A High-Dimensional Data Regression Model Based on Stacking-Gaussian Process Algorithm

Yizhe Feng [1, *], Yongqi An [2]

[1] School of Mathematics and Big Data, Jining University, Qufu, China, 273100

[2] School of Mathematics and Big Data, Dezhou University, Dezhou, China, 253023

* Corresponding Author Email: fyzjnxy00457@163.com

Abstract. In order to mine valuable information and potential rules from a large amount of data, processing high-dimensional data and establishing regression prediction models has become a hot research topic in the field of machine learning. High-dimensional input and nonlinear relationship fitting are two common problems in establishing regression models. Based on this, this paper proposes a high-dimensional data regression model based on a stacking framework and Gaussian process algorithm. Specifically, on the one hand, the stacking framework is used to alleviate the curse of dimensionality problem caused by high input dimensions. On the other hand, based on different kernel functions, the Gaussian process regression model can fit linear or nonlinear connection functions of different complexity. Simulation experiments and real data analysis show that the proposed method has smaller prediction errors, better prediction accuracy, and good generalization performance compared to some contrastive methods such as Random Forests, Lasso Regression, and Back Propagation Neural Network, demonstrating the superiority of the proposed model.

Keywords: stacking-Gaussian process algorithm, high-dimensional data, regression prediction model, Bagging algorithm.

1. Introduction

In the era of big data, with the rapid development of information technology and the diversification of data acquisition methods, a large number of high-dimensional data sets have been generated. The processing and analysis of high-dimensional data sets have important applications in the fields of bioengineering, digital medical care, aerospace, etc. However, in the regression model fitting of high-dimensional features, the simple integration of different feature data cannot guarantee the effectiveness of the mining task, and the phenomenon of the curse of dimensionality will occur. The goal of this paper is to mine the effective information for predicting the response variable in high-dimensional features and find an appropriate way to establish a regression model between high-dimensional feature variables and scalar responses.

In high-dimensional data sets, it is often the case that the feature dimension or attribute dimension of the data result sample is much larger than the number of samples, which will lead to serious overfitting or the curse of dimensionality in statistical learning. At the same time, since it is impossible to determine whether the data in the real-world problem scenario is a linear relationship, there may be complex nonlinear relationships. Firstly, many scholars have conducted research on high-dimensional feature variables. For example, Bai Yongxin, Qian Manling, and Tian Maozai[1] conducted in-depth research on the partially linear additive quantile regression model in the context of the rapid expansion of covariate dimensions to ultra-high dimensions, and demonstrated its wide applicability and high efficiency in the application of similar models through experiments. Fu Manxia and Zhou Shuisheng[2] proposed a non-parametric additive regression model based on fused Lasso. This model cleverly integrates the advantages of L1 penalty and L2 penalty. The introduction of the fused Lasso penalty not only accelerates the calculation process of the model but also endows the model with the ability of local adaptation, making the prediction of quantiles and even extreme quantiles more accurate. This innovative method provides a new perspective and solution for dealing with the problem of nonlinear relationships in high-dimensional data. Li Xiaoguang [3] turned the

research perspective to the issue of the overall coefficient test of high-dimensional linear models and proposed a heritability estimator based on the multivariate ridge regression estimation of the error covariance matrix. This estimator shows excellent accuracy in dealing with the heritability estimation of multiple phenotypic linear combinations, and can give reliable estimation results regardless of whether the coefficients are sparse or dense. In addition, a test statistic based on the overall coefficient estimation is innovatively proposed, and its asymptotic distribution characteristics under the null hypothesis and local alternative hypotheses are deeply discussed. This research result not only enriches the content of the test theory of high-dimensional linear models but also proves the wide validity of the proposed test statistic when the dimension of redundant parameters changes. Huang Wenjing, Deng Dan, Du Jielin[4] and others proposed a new method to optimize the sparse logistic regression model by using the graph structure information between predictor variables. Secondly, for the fitting of nonlinear relationships. Zhang Zhao and Ren Junsheng[5] studied a non-parametric modeling scheme for an underwater vehicle, aiming to achieve continuous-time dynamics modeling. It uses the local weighting strategy to construct a local model to localize the Gaussian process learning, and at the same time designs an adaptive moment estimation method based on the innovative gradient to automatically adjust the hyperparameters of the Gaussian process. Nur Chamidah, Budi Lestari, and I Nyoman Budiantara[6] theoretically discussed an estimation method for a non-parametric regression model with two or more response variables and predictor variables, and used the correlation between the response variables in the mixed estimator. Zhu Nenghui[7] and others combined the robust loss function, B-spline approximation, and adaptive group Lasso to study a high-dimensional additive model to identify insignificant covariates in the "large p, small n" situation. In previous studies, we can observe that most of the results present methods and approaches to solve the fitting of nonlinear problems and the processing of high-dimensional data. In addition, there is also the use of the Random Forest method to solve the problem of predicting the concentration of calcium ions in seawater circulating water [8], the construction of a Back Propagation Neural Network based on non-probabilistic samples to infer the population [9] to deal with similar problems.

In summary, this paper combines the stacking framework and Gaussian process regression to form a new machine learning framework using double-layer Gaussian process regression. Specifically, on the one hand, by placing several Gaussian process regression base models in the first layer of the stacking framework and training them with samples obtained by bootstrap sampling, while balancing the bias and variance of the prediction model, the impact of the curse of dimensionality is also alleviated; on the other hand, using the prediction results of several base models in the first layer of the stacking framework as new input variables, the Gaussian process regression model is constructed again to improve the prediction accuracy and the fitting of the unknown connection structure of the overall model. In addition, the variance of the prediction distribution generated by the second-layer Gaussian process regression can be directly used as a tool for quantifying the uncertainty of the prediction value. Finally, simulation data and actual data analysis show that the proposed method has higher prediction accuracy than some classic prediction models.

2. Gaussian Process Regression Model Based on Stacking Framework

This model adopts the Stacking integrated learning framework. Firstly, the original data set is divided into several sub-data sets and input into multiple base learners of the first-layer prediction model, and each base learner outputs its own prediction result. Then the output of the first layer is used as the input of the second layer to train the meta-learner of the prediction model, and the output of the second layer is used as the final prediction result. The specific implementation steps are as follows:

(1). The model first obtains different sub-training sets by performing different samplings on the original data set in different features and dimensions through the bagging autonomous sampling method. By introducing randomness, the correlation between individual learners is reduced, thereby reducing the variance of the integrated model and improving the overall prediction performance. Here,

the capacity of the sub-training set is the same as that of the original training set. For the random variable$\{\phi_t\}_{t\in T}$, for $\forall n \in N^+, t_1, \ldots, t_n$corresponding to $\phi_1, \ldots \phi_n$, there is $\phi = (\phi_1, \ldots, \phi_n)^T$ satisfying the normal distribution $N(\mu, \sum)$ (μ is the mean of the random variable ϕ , and $\sum$ is the covariance function. Using the kernel trick, each submodel has a different kernel function. Let $\sum_{ij} = k(x_i, x_j)$ (the covariance function can be calculated through k), then $\{\phi_t\}_{t\in T}$ is a Gaussian process, denoted as $\phi_t \sim GP(m(t), k(s,t))$, where:

$$m(t) = E[\phi_t] \tag{1}$$

$$k(t,s) = E\left[(\phi_s - m(s))(\phi_t - m(t))^T\right] \tag{2}$$

If it is assumed that the sample set $\{f(x)\}$ satisfies: $\{f(x)\}_{x\in R^+} \sim GP(m(x), k(x,x'))$

$$m(x) = E[f(x)] \tag{3}$$

$$k(x,x') = E\left[(f(x) - m(x))(f(x') - m(x'))^T\right] \tag{4}$$

(2). Then, the obtained sub-training sets are trained for each Gaussian regression model with different kernel functions, and the divided validation set is put into the trained submodels to obtain the prediction results of multiple base learners, and the predicted values are stored as the new X. Finally, each learner obtains a set of prediction results, which are combined into a set as the prediction result of the first-layer prediction model. The details are as follows: For each submodel, first, the data training set x_i, y_i, obtained by bagging sampling is stored in the data set D. At this time, $D = \{(x_i, y_i)\}_{i=1}^N$. Here, it is assumed that $y = f(x) + \varepsilon$, where $\varepsilon \sim N(0, \delta)$. Define $X = [(x_1, \ldots, x_N)^T]_{N*P}$, $Y = [(y_1, \ldots, y_N)^T]_{N*1}$. Combining the previous assumptions, it can be obtained that $f(x) \sim N(\mu(x), k(x,x'))$ to obtain the Gaussian regression submodel:

$$Y = N(\mu(x), k(x,x') + \delta^2 I) \tag{5}$$

Similarly, all Gaussian regression submodels in the first layer can be obtained, and there is a total of a. Given the data to be predicted, here is the validation set X^* , $X^* = (x_1^*, x_2^*, \ldots, x_a^*)^T$, to find the predicted value Y^* , use

$$\begin{pmatrix} Y \\ f(x^*) \end{pmatrix} \sim N\left(\begin{pmatrix} \mu(x) \\ \mu(x^*) \end{pmatrix}, \begin{pmatrix} k(x,x) + \delta^2 I & k(x,x^*) \\ k(x^*,x) & k(x^*,x^*) \end{pmatrix}\right) \tag{6}$$

Finally, $p(y^*|Y, X, X^*) \sim N(\mu^*, \sum^* + \delta^2 I)$ can be obtained.

(3). The prediction results of multiple base learners are horizontally concatenated into a new data set as the input feature of the second-layer meta-learner, and Y on the validation set is used as the new y for the training of the quadratic Gaussian process regression model, and finally the prediction result, that is, the final prediction distribution, is obtained, and the variance of the prediction distribution can be directly used as an indicator for quantifying the uncertainty. The details are as follows:

The mean of the prediction distribution of each submodel is used as the predicted value Y^*, and all the means are used as the result of the first-layer prediction $Z^* = (Y_1^*, Y_2^*, Y_3^*, \ldots, Y_a^*)$ as the new training set, and the validation set of y is used as the new training set to train the second-layer model, and finally the prediction distribution can be obtained.

(4). The test set data isused as the input and enters each submodel, and the mean value of the Gaussian process regression finally obtained is used as the final predicted value. Flow chart see figure 1.

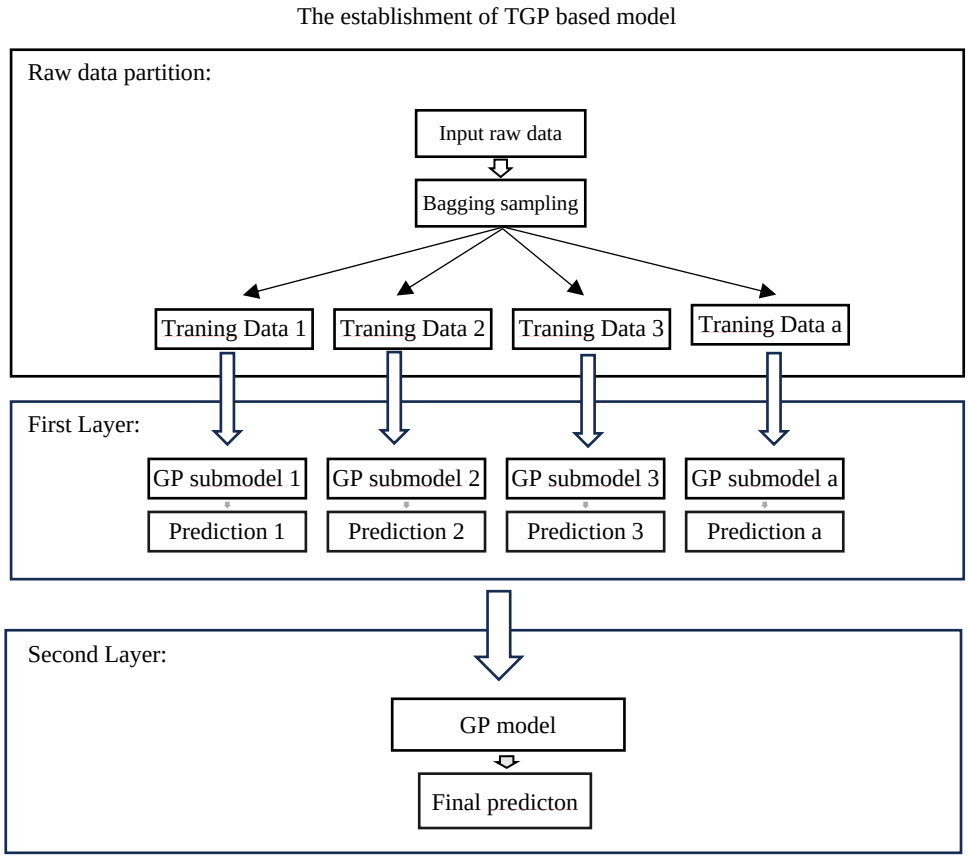


Fig. 1 The establishment of TGP based model

3. Data Analysis

3.1. Data Source and Experimental Setup

In terms of simulated data generation, a data set containing 7 features (x1 to x7) is generated by randomly generating numbers, and each data set contains 100 samples. The y that needs to be fitted is set as the nonlinear combination of x1 - x4, that is

$$y = \sin(\pi \cdot x_1) + e^{x_2} - \ln(x_3 + 1) + 3 \cdot x_4^3 + \varepsilon \tag{7}$$

We added random noise ε to the formula that needs to be fitted and set three irrelevant variables x5, x6, and x7 to verify that the new model can alleviate the impact of irrelevant variables on the reduction of the prediction accuracy of y. This is used as the original simulated data model. The number of samples drawn from the data set is changed to 1000, and the generation distribution of the data set is changed: the samples drawn are from the standard normal distribution and the random floating-point numbers drawn from the uniform distribution, respectively. The fitting function is changed to

$$y = \cos(\pi \cdot x_1) + e^{x_2} - \ln(x_3 + 1) + 3 \cdot x_4^3 + \varepsilon \tag{8}$$

Simulation experiments are carried out on the new model respectively.

The real experimental data comes from the spectral data set of meat samples and the spectral data set of corn samples [10]. The spectral data set of meat samples can be downloaded from http://lib.stat.cmu.edu/datasets/tecor. This data set consists of a total of = 240 samples, each containing the moisture, fat, and protein content of the meat, which are used as different y in the actual data experiment, and the absorption spectrum measured by the near-infrared spectrometer in the wavelength range of 850 - 1050 nanometers (nm). A total of 100 absorption spectrum data

recorded at intervals of every two bands are used as high-latitude features x1 - xn in the data experiment. The spectral data set of corn samples can be obtained from http://www.eigenvector.com/data/Corn/index.html. This data set consists of 80 corn samples, measuring the wavelength range from 1100 to 2498 nanometers at an interval of 2 nanometers, with a total of 700 absorption spectrum data as 700 different features x, and the oil, moisture, and starch content of each corn sample are recorded as three different response variables y.

In the actual data experiment process, we divide the original data set used into 80% training set and 20% test set, and obtain different sub-training sets by performing different samplings on the original data set in different features and dimensions. The capacity of this sub-training set is the same as that of the original training set. For a data set containing n samples, m bootstrap samplings are performed to train m sub-training sets, and the probability of any sample being selected is approximately (1 - (1 - 1/n) ^m). When m → ∞, it is close to 1 - e^(-m/n). Therefore, when m is large enough, each bootstrap sample set roughly contains approximately (1 - e^(-m/n)) n different samples, and approximately e^(-m/n) *n samples are not selected. In the real data set experiment, we draw 100 data sets to train the submodels, use multiple samplings to improve the utilization rate of the data set training model, reduce the unused data, and reduce the impact of outliers on the model training. At the same time, in order to avoid the impact of the randomness of data segmentation on the evaluation of the model's performance, we adopt a design of 10 repeated experiments, and randomly divide the data set by equal probability sampling in each experiment to reduce the deviation of the model training. And the final test set is input into the finally trained model to obtain the final prediction distribution to test the prediction accuracy and stability of the model.

For the specific comparison method, we can use Random Forest (RF), K-nearest neighbor regression (KNN), Lasso Regression (LR), and Back Propagation Neural Network (BPNN) as comparison models. The root means square prediction error of the high-dimensional data regression model based on the stacking-Gaussian process algorithm (referred to as the TGP model later) is subtracted from the root mean square prediction errors of other models, and then divided by the root mean square errors obtained by other models to measure the effectiveness of the TGP model compared with other models. The prediction performance of the model is measured by calculating the root mean square error (RMSE) and the standard deviation on the test set. The root mean square error is the square root of the ratio of the square of the deviation between the predicted value and the true value to the number of observations n, which is used to measure the deviation between the observed value and the true value. The root mean square error can reflect the accuracy of the model fitting. The standard deviation reflects the stability of the model when fitting nonlinear functions in different scenarios by measuring the dispersion degree of a set of numbers themselves.

3.2. Comparison Results

The following Table 1 are the experimental results of the simulated data set:

Table. 1 Experimental results of analog data sets

Random sampling	RMSE	Standard deviation
RF	0.432873	0.087010
KNN	0.530400	0.125833
LR	0.775109	0.180013
BPNN	0.306970	0.051341
TGP	0.244608	0.171866
Change the sampling	RMSE	Standard deviation
RF	0.167058	0.010761
KNN	0.286022	0.010660
LR	0.714115	0.027987
BPNN	0.157900	0.023134
TGP	0.038289	0.009175
Change the fitting	RMSE	Standard deviation
RF	0.407343	0.097206
KNN	0.475921	0.102043
LR	0.731323	0.105775
BPNN	0.275065	0.056524
TGP	0.262868	0.233427

The comparison method mentioned in the previous paragraph can quantify the performance enhancement of the TGP model relative to other traditional models in simulated datasets. Specific results are shown in Table 2 and Figure 2.

Table. 2 performance improvement for TGPmodel relative to other models in analog data sats

Simulated data set	RF	KNN	LR	BPNN
Random sampling	43.49%	53.88%	68.44%	20.32%
Change the sampling	77.08%	86.61%	94.64%	75.75%
Change the fitting	35.47%	44.77%	64.06%	4.43%

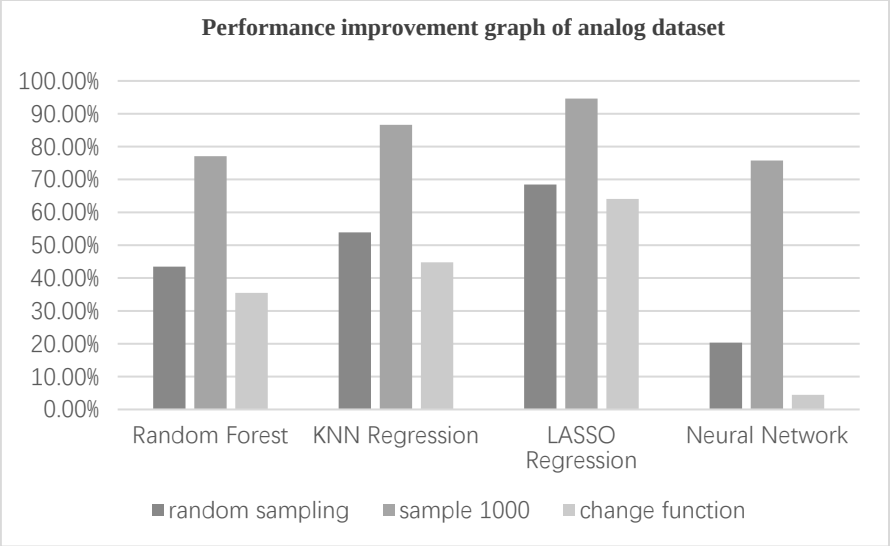


Fig. 2 Performance improvement graph of analog

The chart demonstrates that the TGP model outperforms classical models such as Random Forest, K-nearest neighbor regression, Lasso Regression, and BP neural network in a simulated data environment. It exhibits lower average RMSE and higher prediction accuracy, with an average performance improvement of 52.0% over Random Forest, 61.8% over K-nearest neighbor regression, 75.7% over Lasso Regression, and 33.5% over BP neural network. The TGP model's superiority is particularly significant in high-dimensional datasets with feature dimensions surpassing the number of samples. It also showcases significantly improved prediction accuracy and generalization potential. Overall, the TGP model demonstrates superior performance in handling complex data structures.

The results of the experiments on the meat dataset are shown in Table 3:

Table. 3 Experimental results of meat data sets

Meat dataset(oil)	RMSE	Standard deviation
RF	7.377124	1.176503
KNN	9.093427	0.916180
LR	4.873811	0.370217
BPNN	12.349808	0.887250
TGP	4.128291	2.058619

The comparison method mentioned in the previous paragraph can quantify the performance enhancement of the TGP model relative to other traditional models in meat data datasets. Specific results are shown in Table 4 and Figure 3.

Table. 4 performance improvement for TGPmodel relative to other models in meat data sats

Meat dataset	RF	KNN	LR	BPNN
	44.04%	54.60%	15.30%	66.57%

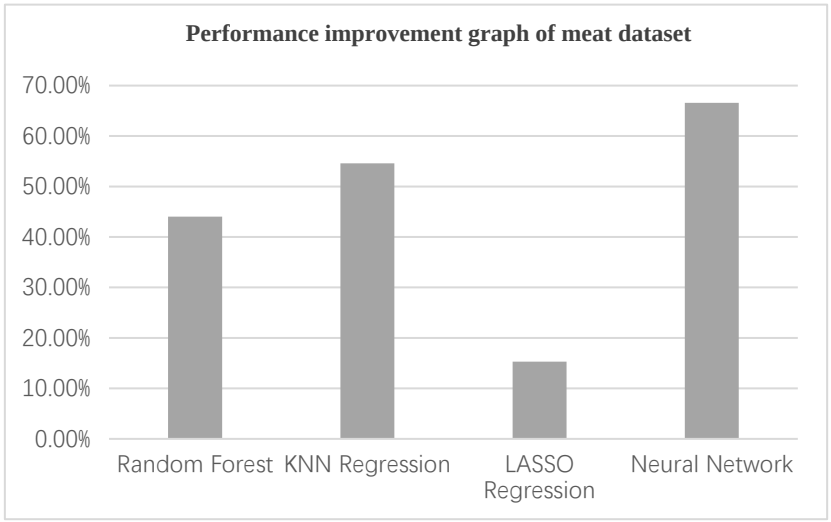


Fig. 3 Performance improvement graph of meat dataset

Then comes Table 5, the experimental results of the corn dataset:

Table. 5 Experimental results of meat corn sets

Corn dataset(oil)	RMSE	Standard deviation
RF	0.251660	0.035209
KNN	0.275698	0.032558
LR	0.373012	0.028828
BPNN	2.069952	0.392928
TGP	0.152058	0.026816
Moisture	RMSE	Standard deviation
RF	0.253388	0.025124
KNN	0.284303	0.038274
LR	0.350688	0.050216
BPNN	2.019418	0.436316
TGP	0.161981	0.040076
Protein	RMSE	Standard deviation
RF	0.504374	0.071781
KNN	0.511583	0.068079
LR	0.471964	0.080765
BPNN	1.519377	0.276627
TGP	0.207098	0.047011
Starch	RMSE	Standard deviation
RF	0.943038	0.105438
KNN	0.871784	0.133594
LR	0.862700	0.118180
BPNN	6.723383	1.147271
TGP	0.640467	0.143557

The comparison method mentioned above can obtain the degree of performance improvement of TGP model relative to other traditional models in the corn data set. Specific results are shown in Table 6 and Figure 4:

Table. 6 performance improvement for TGPmodel relative to other models in corn data sats

Corn dataset	RF	KNN	LR	BPNN
oil	39.58%	44.85%	59.24%	92.65%
Moisture	36.07%	43.03%	53.81%	91.98%
Protein	58.94%	59.52%	56.12%	86.37%
Starch	32.08%	26.53%	25.76%	90.47%

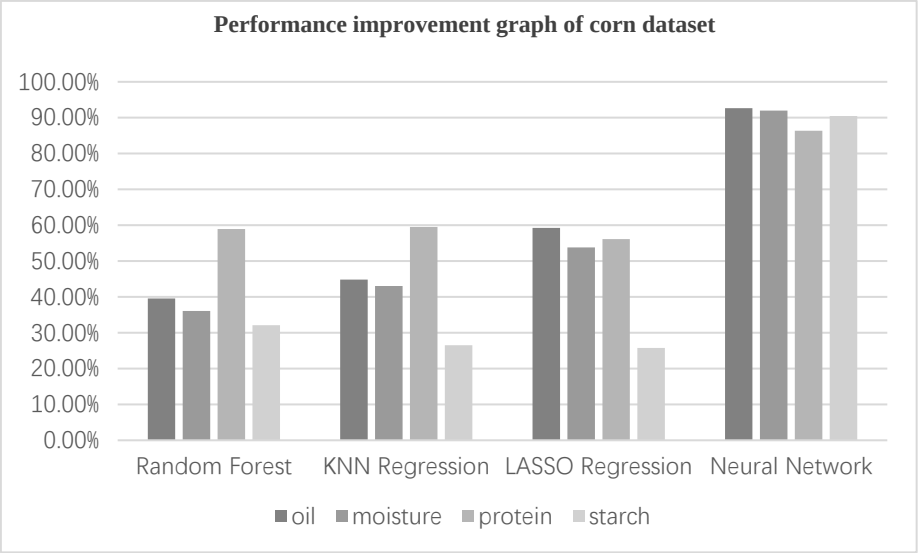


Fig. 4 Performance improvement graph of corn dataset

When we shift our focus to more complex and dynamic real-world data scenarios, the TGP model continues to demonstrate its excellent predictive ability. Whether it is a detailed analysis of key nutritional components in meat products such as fat content, or precise determination of the composition of corn, including water, fat, protein, and starch content in multiple dimensions, the TGP model shows a certain advantage in prediction accuracy compared to traditional models, especially in higher-dimensional datasets. For example, in the corn dataset, the TGP model has an average performance improvement of 41.67%, 43.48%, 48.73%, and 90.37% relative to the four models. This achievement not only demonstrates the TGP model's strong ability to capture complex nonlinear relationships in data, but also highlights its efficiency and robustness in dealing with the inherent high uncertainty and complexity of real-world data.

4. Conclusion

In this paper, we have discussed the challenges of handling high-dimensional data and constructing regression prediction models. We have presented a high-dimensional data regression model based on the stacking framework and Gaussian process algorithm. Firstly, to address the curse of dimensionality associated with high-dimensional data, we applied the stacking framework to the Gaussian process regression model, and this innovative attempt effectively improved the model's predictive accuracy when handling high-dimensional input data. Secondly, by using the Gaussian process regression model, complex unknown structural relationships can be fitted, and the uncertainty of the regression distribution can be quantified. Finally, through simulation experiments and real data analysis, it has been demonstrated that the proposed method has smaller prediction errors and better predictive accuracy compared to other methods such as Random Forest, Lasso Regression, and Back Propagation Neural Network.

After in-depth exploration and verification of the significant advantages of the TGP in high-dimensional data processing and fitting nonlinear relationships, we can apply the model to more practical scenarios, such as digital healthcare, aerospace, biotechnology, and other fields. With the advent of the big data era, the application of non-Euclidean data (such as text, images, social network graphs, etc.) in various industries is becoming increasingly widespread, and the model can be further used in the processing of non-Euclidean data information. For example, in the financial market, investment sentiment factors, as an important type of non-Euclidean dataset, contain rich market information and investor expectations, which have an undeniable impact on stock price prediction. The complexity and unstructured nature of such data pose a great challenge to traditional machine learning models. For non-Euclidean data, especially textual sentiment factors, we can design or select kernel functions that capture semantic information, emotional tendencies, and other deep features,

thereby using the method proposed in this paper to predict stock price trends. With the continuous development of deep learning technology, the integration of deep learning with other models to form new models that are deeper and more efficient will be an important direction for future research. This kind of fusion model is expected to demonstrate superior performance in handling high-dimensional and complex data.

References

[1] Bai Y, Qian M, Tian M. Variable selection in partially linear additive quantile regression model for ultra-high dimensional data[J]. Statistics and Decision, 2024, (09):43-48.

[2] Fu M, Zhou S. A nonparametric additive quantile regression model based on fusion Lasso[J]. Pattern Recognition and Artificial Intelligence,2024, (01):58-72.

[3] Li X. High-dimensional statistical inference based on ridge regression and its application in gene data[D]. Shanghai University of Finance and Economics, 2023. DOI: 10.27296/d.cnki.gshcu.2023.000005.

[4] Huang W, Deng D, Du J, et al. A high-dimensional logistic regression model based on predictive variable graph structure[J]. Journal of Chongqing Technology and Business University (Natural Science Edition),2021,38(05):107-113.

[5] Zhang Z, Ren J. Nonparametric dynamics modeling for underwater vehicles using local adaptive moment estimation Gaussian processes learning[J]. Nonlinear Dynamics. Volume112, Issue 7. 2024.PP 5477-5502.

[6] Nur C, Budi L, I Nyoman B. Estimation of Multiresponse Multipredictor Nonparametric Regression Model Using Mixed Estimator[J]. Symmetry. Volume 16, Issue 4. 2024. DOI: 10.3390/SYM16040386

[7] Zhu N, You J, Xu Q. Iterative adaptive robust variable selection of nonparametric additive models[J]. Applied Probability and Statistics,2024,40(02):201-228.

[8] Zhang Y, Tang Y. The prediction model of calcium ion concentration in seawater circulation cooling system based on random forest algorithm[J]. Salt Science and Technology,2024,53(07):19-22.

[9] Liu Z, Li R, Pan Y. Study on neural network inference method for high-dimensional non-probabilistic sample data[J]. Journal of Hubei University (Natural Science Edition),2023,45(05):684-694.

[10] Zhu R, Zou G, Zhang X. Model averaging method for partial function linear models[J]. Journal of Systems Science and Mathematical Sciences,2018,38(07):777-800.