Using Gaussian process to enhance support vector regression

Zhang, Yi; Yao, Wen; Chen, Xiaoqian; van Keulen, Fred

DOI
10.1007/978-3-319-67988-4_20

Publication date
2017

Document Version
Final published version

Published in
Advances in Structural and Multidisciplinary Optimization

Using Gaussian Process to Enhance Support Vector Regression

Yi Zhang[1(✉)], Wen Yao[1], Xiaoqian Chen[1], and Fred van Keulen[2]

[1] National University of Defense Technology, Changsha, China
gfkdzhangyi@126.com
[2] Delft University of Technology, Delft, The Netherlands
A.vanKeulen@tudelft.nl

Abstract. Support vector regression (SVR) is a common surrogate model for computationally expensive simulation. It is able to balance the model complexity and the error tolerance. Whether SVR interpolates the training samples is dependent on its parameters. For the nonlinear function approximation without noise, when SVR is not an interpolator, it is advisable to model the errors and use them to compensate the prediction response. In this paper, the errors of SVR are modeled by using Gaussian process, and the final model response is obtained by the combination of SVR and the Gaussian process of the errors. The numerical experiments show the proposed method is able to further improve the prediction accuracy of SVR.

Keywords: Support vector regression · Gaussian kernel · Error modeling · Gaussian process

1 Introduction

Surrogate modeling is a technique used to replace computer simulation which is complex enough or computationally costly. In the past several years, intensive research on surrogate methods has been carried out, including polynomial response surface [5], radial basis functions [9], support vector regression [7] and Kriging [6]. Among these surrogates, SVR has gained wide attention because it shows remarkable prediction ability [1–3].

SVR is able to balance the model complexity and the error tolerance. Whether SVR interpolates the training samples is dependent on its parameters. For the nonlinear function approximation without noise, when SVR is not an interpolator, it is advisable to model the errors and use them to compensate the prediction response. Hombal and Mahadevan modeled the errors of the physics-based model [4], but their method is only suitable for model selection. In this paper, we try to use Gaussian process to model the errors of SVR and use them to compensate the prediction response. First, nonlinear SVR is built and its parameters are updated manually. Then, the errors of SVR are modeled by using Gaussian process. Finally, the actual model response is obtained by the

A. Schumacher et al. (eds.), *Advances in Structural and Multidisciplinary Optimization*,
https://doi.org/10.1007/978-3-319-67988-4_20

combination of SVR and the Gaussian process of the errors. Numerical experiments show that the proposed method is able to further improve the prediction accuracy of SVR.

The remainder of the paper is organized as follows. Section 2 describes our method, and Sect. 3 shows the numerical experiments and results. Finally, the conclusion appears in Sect. 4.

2 The Proposed Method

2.1 Support Vector Regression

Considering linear SVR with n training samples $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$ and the model responses $y_1, y_2, ..., y_n$

$$\begin{aligned} \min \quad & \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{n}\sum_{i=1}^{n}(\xi_i + \xi_i^*) \\ \text{s.t.} \quad & \begin{cases} y_i - \mathbf{w}^{\mathrm{T}}\mathbf{x}_i - \mu \le \varepsilon + \xi_i \\ \mathbf{w}^{\mathrm{T}}\mathbf{x}_i + \mu - y_i \le \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0 \end{cases} \end{aligned} \tag{1}$$

where μ and $\mathbf{w}$ are respectively the bias term and weight coefficients of the model, ε are the error limit which can be tolerated, ξ_i and ξ_i^* are the slack variables, C is the regularization parameter to balance the model complexity and errors.

The constrained optimization problem of Eq. (1) is solved using Lagrangian function

$$\begin{aligned} L = & \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{n}\sum_{i=1}^{n}(\xi_i + \xi_i^*) - \sum_{i=1}^{n}(\eta_i\xi_i + \eta_i^*\xi_i^*) \\ & - \sum_{i=1}^{n}\alpha_i(\varepsilon + \xi_i - y_i + \mathbf{w}^{\mathrm{T}}\mathbf{x}_i + \mu) - \sum_{i=1}^{n}\alpha_i^*(\varepsilon + \xi_i^* + y_i - \mathbf{w}^{\mathrm{T}}\mathbf{x}_i - \mu) \end{aligned} \tag{2}$$

where $\eta_i, \eta_i^*, \alpha_i$ and α_i^* are Lagrange multipliers. The minimization of L with respect to $\mathbf{w}, \mu, \xi$ and ξ^* is obtained

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{n}(\alpha_i - \alpha_i^*)\mathbf{x}_i = 0 \tag{3}$$

$$\frac{\partial L}{\partial \mu} = \sum_{i=1}^{n}(\alpha_i - \alpha_i^*) = 0 \tag{4}$$

$$\frac{\partial L}{\partial \xi_i} = \frac{C}{n} - \alpha_i - \eta_i = 0 \tag{5}$$

$$\frac{\partial L}{\partial \xi_i^*} = \frac{C}{n} - \alpha_i^* - \eta_i^* = 0 \tag{6}$$

From Eq. (3)

$$\mathbf{w} = \sum_{i=1}^{n} (\alpha_i - \alpha_i^*)\mathbf{x}_i \tag{7}$$

Substituting Eqs. (3)–(6) into Eq. (2), linear SVR is expressed as

$$y_{\mathrm{SVR}}(\mathbf{x}) = \mu + \sum_{i=1}^{n} (\alpha_i - \alpha_i^*)\mathbf{x}_i^{\mathrm{T}}\mathbf{x} \tag{8}$$

Replacing the dot product $\mathbf{x}_i^{\mathrm{T}}\mathbf{x}$ with a kernel function $\boldsymbol{\phi}(\mathbf{x}_i, \mathbf{x})$, we can get nonlinear SVR

$$y_{\mathrm{SVR}}(\mathbf{x}) = \mu + \sum_{i=1}^{n} (\alpha_i - \alpha_i^*)\boldsymbol{\phi}(\mathbf{x}_i, \mathbf{x}) \tag{9}$$

The most widely used nonlinear kernel function is Gaussian kernel

$$\boldsymbol{\phi}(\mathbf{x}_i, \mathbf{x}) = \exp\Big(- \frac{\|\mathbf{x}_i - \mathbf{x}\|^2}{2\sigma^2}\Big) \tag{10}$$

where σ is the Gaussian kernel parameter.

2.2 Gaussian Process

Gaussian process (GP) regards the real model responses as random variables that are generated from a joint Gaussian distribution [8]. GP is defined as

$$\mathbf{y}_t \sim \mathrm{GP}(\mathbf{F}\boldsymbol{\beta}, \mathbf{C}) \tag{11}$$

$$E[(\mathbf{y}_t - \mathbf{F}\boldsymbol{\beta})(\mathbf{y}_t - \mathbf{F}\boldsymbol{\beta})^{\mathrm{T}}] = \mathbf{C} \tag{12}$$

where $\mathbf{y}_t$ is the true model responses, $\mathbf{F}$ is the regression function matrix of the training samples, and $\boldsymbol{\beta}$ is the regression coefficient vector. In the actual engineering applications, the Gaussian correlation function is often used

$$\mathbf{C}_{ij} = \hat{\sigma}^2 R(\boldsymbol{\theta}, \mathbf{x}_i, \mathbf{x}_j) = \hat{\sigma}^2 \prod_{d=1}^{m} \exp(-\theta^d (x_i^d - x_j^d)^2) \tag{13}$$

where $\hat{\sigma}^2$ is the process variance, and $\boldsymbol{\theta} = [\theta^1, \theta^2, ..., \theta^m]^{\mathrm{T}}$ is the correlation parameter. Commonly, after $\mathbf{F}$ is determined, $\boldsymbol{\beta}$, $\hat{\sigma}^2$ and $\boldsymbol{\theta}$ can be estimated by using maximal likelihood estimation.

After the derivation, the prediction of GP for a new sample $\mathbf{x}$ is

$$y_{\mathrm{GP}}(\mathbf{x}) = \mathbf{f}^{\mathrm{T}}\boldsymbol{\beta} + \mathbf{r}^{\mathrm{T}}\mathbf{R}^{-1}(\mathbf{y}_t - \mathbf{F}\boldsymbol{\beta}) \tag{14}$$

where $\mathbf{f} \in \mathbb{R}^p$ is the regression function vector of $\mathbf{x}$, and $\mathbf{r}$ is the correlation vector between $\mathbf{x}$ and the training samples

$$\mathbf{r} = [R(\boldsymbol{\theta}, \mathbf{x}_1, \mathbf{x}), R(\boldsymbol{\theta}, \mathbf{x}_2, \mathbf{x}), ..., R(\boldsymbol{\theta}, \mathbf{x}_n, \mathbf{x})] \tag{15}$$

2.3 Combination of SVR and GP

Assuming that there is no noise in the training samples, when SVR is unable to interpolate all the training samples, the training errors appear. We try to model the errors by using Gaussian process. The parameters of Gaussian process are automatically estimated by using maximal likelihood estimation. Then, the Gaussian process of the errors and SVR are added to get the final response

$$\hat{y}(\mathbf{x}) = y_{\mathrm{SVR}}(\mathbf{x}) + y_{\mathrm{GP}}(\mathbf{x}) \tag{16}$$

In this way, $\hat{y}(\mathbf{x})$ is still an interpolater at the training samples because Gaussian process interpolates the training errors.

3 Numerical Experiments

3.1 Benchmark Functions

1. Branin-Hoo function

$$\begin{aligned} y(\mathbf{x}) &= \left(x_2 - \frac{5.1x_1^2}{4\pi^2} + \frac{5x_1}{\pi} - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10 \\ x_1 &\in [-5, 10],\ x_2 \in [0, 15] \end{aligned} \tag{17}$$

2. Friedman function

$$\begin{aligned} y(\mathbf{x}) &= 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 \\ x_i &\in [0, 1] \end{aligned} \tag{18}$$

3.2 Designs of Experiments

Latin hypercube sampling (LHS) is used to generate the training and testing samples. The training and testing samples are obtained by maximizing the minimum distance between design samples with 20 iterations. The number of the training and testing samples for the benchmark functions are displayed in Table 1.

Table 1. The number of the training and testing samples for the benchmark functions

	Training	Testing
Branin-Hoo function	12	2000
Friedman function	42	2000

3.3 Results

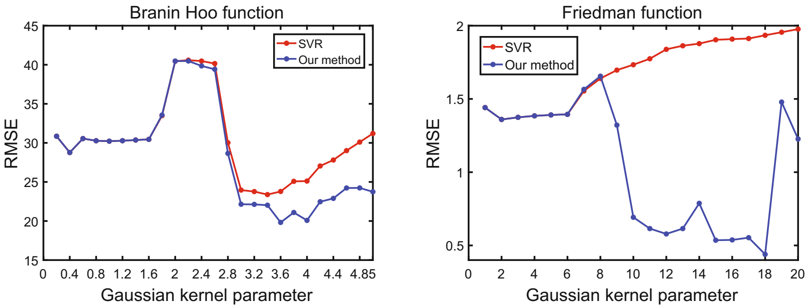
In the numerical experiments, ε is set as 1×10^{-4} and C is set as 1×10^{8}. A constant is used as the regression function of GP, and genetic algorithm is used to search for the optimal solution of maximal likelihood estimation. The prediction accuracy is evaluated by the root mean square error (RMSE), which is defined as follows

$$\mathrm{RMSE} = \sqrt{\frac{1}{n_{\mathrm{test}}} \sum_{i=1}^{n_{\mathrm{test}}} e_i^2} \tag{19}$$

where e_i is the error between the prediction and the actual response in the i-th test sample, and n_{test} is the number of the test samples.

For the Branin Hoo function, the Gaussian kernel parameter increases from 0.2 to 5 by 0.2. In Fig. 1(a), it can be observed that with the increase of the Gaussian kernel parameter from 0.2 to 2.2, SVR is still an interpolater and the training errors are so small, so the Gaussian process response of the errors are also so small and do not have an effect on the prediction accuracy. However, with the increase of the Gaussian kernel parameter from 2.2, SVR is not an interpolater any more and the Gaussian process response of the errors will influence the prediction accuracy.

For the Friedman function, the Gaussian kernel parameter increases from 1 to 20 by 1. In Fig. 1(b), it can be observed that with the increase of the Gaussian kernel parameter from 6 to 20, the RMSE of SVR increases. The RMSE of our method begins to decrease from the Gaussian kernel parameter of 8 and perform better than SVR.



(a)

(b)

Fig. 1. Comparison of RMSE of different kernel parameters for the benchmark functions

4 Conclusion

In this paper, Gaussian process is used to model the errors of SVR, and the final response of the proposed method is the combination of SVR and the Gaussian process of the errors. When SVR does not interpolate the training samples, the Gaussian process of the errors will have an effect and further improve the prediction accuracy of SVR. When SVR interpolates the training samples, the Gaussian process of the errors will not influence the prediction accuracy of SVR. Thus, it is advisable to use Gaussian process to model the errors and enhance the prediction ability of SVR.

References

1. Clarke, S.M., Griebsch, J.H., Simpson, T.W.: Analysis of support vector regression for approximation of complex engineering analyses. J. Mech. Des. **127**(6), 1077–1087 (2005)
2. Forrester, A., Sobester, A., Keane, A.: Engineering Design via Surrogate Modelling: A Practical Guide. Wiley, Chichester (2008)
3. Friedman, J.H.: Multivariate adaptive regression splines, The annals of statistics, 1991
4. Hombal, V., Mahadevan, S.: Model selection among physics-based models. J. Mech. Des. **135**(2), 021003 (2013)
5. Jin, R., Chen, W., Simpson, T.W.: Comparative studies of metamodelling techniques under multiple modelling criteria. Struct. Multidiscip. Optim. **23**(1), 1–13 (2001)
6. Martin, J.D., Simpson, T.W.: Use of kriging models to approximate deterministic computer models. AIAA J. **43**(4), 853–863 (2005)
7. Smola, A.J., Schölkopf, B.: A tutorial on support vector regression. Stat. Comput. **14**(3), 199–222 (2004)
8. Williams, C.K., Rasmussen, C.E.: Gaussian Processes for Machine Learning, vol. 2, No. 3, p. 4. the MIT Press (2006)
9. Yao, W., Chen, X., Zhao, Y., van Tooren, M.: Concurrent subspace width optimization method for RBF neural network modeling. IEEE Trans. Neural Netw. Learn. Syst. **23**(2), 247–259 (2012)