Partial functional linear regression based on the Stacking and RKHS

Jia Chen[1, #], Zixin Chen[2, #], Chengwen Zhang[3, *, #]

[1]School of Mathematical Sciences, Chengdu University of Technology,Yibin, China, 644005

[2]College of Mathematics and Informatics College of Software Engineering, South China Agricultural University, Guangzhou, China, 510630

[3]Faculty of Arts and Sciences, Beijing Normal University, Zhuhai, China, 519087

*Corresponding author: zcw111425@163.com

#These authors contributed equally.

Abstract. Functional regression models represent a crucial research area within functional data analysis. To enhance the flexibility of the model, this paper proposes a partial functional linear regression model based on ensemble learning and kernel techniques. On one hand, this method effectively models the relationship between non-linear predictor variables and scalar response variables by employing the Reproducing Kernel Hilbert Space. On the other hand, it utilizes Functional Principal Component Analysis to approximate and estimate functional predictor variables, and addresses the selection of truncation numbers through the stacking framework. In the stacking framework, the meta model takes a model-free form, further increasing the flexibility of the model and effectively balancing the variance and bias of the prediction model. The results of simulation experiments and real-world data analysis demonstrate that the proposed method is more competitive compared to traditional benchmark methods.

Keywords: Functional Data Regression, Nonlinear Relationship Fitting, Ensemble Learning, Truncation Number.

1. Introduction

With the rapid advancement of sensor and data acquisition technologies, a growing number of fields are able to collect high-frequency, high-precision continuous data. These data often manifest as continuous numerical points that vary with the independent variable and can be represented as curves, surfaces, or other types of continuous graphs, collectively referred to as functional data. Developing regression models for functional data holds significant practical value across various domains. For instance, Wu Hao [1] explored a robust variable selection method within the partially linear model using functional data, highlighting its practical utility; Jin Xueqing [2] primarily investigated principal component analysis and least squares estimation, as well as their applications in functional data models, and Liu Zhuang [3] conducted research on the classification methods of functional data utilizing both supervised and unsupervised classification techniques. A wealth of research exists on functional linear models. Zhu Jie [4] integrated the principal component dimensionality reduction method with dynamic principal component analysis, establishing a high-dimensional functional dynamic principal component linear regression model. Wang Huiwen et al. [5] proposed a generalized linear model incorporating functional covariates and conducted relevant estimations. Wang Qingrong [6] explored functional linear regression models with autoregressive error terms, estimating the slope function through functional principal component analysis. Huang Hua et al. [7] predicted soluble solid content (SSC) in apples using a functional linear regression model based on visible/near-infrared spectroscopy. Zhang Heng et al. [8] applied triangular spline estimation to estimate parameters in functional linear regression models, while Jin Ting [9] employed Group Lasso to identify change points and improve model accuracy. Liu Xuan et al. [10] enhanced the adaptability of functional regression models in complex data analysis using change-point tests.

However, in real-world applications, functional data are often accompanied by vector-valued covariates. In medical statistics, for instance, patient characteristics like age, gender, weight, and medical history, as well as high-dimensional genomic data (such as genotype information and gene expression profiles), are commonly used to analyze disease risks or treatment effects. Similarly, in finance and economics, vector-valued covariates—such as historical transaction data, economic indicators, and policy changes—play a significant role in predictive modeling. Therefore, integrating these vector-valued covariates into functional data regression models is crucial to fully capture variable relationships and enhance prediction accuracy. In response to this challenge, scholars have proposed partially functional linear models that integrate functional data with vector-valued covariates to improve model flexibility and predictive accuracy. For example, Yu Ping et al. [11] combined additive quantile regression with functional linear quantile regression to propose a partially functional linear additive quantile regression model, approximating the slope function using functional principal component basis functions. Daren Wang et al. [12] introduced a functional linear regression model capable of handling both functional and high-dimensional vector covariates. Yang Weiming et al. [13] improved the partially functional linear model using residual function principal component analysis to estimate partial linear relationships. Hu Yang [14] developed a robust estimation method for the partially functional linear regression model, while Wen Liyu [15] proposed a variance homogeneity test framework, important for dealing with complex error structures in such models. Zhu Rong et al. [16] reduced the dimensionality of functional data using functional principal component analysis (FPCA) and employed model averaging to enhance prediction accuracy and robustness. Despite these advancements, several limitations remain. Many studies assume a linear relationship between vector-valued covariates and response variables, restricting model flexibility and applicability. Furthermore, FPCA is typically employed for dimensionality reduction in functional data, and the choice of truncation number plays a critical role. However, this choice is often made subjectively, which can lead to overfitting or underfitting, thus undermining the model's predictive performance and generalization ability.

To address these issues, this paper proposes a partially functional linear regression modeling approach based on the Stacking framework and reproducing kernel theory. First, FPCA is applied to reduce the dimensions of both simulated and real datasets, retaining the most significant components. For each dimension-reduced dataset, K-fold cross-validation is performed to generate K sets of predicted values, which are paired with the corresponding response values to form a new training dataset for a linear regression model. The dataset is then used to predict the actual data, with the predictions integrated by averaging to yield the final output. This integrated prediction is subsequently input into the trained linear regression model to obtain the predicted response values. To assess the efficacy of this method, it is compared with various models such as random forests, support vector regression, Gaussian process regression, extreme gradient boosting, and gradient boosting decision trees. The results demonstrate that the proposed method achieves strong generalization and superior prediction performance, balancing model complexity with fitting accuracy.

2. Methodology

2.1. Functional Principal Component Analysis

Functional Principal Component Analysis (FPCA) is a method suitable for analyzing sample data that is functional in nature and examining the characteristics of its variations. Functional data typically refers to data measured over time, space, or other continuous variables, represented in the form of functions rather than single values. The following outlines the steps for deriving the corresponding principal component functions.

Suppose we have K set of functional data $X_i(t)(i = 0,1,2,...,K)$. Let the sample estimate function for this set be:

$$\hat{c}(s,t) = \frac{1}{n}\sum_{i=1}^{n} X_i(t) \times X_i(s) \tag{1}$$

Similar to PCA, we perform eigenvalue decomposition as follows:

$$\sum \beta = \lambda\beta \tag{2}$$

In which β is the eigenvector, and λ is the eigenvalue. Therefore, the principal component functions of FPCA should be the eigenfunctions of the sample estimate function, and their solution method is similar to the decomposition method of PCA. That is, the spectral decomposition of the FPCA covariance function is:

$$\int_t c(s,t) \times \beta(t)dt = \lambda \times \beta(s) \tag{3}$$

where λ and $\beta(t)$ are the eigenvalues and eigenfunctions of the covariance function $c(s,t)$.

If $\beta(t)$ and $X_i(t)$ are represented as linear combinations of basis functions :

$$X_i(t) = \sum_{k=1}^{K} \hat{a}_{ki} \times \phi_k(t) = \phi^T \cdot a_i \tag{4}$$

$$\beta(t) = \sum_{k=1}^{K} b_k \times \phi_k(t) = \phi^T \cdot b \tag{5}$$

where a_i and b are the coefficients in front of the basis functions.

By substituting (1), (4), and (5) into (3), the following formula can be derived:

$$\frac{1}{n}\sum_{i=1}^{n} a_i^T a_i \cdot \phi^T \int \phi^T \phi dt \cdot b = \lambda \cdot \phi^T \cdot b \tag{6}$$

Let $\int \phi^T \phi dt = w$. By simplifying the formula in (6), the following can be derived:

$$\frac{1}{n}\sum_{i=1}^{n} a_i^T a_i \cdot w \cdot b = \lambda \cdot b \tag{7}$$

Given that $\frac{1}{n}\sum_{i=1}^{n} a_i^T a_i \cdot w$ is a symmetric matrix, λ and b can be understood as the eigenvalues and eigenvectors of this symmetric matrix. When we perform eigenvalue decomposition on $\frac{1}{n}\sum_{i=1}^{n} a_i^T a_i \cdot w$, the eigenvectors are obtained. By substituting eigenvectors into (5), we can get $\beta(t)$, thus we have found the principal component functions in multivariate functional data.

Through FPCA, we can derive that a certain functional data is represented by the following formula using the unit orthogonal principal component functions:

$$X(t) = \sum_{k=1}^{\infty} \xi_k \beta_k(t) \tag{8}$$

where $\beta_k(t)$ is the principal component function, and ξ_k is the amount of information projected onto the corresponding principal component function direction.

2.2. Partial Functional Linear Regression based on Stacking and Reproducing Kernel

Firstly, reproducing kernel estimation is a non-parametric statistical method based on Reproducing Kernel Hilbert Space (RKHS). The core idea is to use the properties of the reproducing kernel to

estimate functions. A reproducing kernel is a specific type of kernel function, which is a kernel function on the Reproducing Kernel Hilbert Space (RKHS) that has the reproducing property. Each function f can be viewed as an infinite-dimensional vector, and the binary function $k(x,y)$ can be viewed as an infinite-dimensional matrix. If it satisfies:

Positive definiteness:

$$\iint f(x)\times k(x,y)\times f(y)dxdy \geq 0 \tag{9}$$

Symmetry:

$$k(x,y)=k(y,x) \tag{10}$$

Then this function is a kernel function. By selecting an appropriate reproducing kernel, we can construct an estimation function. Secondly, Stacking is an ensemble learning method that combines the predictions of multiple models to obtain a more accurate classification approach. The Stacking model can be roughly divided into two stages: Stage one: Use different models to predict the sample data, and their predicted values are used as inputs for stage two; Stage two: Combine the predictions from stage one with the original sample values to create new samples, resulting in the final prediction. Next, let's assume that the partial functional linear model Y is:

$$Y=\sum_{i=1}^{l}\int x_l(t)\times\beta_l(t)dt+g(s)+\varepsilon \tag{11}$$

where A is a nonlinear function. By expanding the functional principal components:

$$x(t)=\sum_{l=1}^{L}\xi_l^T\phi_l(t) \quad \beta(t)=\sum_{l=1}^{L}b_l^T\phi_l(t) \tag{12}$$

and the reproducing kernel representation:

$$g(s)=\sum_{i=1}^{n}a_ik_i(s_i,t) \tag{13}$$

Substituting into the model gives:

$$Y=\sum_{l=1}^{L}\xi_l^Tb_l+\sum_{i=1}^{n}a_ik_i(s_i,t)+\varepsilon \tag{14}$$

Furthermore, the above expression can be transformed into a linear model as follows:

$$Y=(\xi_l^T \quad k^T)\begin{pmatrix}b_l\\a\end{pmatrix}+\varepsilon \tag{15}$$

Solving the above linear model allows for the parameter estimation of the partial functional linear model based on the reproducing kernel principle. Next, we present a partial functional linear ensemble model based on the stacking framework. Specifically, assume we have two sets of data: the first set (①) consists of real input-output data, which is used as the training set; the second set (②) consists of new input data, which is used as the test set. Our goal is to predict the corresponding values for② using the proposed method. The specific steps are as follows:

a. Assume the number of basis expansions is set to k. Divide the sample data of ① into H parts, using the first part as the test set and the remaining (H-1) parts as the training set. After training the model, insert the first part back and obtain the prediction for the first part.

b. Using H-fold cross-validation, take the second part of ① as the test set and the remaining (H-1) parts as the training set. Repeat the steps in a to obtain the prediction for the second part. After all H parts have been processed, the predictions for ① will be obtained

c. Place the test set ② into each trained model to make predictions, resulting in H predictions, denoted as y_1^*. The arithmetic mean of these predictions is then taken to obtain $\bar{y}_1^*$.

d. Change the number of basis expansions, and iterate over $i(i = 2,3,4,...,K)$, repeating steps a-c. This will yield an N*K dimensional prediction matrix for the training set ① under different basis expansions, and a K-dimensional prediction vector $(\bar{y}_1^*, \bar{y}_2^*, \bar{y}_3^*, ..., \bar{y}_K^*)$ for the test set ②.

e. Pair the N*K dimensional prediction matrix with the response data of the training set ① and train a meta-predictor model (which can be any regression model). Then, input $(\bar{y}_1^*, \bar{y}_2^*, \bar{y}_3^*, ..., \bar{y}_K^*)$ into the trained meta-predictor model to obtain the final prediction result $\hat{y}^*$ for the test set ②.

3. Data analysis

3.1. Simulation studies

The purpose of this experiment is to construct and evaluate the Functional Kernel-based Generalized Penalized Regression model.By comparing the performance difference between the FKPGR model and the traditional machine learning model in processing functional data, it provides an effective reference for the modeling of functional data.First, a Fourier basis function is used to generate functional data $\mathrm{X(t)}$, where each sample is represented by a linear combination of 99 basis functions, with the basis function coefficients initially decaying by $\mathrm{y = k^{-\frac{3}{4}}}$ to ensure the smoothness of the function, and 100 points taken equidistant on the interval [0,1];Next, the regression coefficient function is obtained from a linear combination of the basis functions, with the coefficients decaying by k^{-2} to further enhance the smoothness;Then, an initial random variable $\mathrm{S \sim N(0,1)}$ is introduced to generate the nonlinear effects through the initial nonlinear function $\mathrm{gp = \sin(\frac{2}{3}\pi S)}$; Finally, the overall response variable y consists of the inner product of functional data, the non-linear random effects, and the noise term:

$$\mathrm{y} = \int \mathrm{x}(t)\beta(t)\,dt + g(s) + \varepsilon \tag{16}$$

Where $\varepsilon \sim \mathrm{N}(0, 0.01^2)$.In the model training and evaluation section, First the dataset is divided between training set (60%) and test set (40%),and cycling through 10 Monte Carlo experiments with a range of machine learning models such as Random Forest Regression , Support Vector Regression and Gaussian Kernel, Gaussian Process Regression, XGBoost Regression and Gradient Boosted Decision Tree Regression ,where the dataset is randomly divided for each experiment to improve the robustness of the results. Data prediction was then performed through the models and the mean square error of each model on the test set was calculated for model performance evaluation; Finally, the average mean squared error and the standard deviation (SD) of each model is calculated based on the error results of several experiments to compare and illustrate the goodness of the FKGPR model.

Table 1: Mean Squared Error and Standard Deviation for different sample sizes

Models	Value	Sample size			
		n=100	n=200	n=300	n=400
FKGPR	Mean	0.0139	0.0009	0.0023	0.0003
	SD	0.0108	0.0007	0.0035	0.0002
RF	Mean	0.0615	0.0140	0.0134	0.0037
	SD	0.0343	0.0051	0.0089	0.0019
KSVR	Mean	0.1440	0.0631	0.0584	0.0339
	SD	0.0429	0.0224	0.0150	0.0056
GPR	Mean	0.0821	0.0280	0.0342	0.0116
	SD	0.0212	0.0081	0.0104	0.0022
XGBoost	Mean	0.0983	0.0275	0.0203	0.0059
	SD	0.0209	0.0135	0.0133	0.0026
GBDT	Mean	0.0580	0.0126	0.0134	0.0028
	SD	0.0319	0.0072	0.0112	0.0014

We explore whether the FKGPR models are stable in the use of functional data when each parameter is varied. First, the stability and generalization ability of the model when the amount of data varies is explored by using different sample sizes. Experiments are conducted on sample sizes n of 100/200/300/400 while ensuring that other parameters remain constant, and the results obtained are shown in Figure 1 and Table 1. When the sample size increases, the MSE distribution of FKPGR model is more centralized, the fluctuation is smaller, and it still maintains better stability compared with other traditional machine learning models; the FKGPR model has the lowest MSE and smaller standard deviation, and performs the best in all the cases, and with the increase of the sample size, its error decreases rapidly, and the stability is very high.

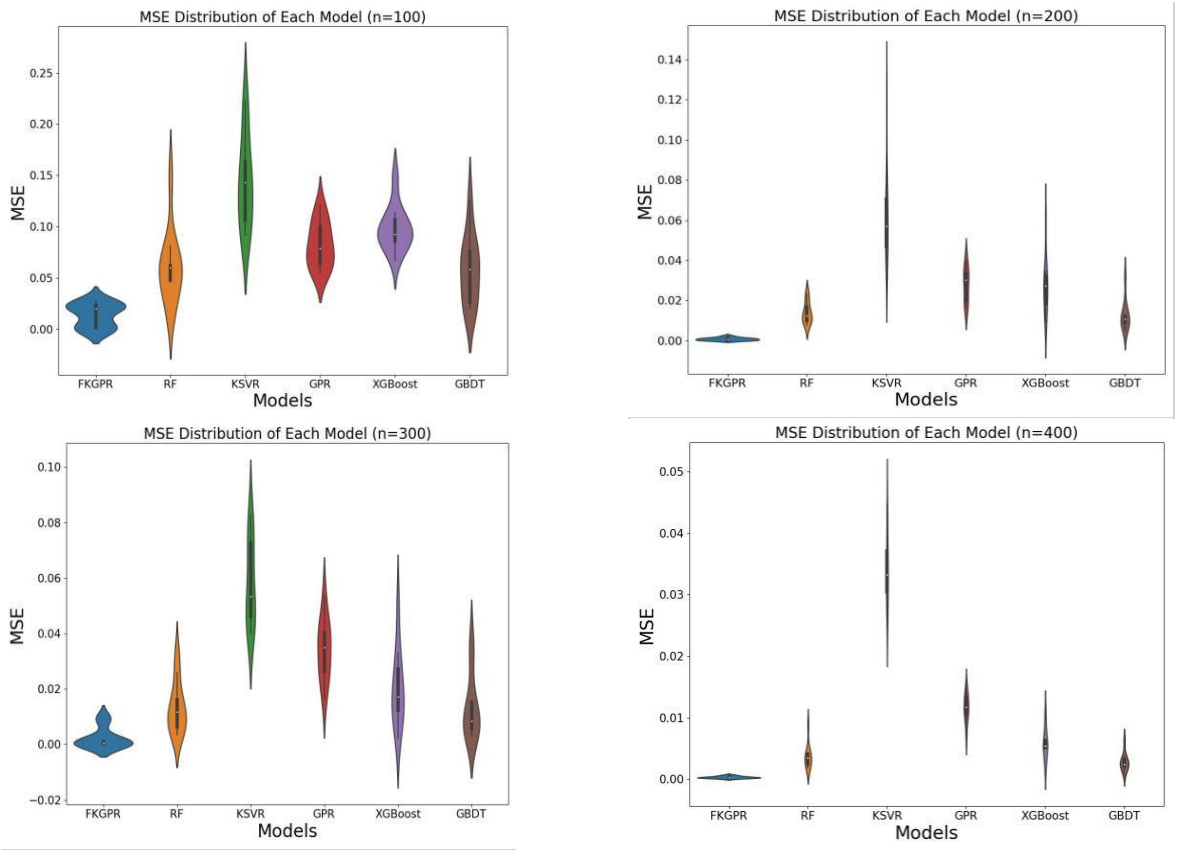

Figure 1: Comparison result of MCMC for different sample sizes on the simulated dataset

Table 2: Different basis expansion coefficient functions

Type	Basis expansion coefficient function
1	$f = k^{-\frac{1}{2}}$
2	$f = k^{-\frac{3}{4}}$
3	$f = k^{-1}$
4	$f = k^{-2}$

Next, the main purpose of the model comparison through different base expansion coefficient functions is to assess the effect of different base expansion coefficients on the model performance. The specific base-expansion coefficients used in the experiments are listed in Table 2 below. The experimental results are shown in Figure 2 and Table 3, the MSE values of the FKPGR model under different base expansion coefficient functions are more concentrated compared with other models, which is especially obvious when the base expansion coefficient functions are types 2 and 3 in Table 3, indicating that the model is more stable and better than the traditional machine learning model when the functional data are more detailed; the FKGPR model shows lower mean square error and standard deviation under all base expansion coefficient functions, indicating that the model has strong prediction accuracy and stability when dealing with function data.

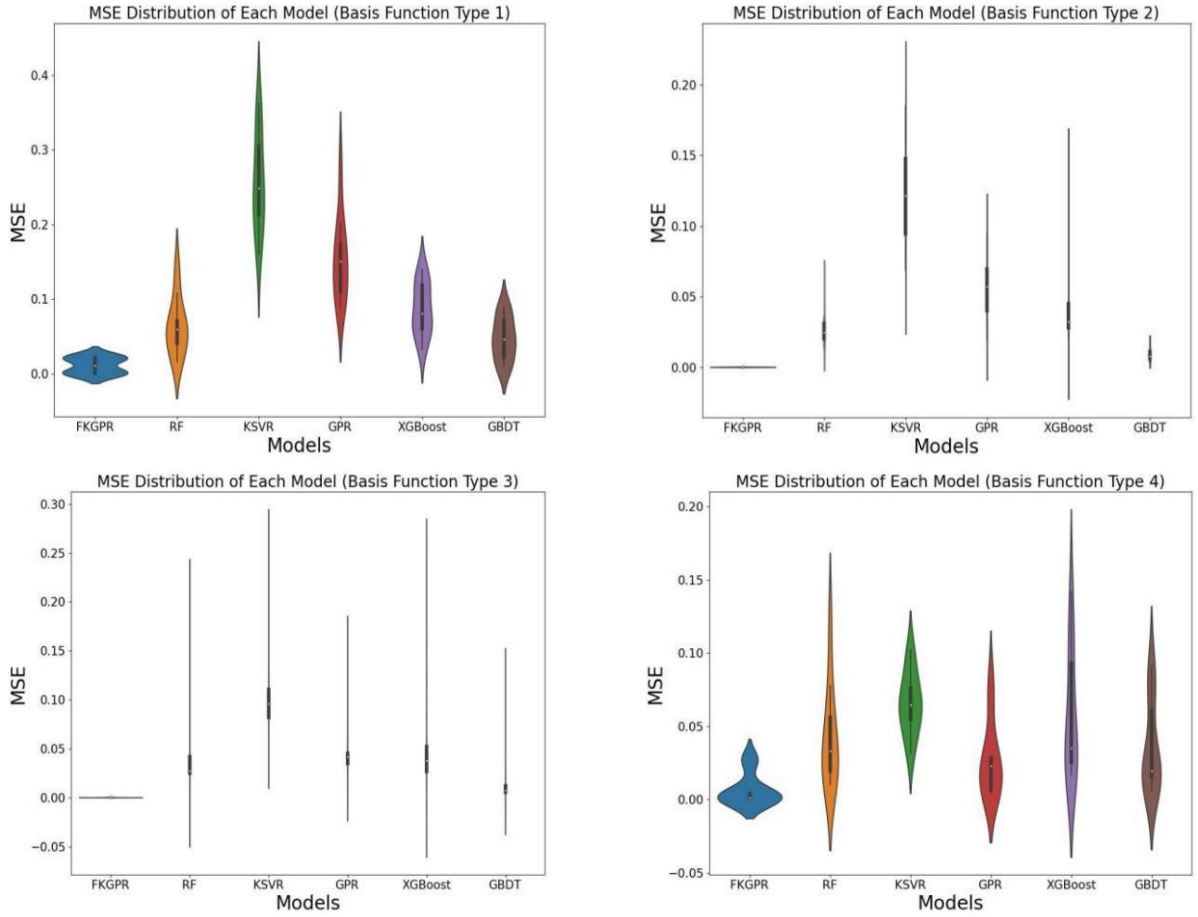

Figure 2: Comparison results of MCMC for different basis expansion coefficient functions

Table 3: Mean Squared Error and Standard Deviation for different basis expansion coefficient

Models	Value	Basis expansion coefficient function			
		$f = k^{-\frac{1}{2}}$	$f = k^{-\frac{3}{4}}$	$f = k^{-1}$	$f = k^{-2}$
FKGPR	Mean	0.0111	0.0003	0.0002	0.0067
	SD	0.0102	0.0001	0.0001	0.0103
RF	Mean	0.0632	0.0280	0.0447	0.0430
	SD	0.0374	0.0123	0.0467	0.0338
KSVR	Mean	0.2570	0.1226	0.1085	0.0659
	SD	0.0628	0.0342	0.0457	0.0204
GPR	Mean	0.1538	0.0558	0.0498	0.0274
	SD	0.0542	0.0214	0.0326	0.0251
XGBoost	Mean	0.0869	0.0474	0.0533	0.0579
	SD	0.0335	0.0325	0.0547	0.0421
GBDT	Mean	0.0468	0.0091	0.0184	0.0361
	SD	0.0277	0.0040	0.0311	0.0301

Finally, to evaluate the impact of different random effects on the performance of the regression model, the non-linear connection function gp is modified to assess each model. Random effects will be used in the experimental process, the$S$defined as a random variable generated from a normal distribution$S \sim N(0,1)$, which represents the uncontrollable or unpredictable part of the data; four different random effects functions were chosen for the experiment: The $\sin(\frac{2}{3}\pi S)$ uses a sine function to simulate the random effects of periodic changes, which is suitable for phenomena with periodic characteristics. The $\cos(\pi S)$ is also a periodic function, but the magnitude of its variation is different from that of the sine function. The e^S is an exponential function, which is suitable for representing the case where the random effect grows rapidly as the variable S increases. The $\ln(|S|)$ is a logarithmic function, which can be used to model the phenomenon that the random effect gradually slows down with the change of S. The results obtained are shown in Figure 3 and Table 4.

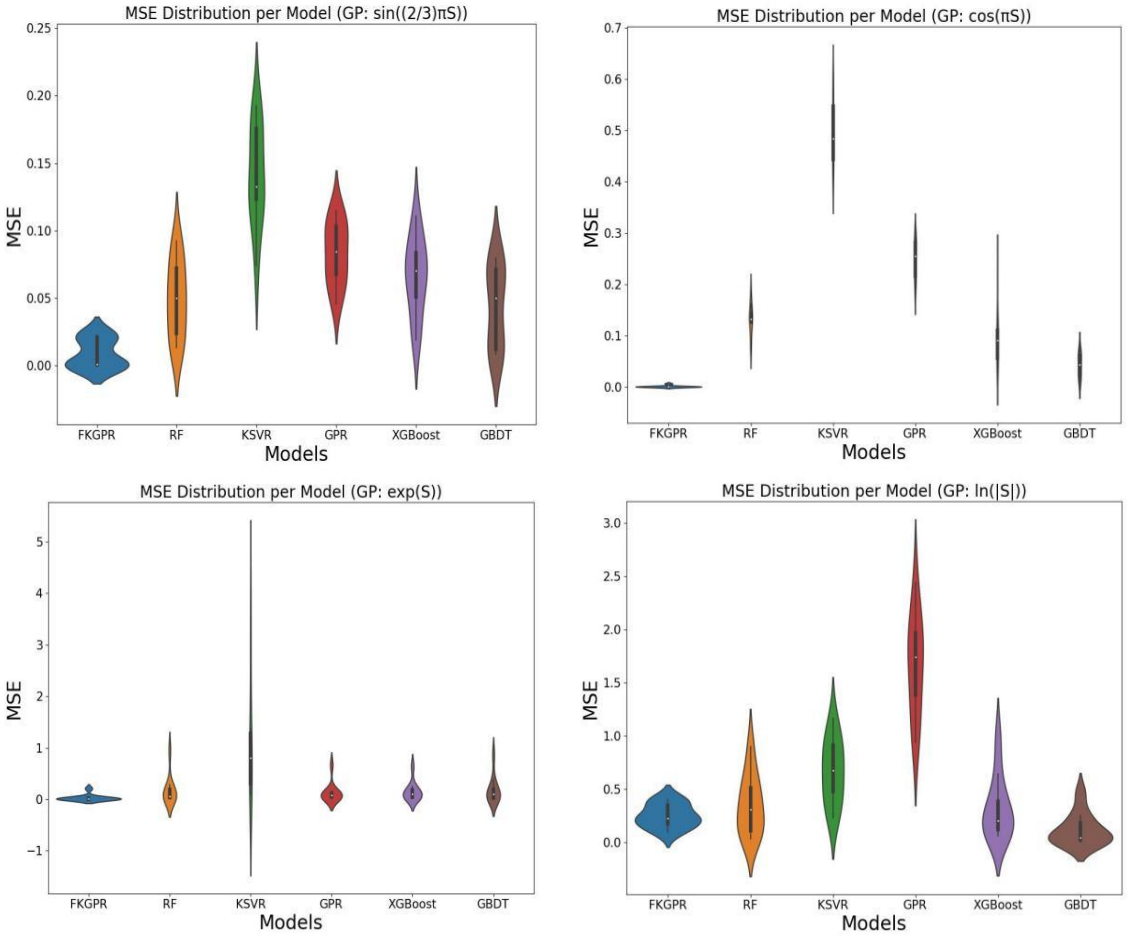

Figure 3: Comparison results of MCMC for different non-linear connection function

Table 4: Mean Squared Error and Standard Deviation for different non-linear functions

Models	Value	Nonlinear connection function			
		$gp = \sin(\frac{2}{3}\pi S)$	$gp = \cos(\pi S)$	$gp = e^S$	$gp = \ln(\|S\|)$
FKGPR	Mean	0.0090	0.0016	0.0289	0.2486
	SD	0.0102	0.0019	0.0608	0.1039
RF	Mean	0.0493	0.1317	0.1771	0.3489
	SD	0.0271	0.0294	0.2705	0.2662
KSVR	Mean	0.1414	0.4958	1.1583	0.6667
	SD	0.0353	0.0603	1.1771	0.2916
GPR	Mean	0.0831	0.2485	0.1318	1.6673
	SD	0.0222	0.0354	0.1821	0.4473
XGBoost	Mean	0.0662	0.0966	0.1538	0.3188
	SD	0.0272	0.0531	0.1776	0.2799
GBDT	Mean	0.0443	0.0430	0.1851	0.1175
	SD	0.0289	0.0223	0.2518	0.1405

As can be seen in Figure 3, among the selected gp functions, the MSE data distribution of the FKPGR model is still "flat gourd-shaped" and relatively stable, while the MSE distribution of the traditional machine learning model is large and fluctuates greatly, and most of them are "crack-shaped". The data in Table 4 show that when different non-linear connection functions are applied to the FKGPR model, its mean square error and standard deviation show more obvious advantages.

In addition, we perform a sensitivity analysis, which is used to assess the sensitivity of the FKPGE model to different parameters. This is done by varying the hyper-parameters of the model $(\lambda_1, \lambda_2, \sigma)$ to understand the impact of these parameters on the model's predictive performance. Changes in these parameters directly affect the learning ability and prediction error of the model, hence sensitivity analysis helps in determining the optimal values of these hyper-parameters. In the experiment, we first discuss the influence of λ_1 on the FKPGR model. We establish the model by taking 100 values in the (0.0001, 10) interval and output the corresponding mean square error, as shown in Figure 4 below. It can be seen that as the value of $\lambda_1$changes, the mean square error of the FKPGR model shows some fluctuations, but there is no obvious monotonic trend or rule, indicating that the value has a small impact on the performance of the model. Although the MSE values are relatively low within certain ranges, the overall changes do not significantly alter the mean square error level of the model.

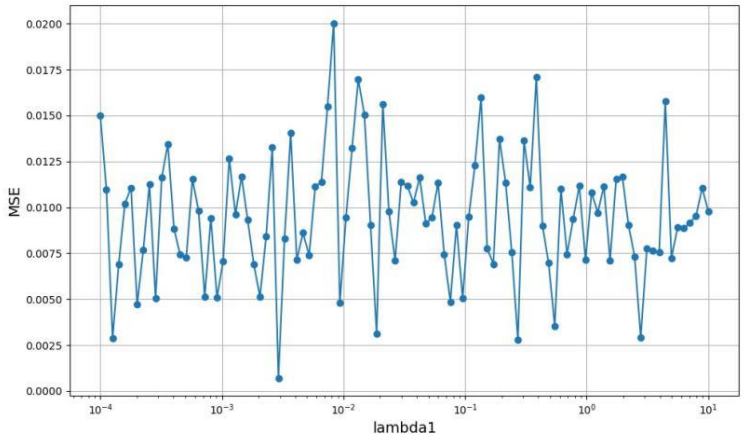

Figure 4: Variation of Mean Squared error for different values of λ_1

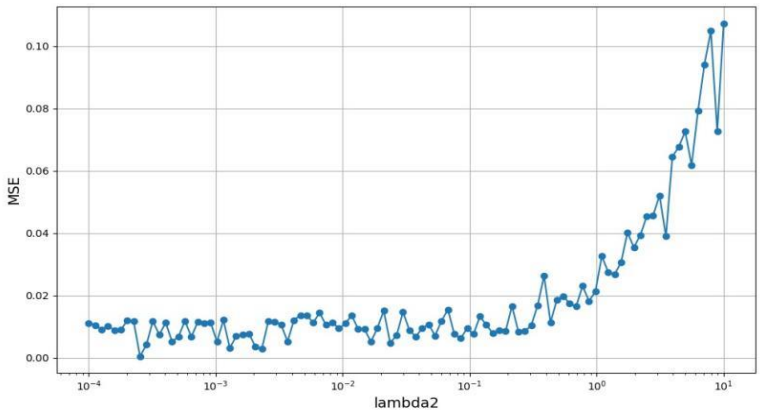

Figure 5: Variation of Mean Squared error for different values of λ_2

Next, we discussed the impact of λ_2 on the FKPGR model. We established the model with 100 values in the (0.0001, 10) interval and output the corresponding mean square error, as shown in Figure 5. When the value of λ_2 increases, the mean square error of the FKPGR model exhibits an exponential growth. When the value of λ_2 is low, the MSE is low, indicating a good fitting effect of the model. However, as the value of λ_2 gradually increases to larger values such as 0.001 and 0.01, the MSE shows a significant increase, especially when the value reaches 0.1 and above, the MSE increases significantly, indicating a significant decline in the performance of the model. This phenomenon indicates that as a regularization parameter, a value of λ_2 that is too large may cause the model to be too smooth, resulting in a loss of effective fitting ability to the data and an increase in error. Therefore, selecting the appropriate$\lambda_2$value is crucial for ensuring the model's generalization ability and prediction accuracy.

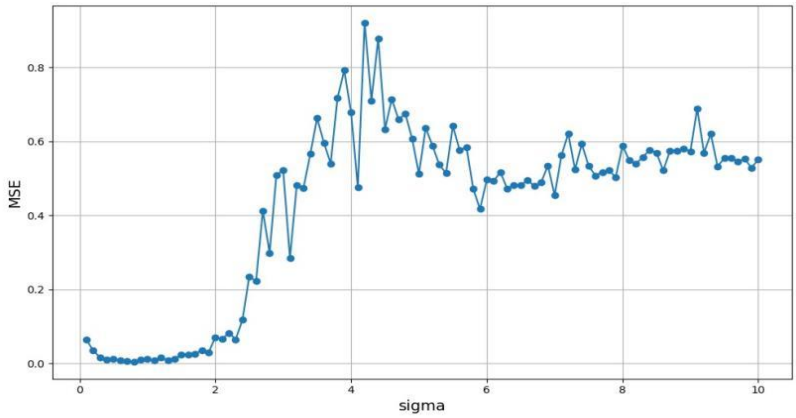

Figure 6: Variation of Mean Squared error for different values of σ

Finally, to discuss the effect of σ on the FKPGR model, we established the model with 100 values in the (0.1, 10) interval and output the corresponding mean square error, as shown in Figure 6. The value of σ has a significant effect on the mean square error of the FKPGR model. When the σ value is small, the MSE shows a decreasing trend, indicating that the model performs better under these parameters and has a stronger fitting ability. However, as the σ value continues to increase, the MSE begins to fluctuate and rise, especially when the σ value reaches 2.4 and above, the MSE increases significantly, indicating that the model's fitting becomes worse and over-fitting or under-fitting may occur. This indicates that choosing the appropriate σ value is crucial, and either too large or too small will adversely affect the performance of the FKPGR model.

3.2. Real data analysis

When conducting actual data analysis, the Tecator dataset is selected. This dataset is a benchmark data for near - infrared spectroscopy analysis widely used in the field of food science. It records the chemical composition and spectral information of 215 meat samples. By capturing the absorbance characteristics of samples through spectroscopy technology, it can be used to study the functional relationships among spectral curves, fat content, moisture content, and protein content, providing a quantitative basis for food quality assessment. The correlation analysis among the contents of the three components in this dataset is shown in Figure 7 below. After performing functional principal component analysis on this dataset, data analysis is carried out using the FKGPR model and other traditional models. The initial basic values are set the same as those in the initial settings of the simulation experiment. The comparison diagrams and Table 5 of each model obtained from the experiment are as follows in Figure 8.

Table 5. Mean Squared Error and Standard Deviation of MSE of Each Model

Models	Mean	SD
FKGPR	0.0854	0.0150
RF	0.1434	0.0337
KSVR	0.2062	0.0445
GPR	0.8632	0.0856
XGBoost	0.1453	0.0342
GBDT	0.1270	0.0345

As can be analyzed from Figure 8, through the functional data analysis of the Tecator dataset, the FKGPR model has a relatively small and stable mean mean squared error, and it exhibits stronger robustness compared to traditional machine - learning models. Moreover, as can be seen from the data in Table 5, the FKGPR model has the lowest mean mean squared error and a relatively small standard deviation in the Tecator dataset, indicating that the model is relatively stable during the prediction process, with small errors and excellent performance. The experimental results obtained

from the real - world data are basically consistent with those in the simulation experiment, further demonstrating the rationality of the construction of the FKGPR model. It has more advantages than other models in functional data analysis.

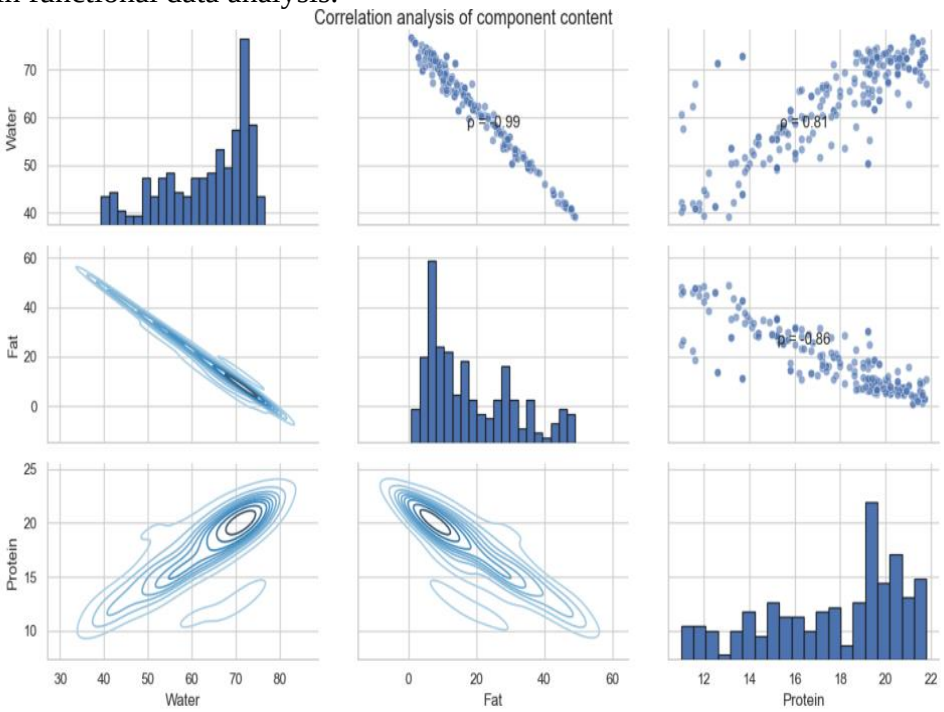

Figure 7: Correlation analysis among the contents of water, fat, and protein in the Tecator dataset

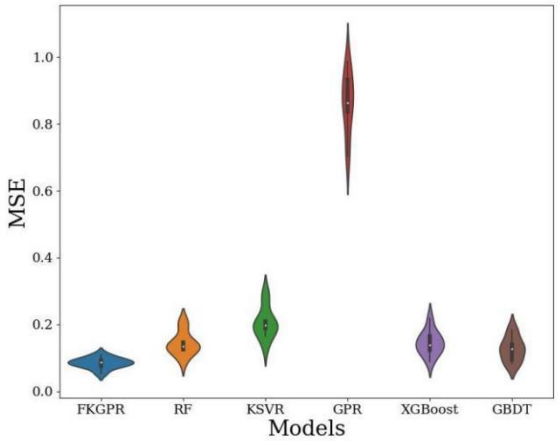

Figure 8. Comparison result of Mean Squared Error on the Tecator Dataset

4. Conclusions

To improve the performance of the functional regression model and address the issue of subjectively selecting the truncation number of functional data, this paper proposes a partially functional linear regression model based on the Stacking framework and the reproducing kernel theory. First, the kernel expansion method is employed to represent the non - linear input functions in the model, and functional principal component analysis (FPCA) is utilized to reduce the dimension of the original dataset, thus effectively decreasing the data dimension. Subsequently, multiple sub - models are trained in combination with K - fold cross - validation, and they are combined through the Stacking ensemble learning method. Finally, a new linear regression model is trained to enhance the prediction accuracy and robustness of the model. Finally, through performance comparisons with

some models based on principal component scores, the effectiveness of the proposed model is verified based on the mean squared error (MSE).

The further research directions mainly include the following two aspects: First, the current model assumes that the non - linear function data are Euclidean plane data, and in the future, it can be extended to the input of non - Euclidean plane data. Second, the Gaussian kernel expansion method is adopted to handle non - linear functions in this paper's model. In the future, the additive model based on kernel methods can be explored for further optimization.

References

[1] Wu, H. (2023). Robust variable selection in partial linear models with functional data and its application (Doctoral dissertation). Chongqing Technology and Business University.

[2] Jin, X. (2021). Several Methods of Functional Data Analysis. Modern Computer, 27(34),77-80.

[3] Liu, Z. (2020). Functional Data Classification Methods and Their Applications (Doctoral dissertation). Hefei University of Technology.

[4] Zhu, J. (2024). Estimation of High-Dimensional Functional Linear Models Based on Dynamic Principal Components (Doctoral dissertation). Zhejiang University of Finance and Economics.

[5] Wang, H., Huang, L., Wang, S. (2016). Generalized linear regression model based on functional data. Journal of Beijing University of Aeronautics and Astronautics, 42(1), 8-12.

[6] Wang, Q. (2020). Research and application of functional principal component analysis and functional linear regression models (Doctoral dissertation). Chongqing Technology and Business University.

[7] Huang, H., Liu, Y., Ma, Y., et al. (2024). Prediction of soluble solid content in mature apples based on visible/near-infrared spectroscopy and functional linear regression models. Spectroscopy and Spectral Analysis, 44(7), 1905-1912.

[8] Zhang, H., Zhu, Y. (2017). Triangular spline estimation for functional linear regression models. Journal of Yunnan University for Nationalities (Natural Science Edition), 26(6), 475-477.

[9] Jin, T. (2023). Change-point study of functional linear regression models based on Group Lasso (Master's thesis). Zhejiang University of Finance and Economics.

[10] Liu, X., Ma, H. (2023). Change-point testing in functional linear regression models. Applied Probability and Statistics, 39(4), 475-490.

[11] Yu, P., Du, J., Zhang, Z. (2017). Partial functional linear additive quantile regression models. Systems Science and Mathematical Sciences, 37(5), 1335-1350.

[12] Wang, D., et al. (2022). Functional linear regression with mixed predictors. Journal of Machine Learning Research, 23, 1-94.

[13] Yang, W., Qin, L. (2024). Functional partial linear models based on residual function principal components and their applications. Engineering Economics, 34(11), 33-45.

[14] Hu, Y. (2022). Robust estimation of partial functional linear regression models (Master's thesis). Guizhou University.

[15] Wen, L. (2022). Homogeneity of variance testing in partial functional linear regression models (Master's thesis). Beijing University of Technology.

[16] Zhu, R., Zou, G., Zhang, X. (2018). Model averaging methods for partial functional linear models. Systems Science and Mathematical Sciences, 38(7), 777-800.

[1]