

A Gaussian process regression model based on Bootstrap

Yong Zeng*

School of Science, Northeast Electric Power university, Jilin, China, 132011

*Corresponding author: cun_zeng@163.com

Abstract. Gaussian process regression is a commonly used non-parametric regression model for the analysis of high-dimensional data. However, due to the complexity of the data fitting curve, the model may be overfitting or underfitting. Based on this, this paper proposes a Gaussian process regression model based on the Bootstrap method (BGP method) to alleviate the overfitting or underfitting phenomenon of a single Gaussian process regression model. Specifically, on the one hand, the proposed method can balance the training error and test error of the prediction model, and make the model have high robustness while reducing the fluctuation level of the estimator. On the other hand, the method uses the weighted variance of the posteriori distribution to give the response variable an uncertainty measure. A large number of simulation experiments show that the BGP method has the best accuracy and robustness compared with several commonly used regression methods, such as LASSO regression, nonlinear support vector machine regression, decision tree regression and random forest regression. Finally, the empirical analysis results on the meat dataset with ultra-high-dimensional characteristics show that the BGP method still has high applicability for high-dimensional data scenarios. In addition, the BGP model can be further extended to the development of kernel types and hyperparameter determination strategies.

Keywords: High-dimensional data, model average, Gaussian process regression, combinatorial kernel function.

1. Introduction

In the era of big data, with the increase of information collection capacity, more and more complex relationships between high-dimensional data need to be effectively processed in order to carry out prediction tasks in the case of high-dimensional data. For example, crawl more than 131913 posts in the EJMR forum from 2014 to 2016, with a total of 1143416 user replies. The open-source dictionary is used to screen the 10,000 words with the highest frequency and establish a word document matrix. There are four levels based on the terms that are used to screen the criteria for the presence of gender discrimination. The input variables of the real dataset used in this paper are 100-dimensional, which also belongs to ultra-high-dimensional data. Moreover, the relationship between the 100-dimensional data and the three response variables is not clear in advance.

When there is a linear relationship between the input variable X and the response variable Y , parametric (linear) regression models with penalties or linear models with generalized linear models or linear models with multiplication can handle the prediction tasks of such cases, e.g., Rahim Alhamzawi, Haithem Taha Mohammad Ali (2018) [1], et al. The adaptive LASSO regression model based on full Bayesian processing and the classical adaptive LASSO regression model were tested. However, it is worth noting that the relationship between the input variables and the response variables is often nonlinear. For example, Pavel and Serhan (2019) [2] et al. studied the robustness of nonparametric regression methods based on kernel estimation on nonlinear datasets. WANG Xia, FU Zhoughao (2020) [3] et al. constructed financial contagion statistics based on nonparametric regression, which can capture various forms of nonlinear contagion effects. Based on the dataset established by finite element analysis of ball indentation, JIN Hongping et al. (2023) [4] used the K-nearest neighbor nonparametric regression model to estimate the elastic modulus of indentation. It is well known that common classical nonparametric statistical methods, such as nonparametric regression, tend to have slow convergence when processing high-dimensional data. For example, in the continuous regression model, SHEN Jia, WU Mingxin et al. [5] studied that the kernel estimation

supernorm rate is $O(\ln_k T (\ln T/T)^{2/5})$ $k \geq 1$ under the classical canonical condition, and this rate can be sharpened under the local irregular condition.

Based on this, this paper focuses on the analysis tools in the Bayesian framework. As a Bayesian method that can measure the uncertainty of the predicted value by the variance of the posterior distribution, the Gaussian process regression model can effectively model complex nonlinear systems. Specifically, the Gaussian process can be regarded as an infinite-dimensional generalization of the multivariate Gaussian distribution, and the nonlinear relationship between the input variables and the response variables can be constructed by using the regenerative kernel Hilbert theory, and it is not affected by the eigendimension in the modeling process. For example, Nhat-Duc Hoang (2016) [6] et al. used a Gaussian regression model to process input factors (low-dimensional data) of eight dimensions, such as cement, fine aggregate, and water, to estimate the compressive strength of high-performance concrete. Dongdong Kong et al. (2018) [7] studied the low-dimensional input factors such as wear width, wear area, wear state and breakage, and Gaussian process regression predicted tool wear. Bo Wang and Aiping Xu (2019) [8] proposed a Gaussian process method for nonparametric functional regression with scalar response or functional response and conducted an empirical analysis.

However, when the Gaussian process regression is faced with the case of more dimensions than the number of samples or the case of ultra-high dimensions, the model may be overfitting or underfitting due to insufficient sample size. Based on this, this paper uses the Bootstrap method to sample multiple times, form multiple GP models and make model average prediction, so as to alleviate the impact of overfitting or underfitting of a single GP model. Specifically, this paper uses the Bootstrap method to extract different subsamples and obtain different GP models. Since each GP process will obtain a posterior distribution estimate, and the Gaussian process can provide a measure of the uncertainty of the model prediction for the maximum posterior variance estimation, the smaller the variance and the smaller the uncertainty, the more confident we are in the model and the higher the confidence of the posterior estimate. Therefore, the normalized vector of the reciprocal variance is used as the linear weighted weight. The innovation of this paper is that on the one hand, it effectively alleviates the problems of overfitting and underfitting Gaussian processes in the context of special data, and on the other hand, it provides an uncertainty measure for the predicted value by using the weighted combination of posterior distributions, which is a property that models such as LASSO regression, nonlinear support vector machine regression, decision tree regression and random forest regression do not have. A large number of simulation experiments show that compared with the traditional regression model, the robustness and accuracy of the GP method improved by Bootstrap are significantly improved (about 51.36%~99.91%) compared with the traditional regression model.

2. Theories and Methods

2.1. Gaussian process regression

This section provides an overview of Gaussian process regression. For the sake of understanding, we only consider one-dimensional capacity and period data, and for cases where both input and response variables are scalars. With analysis, it is easy to extend the study to multidimensional inputs. To begin with, the probability distribution of the Gaussian process (GP) definition function is:

$$f(x) \sim \mathcal{GP}(m(x), \mathcal{K}(x, x')) \quad (1)$$

where $f(x)$ is the probability density function of the Gaussian process, and x is the input variable. $m(x)$ is the mean function; $\mathcal{K}(x, x')$ stands for standard covariance function.

$$\begin{cases} m(x) = \mathbb{E}[f(x)] \\ \mathcal{K}(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))^T] \end{cases} \quad (2)$$

For the finite point set $X = x_1, x_2, \dots, x_N$, this procedure defines $p(f(x_1), \dots, f(x_N))$ as a joint Gaussian. At the same time, this process is determined by the mean function $m(x)$ and the covariance function $K(x)$ given by $K_{ij} = \mathcal{K}(x_i, x_j)$.

Gaussian process regression is a non-parametric regression method that does not assume the form of the function and does not estimate the parameters from it. Let's assume that the function $f(x)$ comes from a sample of the Gaussian process. The covariance function generally chooses the square index (SE):

$$\mathcal{K}_{SE}(x, x') = \theta_f^2 \exp\left(-\frac{1}{\theta_l^2} \|x - x'\|^2\right) \quad (3)$$

where θ_f and θ_l are the parameters of the covariance function $K(x)$, which controls the scaling of the response variable Y and the input variable X , respectively.

2.2. Gaussian process regression based on Bootstrap

Bootstrap is a non-parametric Monte Carlo method, which requires "re-sampling" of observational information to statistically infer the distribution characteristics of the population. Since the mean estimate $E_\theta(\hat{\theta})$ of Bootstrap is numerically equal to the estimate θ , i.e. Bootstrap is an unbiased estimate. The variance $\frac{n}{n-1} \sigma^2$ obtained by Bootstrap is more valid than that of point estimation $\frac{\sigma^2}{n}$ ($n \neq 1$). As a result, Bootstrap may improve the robustness of the primary GP and be able to disperse errors.

2.1 For single-sample prediction, the model in this paper is established as follows:

$$\begin{cases} \hat{f}(x^*) = \sum_{k=1}^K w_k \times GP_k(x^*, \theta) \\ w_k = (1/\sigma_k^2) / \sum_{k=1}^K \frac{1}{\sigma_k^2} \end{cases} \quad (4)$$

where $\hat{f}(x^*)$ is the predicted value obtained by the k-th Gaussian process (GP_k), and x^* is the feature of the input sample. ; θ is the combinatorial nucleus; σ_k^2 is the variance obtained for each Gaussian process GP_k ; w_k is the weight of the k-th Gaussian process.

When the number of samples is n, the model is modified as follows:

$$\begin{cases} \hat{f}(x) = \sum_{k=1}^K w_k \times GP_k(x, \theta) \\ w_k = (1/\sigma_k^2) / \sum_{k=1}^K \frac{1}{\sigma_k^2} \end{cases} \quad (5)$$

Before generating K models, sub-samples need to be randomly selected, and the features and response variables of the sub-samples should be extracted for training the model. Each time the size subsample features are extracted and put into a model, a total of size $\times K$ subsample features need to be inputted.

$$x \longrightarrow \{x_{sub}\}_{sub=1}^K \quad (6)$$

Since each model is considered a Gaussian process, each model is obtained by the following formula

$$GP_{sub} = GP(x_{sub}, y_{sub}) \quad (7)$$

At the same time, each model obeys a normal distribution, i.e., $GP_{sub} \sim N(\mu_{sub}, \sigma_{sub}^2)$

A total of K models need to be generated, and each model is regarded as a Gaussian process, and each Gaussian process is trained. The results obtained by the final K models are averaged by the model, and the formula for model averaging is as follows:

$$\hat{y} = \sum_{sub=1}^K \frac{1}{\sigma_{sub}^2} \times \mu_{sub} \quad (8)$$

The general model of Gaussian process regression based on Bootstrap consists of five basic elements, which are:

(1) Firstly, the simulation dataset and the meat dataset are divided into validation set and training set according to the ratio of 2:8. And determine the number of models that need to be trained K (the value in the simulation experiment is 35).

(2) The size subsamples were randomly generated, their feature x and response variables y_sub were extracted, and the kth Gaussian process model GP_k was trained with the feature and response variables.

(3) The subsample features in the test set are x_test put into the trained Gaussian process model GP_k , and there are K Gaussian process models.

(4) Each model has a corresponding variance σ_k^2 and a predicted y_pred.

(5) $1/\sigma_k^2$ is taken as the weight of the kth model, and the weight vector and the predicted value vector are multiplied by matrix. The predicted value of the model averaged $\hat{f}(x)$.

The Gaussian process regression model based on Bootstrap needs to treat each process as a Gaussian process, and then with the help of Bootstrap's idea to sample samples many times, combined with the characteristics of Gaussian processes, the weights of each Gaussian process are determined, and the prediction results obtained by each Gaussian process are averaged with the corresponding weights, that is, the model is averaged.

3. Data analysis

3.1. Simulator investigation

In the simulation data analysis, six different connection functions are used to construct the relationship between the input variables and the corresponding variables. 300 simulation experiments were performed for each function, the number of simulated samples n was 600, and 80% of the simulation dataset was selected as the training set and 20% as the test set. This simulation experiment uses the squared-exponential kernel in the sklearn library. The kernel is parameterized by the length scale parameter $l > 0$. The kernel is given by:

$$k(x_i, x_j) = \exp\left(-\frac{d(x_i, x_j)^2}{2l^2}\right) \quad (9)$$

Among them, the core length scale $l = 1.0$ is taken, and the float is limited to (1e-3, 1e3).

The program runs on Anaconda3 based on Python 3.11.3 64-bit; Processor: 11th Gen Intel(R) Core(TM) i5-11300H @ 3.10GHz 3.11GHz; System type: x64-based processor; Windows specification: Window10. Since the functions in the simulation experiment are trigonometric

functions, exponential functions, inverse trigonometric functions, power functions, logarithmic functions, etc., the GP method is compared with four models, such as Lasso [1], Decision tree [9], SVM regresiong [10], and Random Forest [9]. Under each function, the mean and standard deviation of the mean square error between the predicted value and the true value were used as the criterion for judging the model, and 300 simulation experiments were carried out. Models 1 through 6 are as follows:

$$\begin{aligned}
 \text{Model1: } & \sin\left(\frac{\pi}{2}x_1\right) + \cos\left(\frac{2\pi}{3}x_2\right) + 2x_3 + \varepsilon \\
 \text{Model2: } & \exp(-0.3 \times x_1) + \tan(x_2) + 5x_3 + \varepsilon \\
 \text{Model3: } & \ln\left(\frac{x_1}{10}\right) + 9 \times x_2 + \arctan(3x_3) + \varepsilon \\
 \text{Model4: } & \sin(x_1) + \arctan(10 \times x_2) + 7x_3 + \varepsilon \\
 \text{Model5: } & \ln(10x_1) + \cos\left(\frac{x_2}{2}\right) + \exp(3x_3) + \varepsilon \\
 \text{Model6: } & \exp(x_1) + 5 \times x_2 + \arctan(3x_3) + \varepsilon
 \end{aligned} \tag{10}$$

where x_i ($i = 1, 2, 3$) obeys a uniform distribution, while ε obeys a normal distribution with a mean value of 0 and a standard deviation of 0.01.

The mean squared error is calculated according to the following formula:

$$MSE = \frac{1}{K} \sum_{k=1}^K \omega_k (y_k - \hat{y}_k)^2 \tag{11}$$

where w_k is the weight corresponding to the k-th Gaussian process; y_k represents the true value of the k-th Gaussian process; $\hat{y}_k$ denotes the predicted value of the k-th Gaussian process. The results are shown in Table 1:

Table.1. The mean error of multiple models

Method	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6
LASSO	0.3565 (0.0846)	0.2594 (0.0690)	0.6590 (0.3137)	0.2858 (0.0692)	0.5053 (0.2701)	0.3704 (0.866)
Decision Tree	0.0727 (0.0190)	0.0620 (0.0163)	0.4811 (0.2366)	0.1175 (0.0400)	0.1328 (0.1171)	0.1557 (0.0361)
SVM(kernel)	0.0084 (0.0051)	0.0240 (0.0137)	0.3123 (0.2297)	0.0612 (0.0383)	0.1572 (0.1602)	0.0277 (0.0183)
RF	0.0278 (0.0086)	0.0282 (0.0088)	0.2533 (0.1589)	0.0648 (0.0237)	0.0605 (0.0708)	0.0675 (0.0164)
GP method	0.0001 (2.8e-5)	0.0002 (4.2e-5)	0.0761 (0.1067)	0.0017 (0.0011)	0.0646 (0.0931)	0.0001 (3.2e-5)

The values in parentheses represent the standard deviation and the values outside the parentheses represent the mean square error.

In the simulation experiment, the mean square error was used as the evaluation criterion to calculate the relative accuracy of GPmethod compared with the four methods, and the accuracy of GPmethod was increased by 88.42%~99.97% compared with LASSO. Compared with the decision tree, it is improved by 51.36%~99.91%; Compared with SVM (kernel), it is increased by 58.91%~99.52%; Compared with random forest, it is improved by 69.97%~99.80%. Compared with other traditional methods, the GP method is mainly improved by the nonlinear model in the

experiment, while the LASSO is a linear regression model; Although the decision tree is a nonlinear model, its smoothness is not enough. Random forests are smoother than decision trees, so the effect is not too bad; The kernel uses nonlinear kernel functions, but the decision functions are not smooth enough to generate decision boundaries. Compared with these methods, the GP method not only uses a nonlinear kernel function, but also is smooth enough.

The mean square error and standard deviation of the GP method are the smallest in the experiment, and the simulation results show that compared with the traditional regression model, the mean square error and standard deviation of the GP method are significantly lower than those of the traditional regression model, and the robustness and accuracy of the GP method improved by Bootstrap are significantly improved.

3.2. Real data analysis

The meat dataset used in this empirical analysis comes from the website: <http://lib.stat.cmu.edu/datasets/tecator>. The sample dimension in the meat dataset is 100 dimensions, which is ultra-high-dimensional data. The response variables were fat, water, and protein. In order to ensure the robustness and effectiveness of the method, the experiment will be repeated 200 times for each actual data verification, 80% of the 215 samples will be selected as the training set (the remaining 20% will be used as the test set), and size=50 samples will be randomly selected each time. A total of 25 models were generated for model averaging to obtain the final results.

Due to the unclear relationship between the true models of real data, the GP method is not as good as LASSO for the true model with a strong linear relationship, but the GP method still performs better for the true model with a strong nonlinear relationship. The experimental results are shown in Table 2:

Table.2. The mean error of the three response variables

Response variable	Protein	Water	Fat
LASSO	7.8901 (1.6523)	17.3350 (4.0722)	23.5135 (5.1846)
Decision Tree	6.8224 (1.9470)	56.8617 (18.9728)	98.1721 (32.0580)
SVM(kernel)	8.0904 (1.6970)	79.4633 (18.0380)	137.6901 (29.3922)
RF	4.2941 (0.9629)	32.6866 (9.4887)	55.8430 (15.7409)
GP method	2.4173 (0.6841)	28.6802 (16.7242)	40.2254 (23.8948)

4. Conclusions

The trend of regression model in the face of massive data provides a basis for the establishment of nonlinear model prediction, while the traditional regression model may have poor fitting effect due to high dimension or large sample size of sample data. On the other hand, there may be a situation where the sample size of the input is too large, and the overfitting effect is significant. Therefore, this paper proposes a Gaussian process regression model based on Bootstrap for the regression prediction task with small sample size and ultra-high sample dimension, which has good robustness and resistance to overfitting. The model can effectively reduce the fluctuation of the estimator and balance the training error and the test error by screening the sample size.

5. Promotion

The results of a large number of simulation experiments and empirical analysis in this paper show that compared with several regression prediction methods such as LASSO and SVM(kernel), the

mean square error and standard deviation of the GP method are the smallest, and the reliability of the prediction results is obviously the highest. Therefore, the Gaussian regression model based on Bootstrap has certain practical application value. The Gaussian regression method proposed in this paper is based on the Bootstrap method, but it is only used for sample selection for screening. Therefore, the model can also be extended to several aspects, such as using the Bootstrap method for the selection of model variables, using the Bootstrap method for kernel filtering, and using the Bootstrap method for the selection of hyperparameters in the kernel, and the Bootstrap method can also be used in the above three aspects at the same time.

References

- [1] Rahim Alhamzawi and Haithem Taha Mohammad Ali. 2018. The Bayesian adaptive lasso regression, *Mathematical Biosciences*, 303, 75-82. <https://doi.org/10.1016/j.mbs.2018.06.004>
- [2] Pavel Čížek and Serhan Sadıkoğlu. 2019. Robust nonparametric regression: A review, *WIREs Computational Statistics*, 12(3). <https://doi.org/10.1002/wics.1492>
- [3] WANG Xia, Fu Zhong-hao, HONG Yong-miao and ZHANG Dong-yue. 2020. Nonparametric regression based testing for financial contagion. *Systems Engineering – Theory & Practice* (06), 1398-1418.
- [4] JIN Hong-ping. 2023. Estimation of Indetation Elastic Modulus Based on K-nearest Neighbor Nonparametric Regression. *Journal of Hubei University of Automotive Technology* (04), 76-80
- [5] SHEN Jia, WU Ming-xin. 2003. Uniform Quadric Error of the Estimator of Nonparametric Regression in Continuous Time Process. *Journal of Fudan University(Natural Science)*(02), 234-239. [doi:10.15943/j.cnki.fdxh-jns.2003.02.018](https://doi.org/10.15943/j.cnki.fdxh-jns.2003.02.018).
- [6] Nhat-Duc Hoang, Anh-Duc Pham, Quoc-Lam Nguyen, Quang-Nhat Pham, 2016. Estimating Compressive Strength of High Performance Concrete with Gaussian Process Regression Model, *Advance in Civil Engineering*. 1-8. <https://doi.org/10.1155/2016/2861380>
- [7] Dongdong Kong, Yongjie Chen, Ning Li. 2018. Gaussian process regression for tool wear prediction. *Mechanical Systems and Signal Processing*, 104, 556-574. <https://doi.org/10.1016/j.ymssp.2017.11.021>
- [8] Bo Wang and Aiping Xu. 2019. Gaussian prcess methods for nonparametric functional regression with mixed predictors. *Computational Statistics & Data Analysis*, 131, 80-90. <https://doi.org/10.1016/j.csda.2018.07.009>
- [9] Charbuty, B., & Abdulazeez, A. 2021. Classification Based on Decision Tree Algorithm for Machine Learning. *Journal of Applied Science and Technology Trends*, 2(01), 20-28. <https://doi.org/10.38094/jastt20165>
- [10] Roman, I., Santana, R., Mendiburu, A. et al. 2021. In-depth analysis of SVM kernel learning and its components. *Neural Comput & Applic.* 33, 6575–6594. <https://doi.org/10.1007/s00521-020-05419-z>