mathbf{y}_i-\mathbf{g}_i) - \frac{1}{2}(\mathbf{G}-\mathbf{b})^T\mathbf{K}_M{}^{-1}(\mathbf{G}-\mathbf{b}))}{\int\exp(\sum\limits_{i=1}^{N_f}(-\mathbf{C}_i l(\mathbf{y}_i-\mathbf{g}_i) - \frac{1}{2}(\mathbf{G}-\mathbf{b})^T\mathbf{K}_M{}^{-1}(\mathbf{G}-\mathbf{b})))dG}
\end{aligned}
$$

Appendix B

The Lagrangian function of the primal optimization problem in (53) reads:

$$
\mathcal{L}(\mathbf{G},\boldsymbol{\alpha},\mathbf{e},b) = \frac{1}{2}(\mathbf{G}-\mathbf{b})\mathbf{K}_M^{-1}(\mathbf{G}-\mathbf{b}) + \frac{1}{2}\mathbf{C}_M\mathbf{e}^2 + \sum_{r=1}^{N_fN}\beta_r(y_r-g_r-e_r)
$$

where $\mathbf{e} = [e_1, e_2, \cdots, e_{N_fN}]^{\mathrm{T}}$, $\boldsymbol{\beta} = [\beta_1, \beta_2, \cdots, \beta_{N_fN}]^{\mathrm{T}}$ is Lagrange multiplier. According to the Karush–Kuhn–Tucker conditions, we can obtain:

$$
\begin{aligned}
\frac{\partial\mathcal{L}(\mathbf{G},\boldsymbol{\beta},\mathbf{e},b)}{\partial\mathbf{G}} &= 0 \rightarrow \mathbf{G} = \mathbf{K}_M\boldsymbol{\beta} + \mathbf{b} \\
\frac{\partial\mathcal{L}(\mathbf{G},\boldsymbol{\beta},\mathbf{e},b)}{\partial e_r} &= 0 \rightarrow \sum_{r=1}^{N_fN}\beta_r = \mathbf{C}_M\mathbf{e} \\
\frac{\partial\mathcal{L}(\mathbf{G},\boldsymbol{\beta},\mathbf{e},b)}{\partial\beta_r} &= 0 \rightarrow y_r = g_r + e_r \\
\frac{\partial\mathcal{L}(\mathbf{G},\boldsymbol{\beta},\mathbf{e},b)}{\partial b} &= 0 \rightarrow \sum_{r=1}^{N_fN}\beta_r = 0
\end{aligned}
$$

Solving the above equation, we obtain the optimal values of G as

$$\hat{\mathbf{G}} = \mathbf{K}_M\boldsymbol{\beta} + \mathbf{b}$$

Appendix C

The expression of the Forrester function is:

$$f_1(x) = 1.5(x+2.5)\sqrt{(6x-2)^2\sin(12x-4)+10}, x \in [0,1]$$

$$f_2(x) = (6x-2)^2\sin(12x-4)+10, x \in [0,1]$$

The expression of the Branin function is:

$$f_1(x) = (x_2 - \frac{3}{4\pi}x_1^2 + \frac{4}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos(x_1) + 2x_1 - 9x_2 + 32$$

$$f_2(x) = (x_2 - \frac{5.1}{4\pi}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos(x_1) + 10$$

$$x_1 \in [-5,10], x_2 \in [0,15]$$

References

1. Ghattas, B.; Manzon, D. Machine Learning Alternatives to Response Surface Models. *Mathematics* **2023**, *11*, 3406. [CrossRef]
2. Yu, H.; Yang, K.; Zhang, L.; Wang, W.; Ouyang, M.; Ma, B.; Yang, S.; Li, J.; Liu, X. Multi-Output Ensemble Deep Learning: A Framework for Simultaneous Prediction of Multiple Electrode Material Properties. *Chem. Eng. J.* **2023**, *475*, 146280. [CrossRef]
3. Zhou, Y.; Chang, F.-J.; Chang, L.-C.; Kao, I.-F.; Wang, Y.-S.; Kang, C.-C. Multi-Output Support Vector Machine for Regional Multi-Step-Ahead PM2.5 Forecasting. *Sci. Total Environ.* **2019**, *651*, 230–240. [CrossRef] [PubMed]
4. Nguyen, N.-H.; Abellán-García, J.; Lee, S.; Nguyen, T.-K.; Vo, T.P. Simultaneous Prediction the Strain and Energy Absorption Capacity of Ultra-High Performance Fiber Reinforced Concretes by Using Multi-Output Regression Model. *Constr. Build. Mater.* **2023**, *384*, 131418. [CrossRef]
5. Vapnik, V.N. *The Nature of Statistical Learning Theory*; Springer: New York, NY, USA, 2000; ISBN 978-1-4419-3160-3.
6. Roy, A.; Chakraborty, S. Support Vector Machine in Structural Reliability Analysis: A Review. *Reliab. Eng. Syst. Saf.* **2023**, *233*, 109126. [CrossRef]
7. Chen, L.; Pan, Y.; Zhang, D. Prediction of Carbon Emissions Level in China's Logistics Industry Based on the PSO-SVR Model. *Mathematics* **2024**, *12*, 1980. [CrossRef]
8. Chu, W.; Keerthi, S.S.; Ong, C.J. Bayesian Support Vector Regression Using a Unified Loss Function. *IEEE Trans. Neural Netw.* **2004**, *15*, 29–44. [CrossRef] [PubMed]
9. Wang, J.; Li, C.; Xu, G.; Li, Y.; Kareem, A. Efficient Structural Reliability Analysis Based on Adaptive Bayesian Support Vector Regression. *Comput. Methods Appl. Mech. Eng.* **2021**, *387*, 114172. [CrossRef]
10. Cheng, K.; Lu, Z. Adaptive Bayesian Support Vector Regression Model for Structural Reliability Analysis. *Reliab. Eng. Syst. Saf.* **2021**, *206*, 107286. [CrossRef]
11. Cheng, K.; Lu, Z. Active Learning Bayesian Support Vector Regression Model for Global Approximation. *Inf. Sci.* **2021**, *544*, 549–563. [CrossRef]
12. Gao, H.; Ma, Z. Geometric Metric Learning for Multi-Output Learning. *Mathematics* **2022**, *10*, 1632. [CrossRef]
13. Liu, H.; Cai, J.; Ong, Y.-S. Remarks on Multi-Output Gaussian Process Regression. *Knowl.-Based Syst.* **2018**, *144*, 102–121. [CrossRef]
14. Perez-Cruz, F.; Camps-Valls, G.; Soria-Olivas, E.; Perez-Ruixo, J.J.; Figueiras-Vidal, A.R.; Artes-Rodrıguez, A. *Multi-Dimensional Function Approximation and Regression Estimation*; Springer: Berlin/Heidelberg, Germany, 2002.
15. Zhang, W.; Liu, X.; Ding, Y.; Shi, D. Multi-Output LS-SVR Machine in Extended Feature Space. In Proceedings of the 2012 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications (CIMSA) Proceedings, Tianjin, China, 2–4 July 2012; pp. 130–134.
16. Xu, S.; An, X.; Qiao, X.; Zhu, L.; Li, L. Multi-Output Least-Squares Support Vector Regression Machines. *Pattern Recognit. Lett.* **2013**, *34*, 1078–1084. [CrossRef]
17. Tran, N.K.; Kühle, L.C.; Klau, G.W. A Critical Review of Multi-Output Support Vector Regression. *Pattern Recognit. Lett.* **2024**, *178*, 69–75. [CrossRef]
18. Bayes, T. An Essay towards Solving a Problem in the Doctrine of Chances. *Reson* **2003**, *8*, 80–88. [CrossRef]
19. Fricker, T.E.; Oakley, J.E.; Urban, N.M. Multivariate Gaussian Process Emulators with Nonseparable Covariance Structures. *Technometrics* **2013**, *55*, 47–56. [CrossRef]

20. Karal, O. Maximum Likelihood Optimal and Robust Support Vector Regression with Lncosh Loss Function. *Neural Netw.* **2017**, *94*, 1–12. [CrossRef] [PubMed]
21. Li, M.; Hu, C. Multivariate System Reliability Analysis Considering Highly Nonlinear and Dependent Safety Events. *Reliab. Eng. Syst. Saf.* **2018**, *180*, 189–200. [CrossRef]
22. Zhu, X.; Gao, Z. An Efficient Gradient-Based Model Selection Algorithm for Multi-Output Least-Squares Support Vector Regression Machines. *Pattern Recognit. Lett.* **2018**, *111*, 16–22. [CrossRef]
23. Liu, H.; Ding, J.; Xie, X.; Jiang, X.; Zhao, Y.; Wang, X. Scalable Multi-Task Gaussian Processes with Neural Embedding of Coregionalization. *Knowl.-Based Syst.* **2022**, *247*, 108775. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.