Problem Chosen	2024 ShuWei Cup Summary Sheet	Team Control Number
B		2024090828988

Research on the Collaborative Estimation Method for Spatial Variables

Summary

This paper focuses on the research of collaborative estimation methods for spatial variables. In mathematical statistics, traditional sampling methods assume that data points are independent, suitable for scenarios where data follows an independent and identically distributed (IID) pattern. However, in spatial statistics, data often exhibits spatial dependence, meaning that observations at nearby locations are correlated. Based on this characteristic, Kriging interpolation is a popular method for predicting values at unobserved locations, particularly useful for sparse data, as it leverages the spatial structure and trends within the data.

Addressing the issue of high costs or difficulties in measuring target variables, the Co-Kriging algorithm solves this by using auxiliary variables that are easier to measure and related to the target variable, despite the increased complexity due to the calculation of cross-covariances. In this study, appropriate analytical methods such as Ordinary Kriging, Co-Kriging, and Neural Network Kriging algorithms were selected for the spatially correlated data characteristics. These methods can effectively predict the values at unobserved locations even with sparse data by fully utilizing the information of spatial structure and trends.

The paper specifically explores four issues:

1. Kriging interpolation was performed on the F1 target variable data, choosing the spherical model for interpolation. It was found that larger resampling sample sizes led to smaller estimation errors.

2. Covariates 1 and 4, with higher Bivariate Moran's I values, were selected as estimation covariates for the target variable.

3. The Co-Kriging algorithm showed decreasing mean square error (MSE) with increasing sampling rate until it plateaued, and overfitting occurred around a sampling rate of 0.06. In contrast, the Neural Network Kriging algorithm performed better at high sampling rates without overfitting and with lower error.

4. For the F2 target variable with limited data, Covariate 3 was cho.

Keywords: Kriging interpolation; Bivariate Moran's I; Co-Kriging algorithm; Neural Network Kriging;

Content

1. Introduction

1.1 Background

In mathematical statistics, traditional sampling methods assume data points are independent, which applies when data follows an independent and identically distributed (IID) pattern. However, in spatial statistics, data often shows spatial dependence, meaning observations at nearby locations are correlated. Spatial statistics models and predicts variables based on this correlation.

Kriging is a popular method in spatial estimation that predicts values at unobserved locations by considering spatial correlation. It is especially useful for sparse data, as it leverages the spatial structure and trends within the data.

In practice, different measurement techniques may be used for the same spatial variable. While these methods can introduce variability, the variables still maintain strong spatial correlation, supporting co-estimation. Co-estimation combines information from multiple variables to improve prediction accuracy.

A challenge arises when the measurement cost of the target variable is high or difficult. Co-Kriging addresses this by using auxiliary variables, which are easier to measure and correlated with the target variable. However, calculating cross-covariance between variables adds computational complexity, limiting its widespread use.

Recently, AI and machine learning have offered new methods to overcome some of the limitations of traditional geostatistics. Deep learning and ensemble techniques show promise in spatial variable modeling, improving accuracy and enabling efficient co-estimation, especially for large-scale datasets. These advances hold significant potential in fields like geology, mining, and urban planning.

1.2 Work

The problem asks us to research appropriate methods for the collaborative estimation of spatial attribute data given in the attachments:

·Problem 1: Use the data in Attachment 1 to study the variation patterns of one of the spatial variables (F1_target variable).

·Problem 2: Study the correlation between the target variable and collaborative variables using the data in Attachment 1. Select two collaborative variables as the estimating collaborative variables for the target variable.

·Problem 3: Using the data in Attachment 1 and the findings from Problem 2, select one or two collaborative variables and study the variation patterns of the spatial variable (F1_target variable).

·Problem 4: The target variable (F2_target variable) in Attachment 2 has insufficient sampling data. Select the optimal method from Problem 3 to estimate the trend of the target variable and present the results as a contouring map.

2. Problem analysis

2.1 Data analysis

The attachments 1 and 2 provide measurement data for four spatial attributes and one target variable within a designated rectangular area. This rectangular area covers a 266×266 grid, with each grid cell representing an area of 50 meters × 50 meters, totaling 70,756 grid points. Due to the high measurement cost of the target variable, which is significantly higher compared to other spatial attributes, our analysis objective is to explore the spatial distribution characteristics of the variables based on this data and optimize the sampling strategy to maximize cost-effectiveness.

Specifically, the F2 target variable was sampled at 1,000 grid points within the study area. Given the high measurement cost of the target variable, the number of sampling points for the target variable is relatively small compared to other spatial attributes.

For subsequent analysis, the data in the attachments need to be processed: first, the 266×266 data points in the .txt file of the attachment should be converted into a matrix format with 266 rows and 266 columns. Then, the positions of the data points for the F2 target variable need to be aligned with those of the other spatial attributes in order to facilitate further data analysis and comparison.

2.2 Analysis of problem 1

Problem 1 involves randomly resampling the F1 target variable and using these resampled values to estimate the spatial variable at unsampled locations. The estimation results should be shown on a contour map. Additionally, the relationship between sample size and estimation error should be explored by varying the resampling size. Given the spatial nature of the target variable, Kriging interpolation is used for estimation. The Mean Squared Error (MSE) is then applied to assess the estimation error, as it effectively measures the deviation between estimated and actual values.

2.3 Analysis of problem 2

Problem 2 involves using the data in Attachment 1 to examine the correlation between the target variable and covariates, and selecting two covariates for estimating the target variable. Since both the target variable and covariates are spatial variables with potential spatial dependence, we will apply Bivariate Moran's I to assess their spatial correlation. This method quantifies the relationship between two spatial variables, capturing their spatial autocorrelation—whether local variations in one variable are similar or dissimilar to those in the other. By calculating Bivariate Moran's I, we can evaluate the spatial relationship between the target variable and covariates, identifying clusters or extrapolation effects to guide variable selection and model development.

2.4 Analysis of problem 3

Problem 3 involves using the data from Attachment 1 and results from Question 2 to select one or two covariates. We will then perform random resampling of both the target variable and covariates to estimate the target variable at unsampled locations, presenting the results as contour maps. We will explore the relationship between sample size and estimation error and compare the effectiveness of two methods. In particular, we will use both the target variable's spatial trends and the covariates' distribution to improve estimation accuracy.

For method comparison, we will first use the Co-Kriging algorithm, a standard spatial interpolation method that incorporates covariate information. We will also compare it with the Deep Kriging algorithm, which integrates deep learning models to improve prediction accuracy through nonlinear modeling.

2.5 Analysis of problem 4

Problem 4 involves selecting the optimal method from Problem 3 to estimate the trend of the target variable with limited data from Attachment 2, and presenting the results as a contour map. We choose the method with the smallest error, the neural network Kriging, to predict the F2 target variable using F2 collaborative spatial variables. This method combines neural network architecture with Kriging interpolation, leveraging spatial data and correlation to overcome data limitations and provide an accurate, reliable prediction of the F2 target trend.

3. Symbol and Assumptions

3. 1 Symbol Description

Parameter	Meanings
x	Column value of spatial variables
y	Row value of spatial variables
z	Spatial variable value
$\hat{z}$	Predicted value of spatial variable

3.2 Fundamental assumptions

1. Assume that spatial variables are stationary, meaning their statistical properties (such as mean, variance, and covariance) do not change with spatial location, or at least can be considered stationary within the local scope of the study area.
2. Assume that different methods used to measure the target variable and covariates may introduce errors, but these errors are systematic and can be modeled.
3. Assume that the covariance between spatial variables can be adequately modeled using a Gaussian (or other appropriate) covariance function.
4. Assume that the provided datasets (Appendix 1 and Appendix 2) are of sufficient quality, with no significant outliers or missing data that would notably affect the analysis.
5. Assume that contour maps are an effective and useful method for visualizing spatial estimation results.
6. Assume that although the covariates may have different physical meanings or units, we assume that these variables can be appropriately transformed (e.g., standardized or scaled) to standardize their measurement units, so that they can be used in the joint estimation process.

4. Model

4.1 Preparation of data

Parameter	Meanings
I	Moran's I
C	Geary's C
n	Total sample size
w_{ij}	Spatial weight of variable z_i and z_j
z_i	Spatial variable value
$\bar{z}$	Average value of spatial variables

The data provided in the attachments is stored in text format, with each row of 266 data points organized into a 54-row by 5-column structure. In order to better utilize the data from the attachments and to enable the model to more effectively explore the spatial variable co-estimation method, we applied an algorithm to preprocess the data before formally solving the problem.

we read the data from each .txt file in Attachments 1 and 2 and converted them into datasets with 266 rows and 266 columns, respectively:

Chart 1 Processed dataset (showing only part of the data)

0	1	2	3	4	5	6	7	8	9
373.0228	372.1023	371.1602	370.1951	369.2138	368.2181	367.209	366.1871	365.1606	364.1306
372.3001	371.4718	370.6194	369.7416	368.8435	367.9277	366.9948	366.0463	365.0891	364.1261
371.6777	370.9422	370.1781	369.3887	368.5737	367.7356	366.8798	366.0056	365.1197	364.2241
371.1478	370.5042	369.8296	369.1265	368.3961	367.6382	366.8585	366.0593	365.2456	364.4197
370.7044	370.1535	369.5691	368.9527	368.3049	367.6277	366.9273	366.2039	365.4632	364.7072
370.3386	369.8817	369.3873	368.8571	368.294	367.7007	367.0788	366.4331	365.7659	365.0807
370.0482	369.6823	369.2783	368.8368	368.3599	367.8487	367.3087	366.7403	366.1492	365.5382
369.822	369.5508	369.2378	368.8846	368.4941	368.0695	367.611	367.1229	366.6092	366.0748
369.6533	369.478	369.258	368.9961	368.6946	368.3549	367.98	367.5746	367.1414	366.6854
369.5408	369.4616	369.3351	369.1649	368.9527	368.7015	368.4132	368.0919	367.7417	367.3663

Before proceeding with further questions, we must first conduct a spatial autocorrelation test on each spatial variable. This test evaluates the spatial dependence between observations, confirming whether the sampling method reflects the spatial distribution and dependence structure of the data. The results will provide a foundation for subsequent analysis, ensuring its validity and guiding model selection and parameter estimation to improve accuracy and reliability.

We use Moran's I index and Geary's C index as tools to test spatial autocorrelation. Moran's I is a statistical measure that assesses the degree of global spatial autocorrelation in spatial data. It quantifies the spatial dependence of the data by comparing the differences between each observation and its neighboring observations. The value of Moran's I typically ranges from -1 to +1. Positive values indicate the presence of positive spatial autocorrelation (i.e., similar values tend to cluster together), negative values suggest negative spatial autocorrelation (i.e., dissimilar values tend to cluster together), and a value of zero indicates a completely random spatial distribution.

$$I = \frac{n}{S_0} \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} w_{i,j} z_i z_j}{\sum_{i=1}^{n} {z_i}^2}$$

In contrast, Geary's C index focuses more on local spatial autocorrelation. It measures the differences between adjacent observations to evaluate the local spatial structure of the data. Geary's C ranges from 0 to 2, where values close to 0 indicate strong positive spatial autocorrelation, values approaching 2 suggest strong negative spatial autocorrelation, and a value of 1 indicates a random spatial distribution.

$$C = \frac{(n-1) \sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} (z_i - z_j)^2}{2 \sum_{i=1}^{n} \sum_{j \neq 1}^{n} w_{ij} \sum_{i=1}^{n} (z_i - \overline{z})^2}$$

These two indices each have their distinct characteristics. Moran's I emphasizes global spatial autocorrelation, while Geary's C is more concerned with local spatial dependence. By using both spatial autocorrelation tests in tandem, we can achieve a more comprehensive understanding of the spatial distribution and underlying spatial structure of the data, thereby providing more reliable and precise foundations for subsequent spatial analyses.

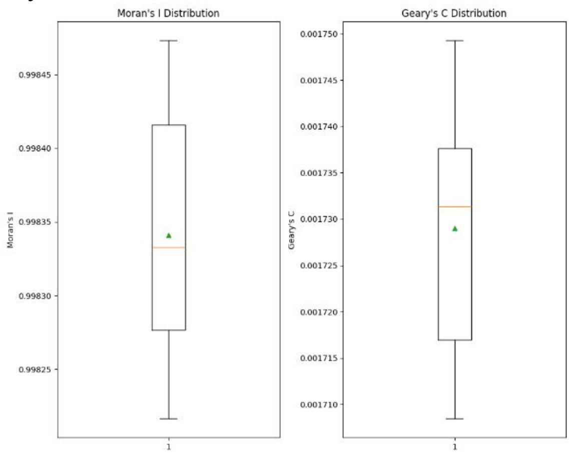

Figure 1 Moran's I and Geary's C

By conducting multiple random samplings and calculating the spatial autocorrelation statistics, we have verified the validity and representativeness of the sampling method. The results show that the values of Moran's I and Geary's C from different samplings are highly consistent, indicating that the sampling method can accurately reflect the overall spatial autocorrelation. This conclusion provides a reliable basis for further spatial data analysis.

4.2 Model of problem 1

Parameter	Meanings
x_0	Target spatial position x
x_i	Sampling points'patial position x
y_0	Target spatial position y
y_i	Sampling point's spatial position y
$\hat{z}_0$	Target predicted value
z_0	True target value
z_i	sampling value
h	Lag distance
a	Range parameter

$\gamma(h)$	Semivariogram value
C_0	Partial sill value
C	Nugget effect
n	Sample size
$F(z)$	Cumulative Probability Density Function of the Normal Distribution

Since the data has spatial attributes, spatial location factors must be considered, as spatial dependence and distance decay affect relationships between data points. Points closer together have a stronger influence, while those farther apart have a weaker influence. To better handle spatial data, we convert each element a_{ij} from a list to a structured dictionary format: {aij: value, x: location_x, y: location_y}. This transformation clarifies spatial relationships and provides essential references for further analysis.

[2]it is necessary to examine the distribution of the data, analyze the trends, and so on. The exploratory spatial data analysis module provides a series of data analysis tools. The Kriging method requires that, to some extent, all the data exhibit the same variability and follow a normal distribution. We used normal Q-Q plots and AD test to test the distribution characteristics of the data.

Anderson-Darling(AD) test formula:

$$AD =- n - S$$

$$S = \sum_{i=1}^{n} \frac{2i-1}{n}\left[\ln\left(F(z_i)\right) + \ln\left(1 - F(z_{n-i+1})\right)\right]$$

Next, resampling is performed on the overall sample. The purpose of resampling is to construct a more representative subset by resampling the existing data, thereby improving the accuracy and reliability of spatial analysis results.

After the resampling the Kriging interpolation algorithm is used to predict the values at unsampled locations. [1]The Kriging algorithm is an optimal weighted method based on spatial autocorrelation. It generates the best predicted values for unobserved regions by considering the spatial dependence structure and distance relationships between spatial data points. Kriging interpolation not only provides point estimates but also gives the error range for each predicted value, thereby providing more information for spatial prediction.

Spherical Model:

$$\gamma(h)=\begin{cases}C_0+C\left(\frac{3h}{2a}-\frac{h^3}{2a^3}\right),0\le h\le a\\ C_0+C,h>a\end{cases}$$

Linear Model:

$$\gamma(h)=C_0+C\cdot\frac{h}{a}$$

Gaussian Model:

$$\gamma(h)=C_0+C(1-e^{\left(-\left(\frac{h}{a}\right)^2\right)})$$

Kriging interpolation formula:

$$\widehat{z_0}(x_0,y_0)=\sum_{i=1}^{n}\lambda_i z_i(x_i,y_i)$$

[3] λ_i is the weight coefficient, a set of optimal coefficients that can satisfy the smallest difference between the estimated value $\hat{z}_0$ and the true value z_0 at the point (x_0,y_0):

$$minVar(\widehat{z_0}-z_0)$$

$$\sum_{i=1}^{n}\lambda_i=1$$

$$J=Var(\widehat{z_0}-z_0)$$

Expand J:

$$J=\sum_{i=1}^{n}\sum_{j=0}^{n}\lambda_i\lambda_j Cov(z_iz_j)-2\sum_{i=1}^{n}\lambda_i\,Cov(z_i,z_0)+Cov(z_0,z_0)$$

Our goal is to find the set of (λ_i) that minimizes (J), where (J) is a function of (λ_i). Therefore, we can directly take the partial derivative of (J) with respect to (λ_i) and set it equal to zero:

$$\frac{\partial J}{\partial\lambda_i}=0;i=1,2,\cdots,n$$

Next, to further explore the effect of resampling sample size on estimation accuracy, the resampling sample size will be adjusted, and the resampling process will be repeated, followed by applying the Kriging interpolation method to estimate the values at unsampled points. This process will allow an examination of how the interpolation results change under different sample sizes and assess the fluctuations and trends of estimation errors under varying sample size conditions.

In the final analysis phase, the Mean Squared Error (MSE) will be used as the standard metric to measure the estimation errors. The MSE is a commonly used error metric in statistics that reflects the degree of deviation between the true values and the estimated values by calculating the average of the squared differences between them. By calculating the MSE for different resampling sample sizes, the relationship between sample size and estimation error can be systematically analyzed. Finally, a graph will be drawn to show the relationship between resampling sample size and MSE, providing a basis for optimizing sampling strategies in future work.

$$MSE(\hat{z}) = E(\hat{z} - z)^2$$

4.3 Model of problem 2

Parameter	Meanings
I_{zv}	Bivariate Moran's I
N	Sample size
W	Sum of weights
w_{ij}	Weight
z_i	Target value
$\bar{z}$	Mean of target
v_i	Covariate value
$\bar{v}$	Mean of covariate

The ordinary Moran's I is primarily used to measure the spatial autocorrelation of a single variable, that is, the correlation of the same variable across space (for example, whether the value of a variable in one area is related to the values in neighboring areas). The Bivariate Moran's I, on the other hand, is used to measure the spatial correlation between two variables, and is suitable for analyzing the spatial correlation between the target variable and explanatory variables. Therefore, we calculate the Bivariate Moran's I between the target variable and each of the four explanatory variables, and the results reflect the spatial correlation between the target variable and each explanatory variable.

Bivariate Moran’s I:

$$I_{zv} = \frac{N}{W} \cdot \frac{\sum_{i=1}^{N} \sum_{j=1}^{N} w_{ij} (z_i - \bar{z})(v_i - \bar{v})}{\sum_{i=1}^{N} (z_i - \bar{z})^2}$$

Based on the results of the Bivariate Moran's I, we select the two explanatory variables with strong spatial correlation with the target variable as the explanatory variables for estimating the target variable.

4.4 Model of problem 3

Parameter	Meanings
$\hat{z}_0$	Target predicted value
x_0	Target spatial position x
y_0	Target spatial position y
z_{ji}	Spatial covariates z_{ji}
x_{ji}	Spatial covariatesl x_{ji}
y_{ji}	Spatial covariates y_{ji}
n	Sample size
w_{ij}	Spatial location weights
M	Total classes
σ_b^2	Base variance
σ_w^2	Weight variance
$\widehat{f_{UQ}}(y\|x)$	Estimated conditional probability density function
$\|T_m\|$	Size of interval or region m
$\widehat{p_m}(s_0)$	Estimated probability of interval m at reference point

$z(s_n)$	Actual label
c_m	Class threshold
$F(c_m;s_n)$	Predicted probability

For Problem 3, we selected two covariates from Problem 2 that exhibit a high correlation with the target variable, and these were chosen as the key variables for further analysis. In addressing the relationship between these covariates and the target variable, we first performed resampling on the data. Subsequently, we applied the Co-Kriging method to predict the target variable. The Co-Kriging method is a multivariate geostatistical approach that takes into account the spatial correlation between different variables, enabling a more accurate prediction of the target variable by leveraging this spatial relationship.

Co-Kriging algorithm:

$$\widehat{z_0}(x_0,y_0)=\sum_{i=1}^{n}a_{1_i}z_1(x_i,y_i)+\sum_{i=1}^{n}a_{2_i}z_2(x_i,y_i)$$

We introduced "Neural Network Kriging," a predictive approach that combines traditional Kriging with deep learning. This method uses neural networks to model complex nonlinear relationships between covariates and the target variable, improving predictive accuracy. The deep learning model extracts features from the covariates, while Co-Kriging performs spatial interpolation and prediction. This integration of geostatistics and machine learning enhances Kriging's performance with nonlinear and high-dimensional data, leading to more accurate predictions.

[6]Neural Network Kriging-induced covariance function recursive formula:

$$C^{l}\left(z_{ji},z_{ji}'\right)=\sigma_b^2+\sigma_w^2F_{\psi}\left(C^{l-1}(z_{1i}',z_{2i}\ \ ),C^{l-1}(z_{1i}\ \ ),C^{l-1}(z_{2i}',z_{3i}')\right)$$

Neural Network Kriging density prediction formula:

$$\widehat{f_{UQ}}(y|x)=\sum_{m=1}^{M+1}\frac{\widehat{p_m}(s_0)}{|T_m|}\mathbb{1}\{y\in T_m\}$$

Joint binary cross-entropy loss function JBCE formula:

$$\mathrm{JBCE}=\sum_{n=1}^{N}\sum_{m=1}^{M}[\mathbb{1}\{z(s_n)\le c_m\}\log\{F(c_m;s_n)\}$$
$$+\mathbb{1}\{z(s_n)>c_m\}\log\{1-F(c_m;s_n)\}]$$

4.5 Model of problem 4

Since the observational data for variable F2 consists of only 1,000 samples, in order to overcome the limitations posed by the small dataset, the analysis first combines F2 with covariates that share the same spatial values, and applies Bivariate Moran's I to assess spatial autocorrelation. The preliminary results indicate that cor3 performs best in the correlation analysis. However, since the correlation between cor3 and F2 is only 0.09, which is relatively weak, it is hypothesized that this low correlation might be due to the small sample size.

To enhance the accuracy and stability of the predictions, Ordinary Kriging interpolation is employed to spatially interpolate the F2 variable. Through Kriging interpolation, the entire region of F2 data is populated, resulting in a more complete spatial distribution. The updated F2 data is then used in conjunction with multiple covariates to recalculate Moran's I, assessing spatial autocorrelation among the covariates, and ultimately selecting cor3 as the most spatially correlated covariate with F2.

Next, the complete F2 dataset and corresponding cor3 values are used for deep Kriging interpolation to further optimize the spatial interpolation results. Finally, the interpolated cor3 image is compared with the original F2 image, as the two are strongly correlated spatially, thereby validating the quality and consistency of the interpolated data through visual comparison and ensuring the reliability and precision of the research findings.

5.Test the Models

5.1 Test of problem 1

Conduct a normality test on the target variable:

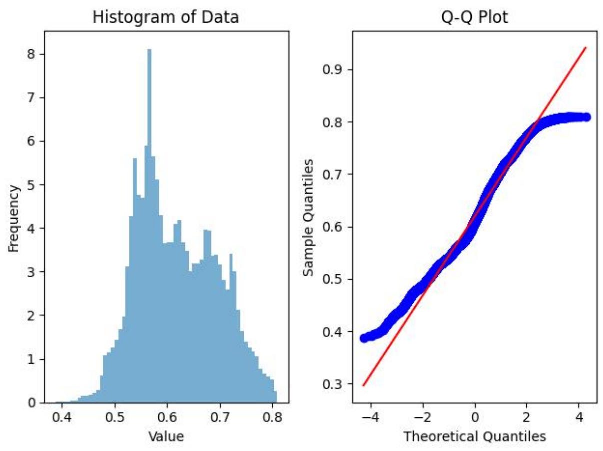

Figure 2 Histogram of Data and Q-Q plot

The histogram of data and the Q-Q plot indicate that the data of the F1 target variable is approximately normal. We can use the Kriging interpolation method.

In addition, we also carried out the Anderson - Darling (AD) test:

AndersonResult(statistic=590.8416980968614, critical_values=array([0.576, 0.656, 0.787, 0.918, 1.092]), significance_level=array([15. , 10. , 5. , 2.5, 1.]), fit_result= params: FitParams(loc=0.6185998558426142, scale=0.07630380214465288) success: True message: '`anderson` successfully fit the distribution to the data.')

We compared three types of variogram parameters: the spherical model, the linear model, and the Gaussian model. Taking the mean square error after comparing the interpolated data of the unsampled data with the original true data as the selection criterion, we selected the variogram that minimizes the mean square error as the parameter for ordinary Kriging sampling.

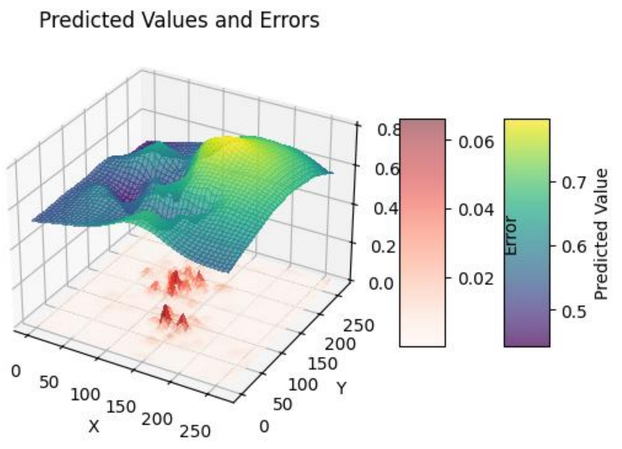

Figure 3 3D contour map and error distribution when the parameter is the Spherical

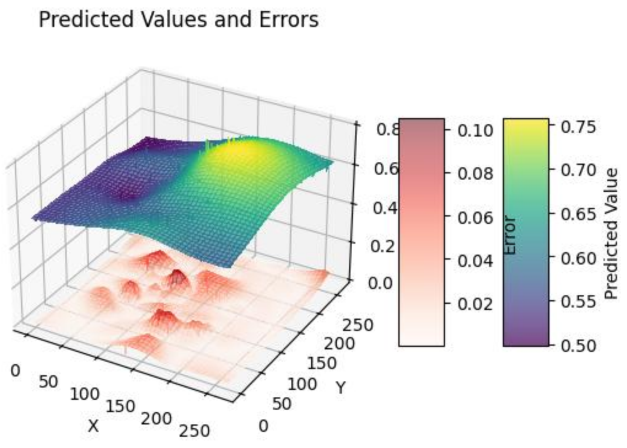

Figure 4 3D contour map and error distribution when the parameter is the Linear

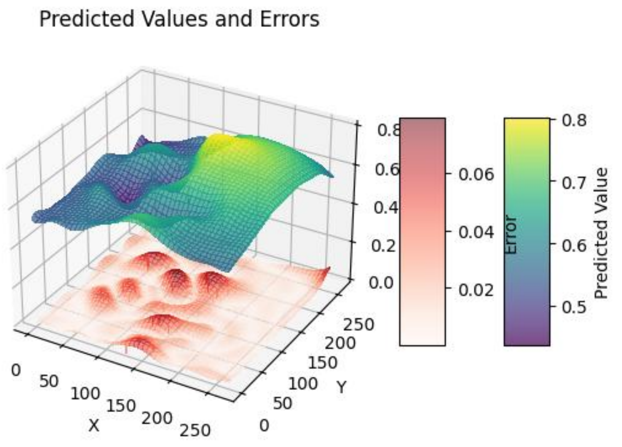

Figure 5 3D contour map and error distribution when the parameter is the Gaussian

Finally, the spherical model was selected to perform ordinary Kriging interpolation to obtain the contour map after interpolation, as shown in the figure.

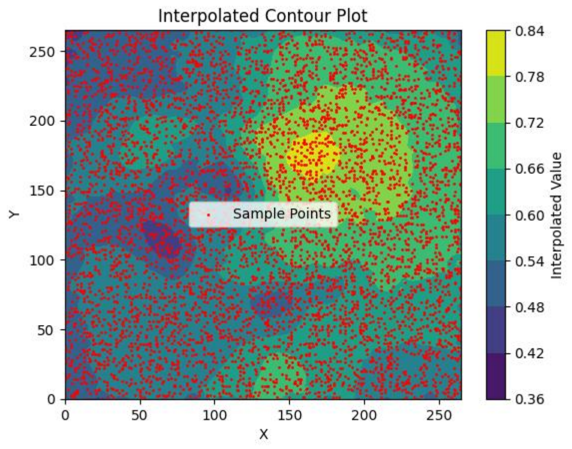

Figure 6 the contour map after interpolation

By varying the resampling sample size, the relationship between the sample size and the estimation error needs to be explored.

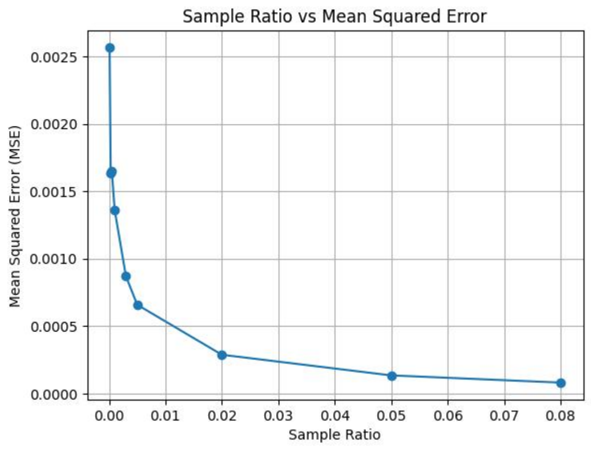

Figure 7 The Relationship between Sampling Rate (Sample Size) and Estimation Error

It can be seen from the figure that the larger the resampling sample size is, the smaller the error of Kriging interpolation estimation will be.

5.2 Test of problem 2

It can be seen that the Moran's I of covariate 1 reached 0.764336170463134, and that of covariate 4 reached 0.3428830246170835. The indices of the other two variables are relatively small. According to the requirements of the task, we finally selected covariates 1 and 4 as the estimation covariates for the target variable.

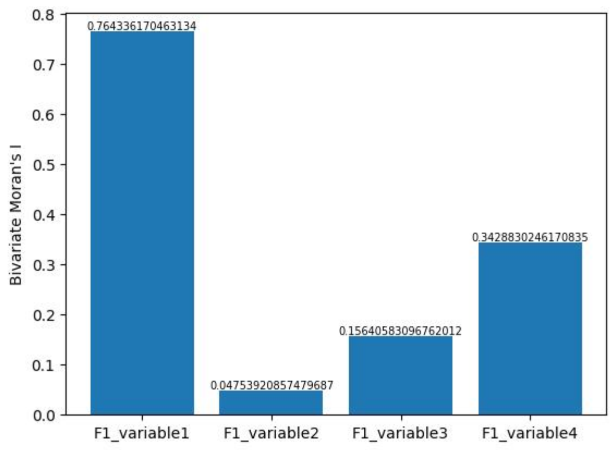

Figure 8 The Bivariate Moran's I between the F2 target variable and the covariates

5.3 Test of problem 3

Based on the results of Question 2, we obtained Covariates 1 and 4 to estimate the target variable.

Unlike the ordinary Kriging algorithm, the co-Kriging algorithm can integrate the information among multiple variables more effectively, fully taking into account the spatial correlations and synergistic effects of different variables.

We first selected the co-Kriging algorithm to predict the target variable after resampling. The drawn contour map is shown as follows:

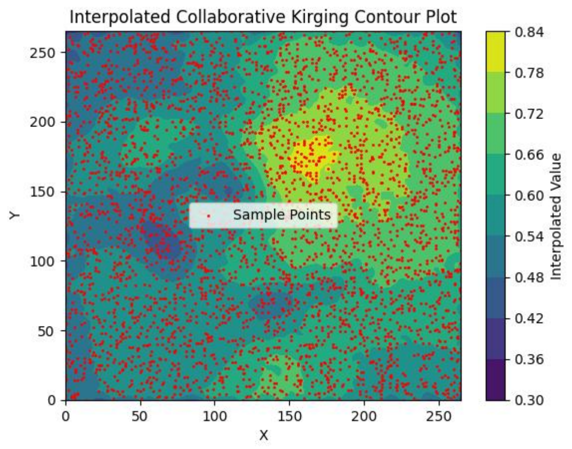

Figure 9 The contour map of the target variable estimated by the co-kriging method

3D contour map:

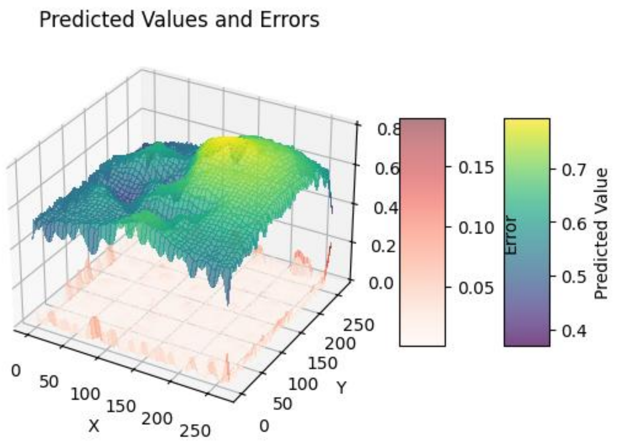

Figure 10 The 3D contour map of the target variable estimated by the co-kriging method

Next, we adjusted the parameter: sampling rate. By changing the sample size, we explored the relationship between the sample size and the mean square error of the estimation, and obtained the following line chart:

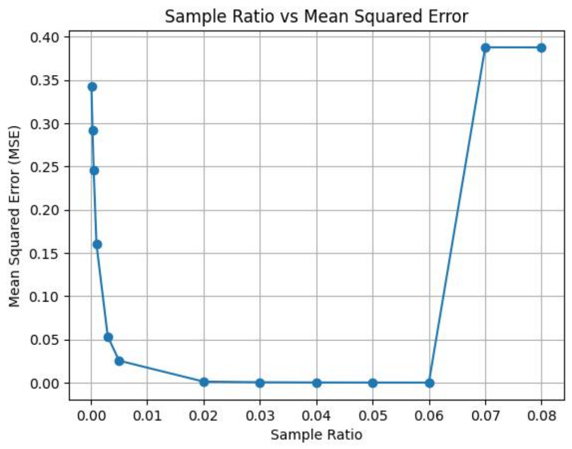

Figure 11 The Relationship between Sampling Rate (Sample Size) and the Estimation Error of Co - Kriging Method

As the sampling rate increases, the mean square error (MSE) initially decreases, but the rate of decrease slows over time. With a low sampling rate, the small sample size leads to a higher MSE. As the rate increases, more samples improve accuracy, reducing MSE. However, beyond a certain point, the improvement becomes marginal, and further increases in the sampling rate yield diminishing returns. At a sampling rate of around 0.06, the MSE rises sharply, indicating overfitting.

According to the requirements, we then selected the neural network Kriging algorithm to perform interpolation prediction on the target variable with respect to the two covariates. We explored the relationship between the sample size and the mean square error of the estimation, and obtained the following line chart:

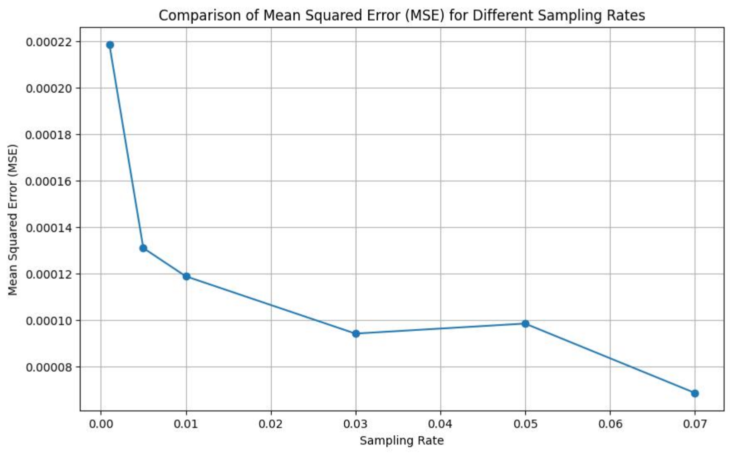

Figure 12 The Relationship between Sampling Rate (Sample Size) and the Estimation Error of the neural network Kriging algorithm

Overall, as the sampling rate changes, the Mean Square Error (MSE) is relatively high at a low sampling rate because the insufficient sample size leads to inaccurate model estimations. As the sampling rate increases, the MSE gradually decreases, indicating that more samples are helpful in improving the accuracy of the model.

Compared with the results of the co-Kriging algorithm, as the sampling rate increases, the neural network Kriging algorithm does not show the problem of overfitting, which benefits from the help of deep learning and neural networks. Moreover, when the sampling rate is relatively high, under the same sampling rate, the MSE of the neural network Kriging algorithm is significantly lower than that of the co-Kriging algorithm. Therefore, we choose the neural network Kriging algorithm as the optimal one among the two.

5.4 Test of problem 4

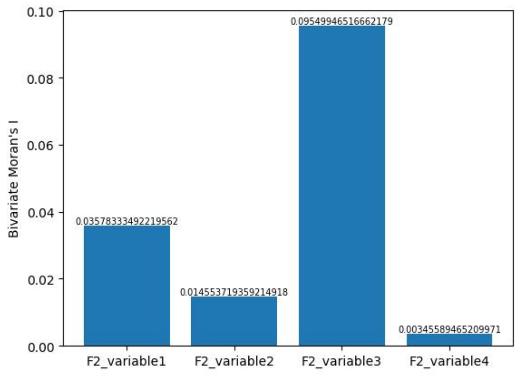

Figure 13 The Moran's I between the F2 target variable and the covariates (left - joined processing)

Considering that the correlation values of other covariates are relatively low, we only selected Covariate 3.

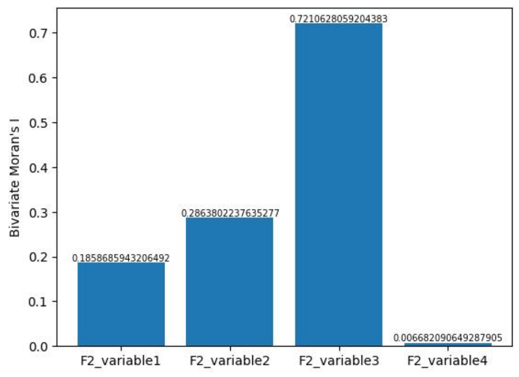

Figure 14 The Moran's I between the F2 target variable and the covariates (right-joined processing)

We visualize the data distribution situation of Covariate 3:

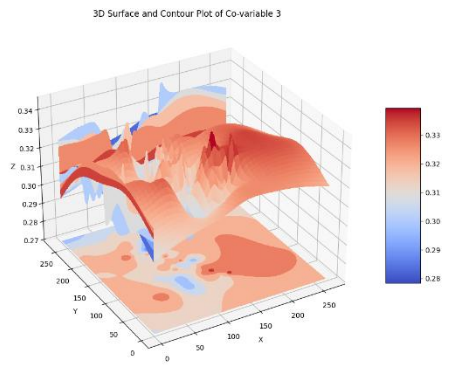

Figure 15 Contour map of Covariate 3

Next, we will adopt the deep Kriging algorithm to perform interpolation prediction on the target variable by utilizing Covariate 3:

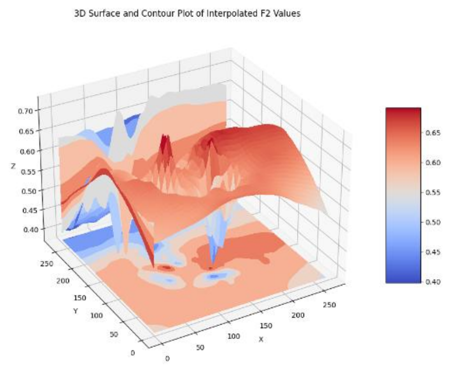

Figure 16 Contour map of the target variable

Similar to the distribution situation of Covariate 3 above, it indicates that there is no extremely large error in our predicted results.

6.Sensitivity Analysis

6.1 Analysis of problem 1

Sensitivity of Sample Size: By changing the sample size of resampling, the impact on the results of Kriging interpolation estimation was observed. In the testing part of Question 1, the mean square error (MSE) under different sample sizes was calculated. The results show that as the sample size increases, the error of Kriging interpolation estimation gradually decreases. This indicates that the sample size has a significant impact on the accuracy of the model. The larger the sample size, the more accurately the model can capture the characteristics of spatial variables, thereby reducing the estimation error.

Sensitivity of Variogram Model Parameters: The parameters of three variogram models, namely the spherical model, the linear model, and the Gaussian model, were compared. Taking the mean square error of the interpolated data with the original real data as the selection criterion, it was found that the spherical model performed best in this problem. This implies that different variogram models have different degrees of impact on the estimation results. The selection of the model needs to be determined according to the specific characteristics of the data and the purpose of the analysis. Inappropriate model parameters may lead to relatively large estimation errors.

6.2 Analysis of problem 2

Sensitivity of Covariate Selection: The Bivariate Moran's I between the target variable and the four covariates was calculated. Based on the results, two covariates were selected as the estimating covariates. The Moran's I of Covariate 1 reached 0.764336170463134, and that of Covariate 4 reached 0.3428830246170835, while the indices of the other two variables were relatively small.

6.3 Analysis of problem 3

Sensitivity of Sampling Rate (Co-Kriging Algorithm): When using the Co-Kriging algorithm to predict the target variable, the sampling rate (i.e., sample size) was adjusted to explore the relationship between it and the estimated mean square error. The results showed that as the sampling rate increased, the mean square error decreased gradually at first and then the rate of decrease slowed down. When the sampling rate reached approximately 0.06, the MSE increased sharply, and an overfitting problem occurred. This indicates that the Co-Kriging algorithm is relatively sensitive to the sampling rate, and an appropriate sampling rate range is crucial for ensuring the accuracy of the model.

6.4 Analysis of problem 4

The covariates were mainly selected. By calculating the Moran's I between the F2 target variable and each covariate, it was found that the correlation values of other covariates were relatively low, and only Covariate 3 was selected. Other sensitivity analysis contents were not clearly mentioned and might need further research or supplementation. Overall, sensitivity analysis helps to deeply understand the impact of various factors in the model on the results, provides an important reference basis for model optimization and practical application, and assists researchers in selecting appropriate parameters and methods according to data characteristics and requirements, thereby improving the model performance and prediction accuracy.

7.Strengths and Weakness

7.1 Strengths

1.For the characteristics of spatial variables, appropriate analysis methods have been selected. For example, when dealing with data with spatial correlation, Kriging interpolation algorithms (including Ordinary Kriging, Co-Kriging, and Neural Network Kriging, etc.) are adopted. These methods can make full use of the information of spatial structure and trends, and can effectively predict the values of unobserved locations even when the data is sparse.

2.The Bivariate Moran's I is used to quantify the spatial correlation between the target variable and the covariates, providing a scientific basis for variable selection and model construction. It can accurately screen out the covariates that have a strong correlation with the target variable, which helps to improve the prediction accuracy of the model.

3.Detailed preprocessing was carried out on the data in the attachment, including converting the data from text format to matrix format to make it easier to analyze and process.

7.2 Weakness

1.The neural network Kriging algorithm has high prediction accuracy, but the deep learning model is complex and has poor interpretability.

8.Conclusion

In this study, we focused on the collaborative estimation of spatial variables. The main results are as follows:

Problem 1: The F1 target variable data was approximately normal, and the spherical model was selected for Kriging interpolation. Larger resampling sample sizes led to smaller estimation errors.

Problem 2: Covariates 1 and 4, with relatively high Bivariate Moran's I values, were chosen as estimation covariates for the target variable.

Problem 3: The co-Kriging algorithm initially showed decreasing mean square error with increasing sampling rate until overfitting at around 0.06. The neural network Kriging algorithm outperformed it with no overfitting and lower error at high sampling rates.

Problem 4: For the F2 target variable with limited data, Covariate 3 was selected, and the deep Kriging algorithm was used to interpolate it, yielding reasonable results.

Overall, the research effectively addressed the spatial variable estimation problems, and the methods and algorithms employed provided valuable insights and techniques for future applications.

References

[1] DING Ziwei, LIU Jiang, WANG Xiaoyong, CHANG Maomao, LIAO Jinglong. Three-dimensional geological modeling technology based on PSO-Kriging algorithm[J]. COAL ENGINEE RING, 2024, 56(10):82-89.
[2] Xiao Juanjuan, Zhang Jinfen, Wu Da, etc. The spatial interpolation of polar sea ice density based on the Kkin method[J]. Journal of Dalian Maritime University, 2023,49(1) : 66-74.
[3] xg1990, "The Principle and Formula Derivation of Kriging Interpolation," https://xg1990.com/blog/archives/222, November 17, 2024.
[4] Jin Guodong, Liu Yancong, Niu Wenjie. Comparison of distance-weighted inverse interpolation and Crekin interpolation [J]. Journal of Changchun University of Technology (Natural Science Edition), 2003 (03): 53-57.
[5] Jiang Xin, Tuo Xianguo, Liu Bingqi, et al. Coordinated Kerry valuation of karma copper polymetallic deposit reserves in Tibet [J]. Exploration Technology, 2015,37 (03): 372-378.
[6] Li, Yuxiao, Ying Sun and Brian J. Reich. “DeepKriging: Spatially Dependent Deep Neural Networks for Spatial Prediction.” *ArXiv* abs/2007.11972 (2020): n. pag.

Appendix

```
#Import thr required libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import scipy.stats as stats
from pysal.lib import weights
from esda import Moran, Geary
from sklearn.neighbors import kneighbors_graph
from mpl_toolkits.mplot3d import Axes3D
import gstools as gs
from sklearn.metrics import mean_squared_error,mean_absolute_error
from pykrige.ok import OrdinaryKriging

from scipy.spatial.distance import cdist
from scipy.optimize import curve_fit
from sklearn.preprocessing import StandardScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.optimizers import SGD
import libpysal as lp
from esda.moran import Moran_BV
from scipy.interpolate import griddata
import pysal as ps

from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense, Dropout, BatchNormalization
from sklearn.model_selection import train_test_split
import tensorflow as tf
from matplotlib import cm

#Convert data to dataframe
def read_data_to_df(file_path, group_size=266, skip_rows=6):

    all_values = []

    with open(file_path, 'r') as file:
        for _ in range(skip_rows):
            next(file)
        for line in file:
            values = [float(val) for val in line.strip().split()]
            all_values.extend(values)

    num_columns = len(all_values) // group_size

    if len(all_values) % group_size != 0:
        raise ValueError(f"error")

    df = pd.DataFrame(columns=range(num_columns))

    for i in range(0, len(all_values), group_size):
        col_index = i // group_size
        df[col_index] = all_values[i:i + group_size]

    df = df.T

    return df


file_paths = [
    'D:/ShuweiCup/Attachment 1/F1_target_variable.txt',
    'D:/ShuweiCup/Attachment 1/F1_collaborative_variable1.txt',
    'D:/ShuweiCup/Attachment 1/F1_collaborative_variable2.txt',
    'D:/ShuweiCup/Attachment 1/F1_collaborative_variable3.txt',
    'D:/ShuweiCup/Attachment 1/F1_collaborative_variable4.txt',
    'D:/ShuweiCup/Attachment 2/F2_collaborative_variable1.txt',
    'D:/ShuweiCup/Attachment 2/F2_collaborative_variable2.txt',
    'D:/ShuweiCup/Attachment 2/F2_collaborative_variable3.txt',
    'D:/ShuweiCup/Attachment 2/F2_collaborative_variable4.txt'
]

for file_path in file_paths:
    df = read_data_to_df(file_path, skip_rows=6)

    file_name = file_path.split('\\')[-1].split('.')[0]

    print(f"Processing file: {file_name}")
    print(df.head())

    output_file = f'{file_name}_output.csv'
    df.to_csv(output_file, index=False)
    print(f"Saved to: {output_file}\n")
```

Question 1

```
F1_target_variable = pd.read_csv("D:/ShuweiCup/Attachment 1/F1_target_variable.csv")
num_columns = F1_target_variable.shape[1]
new_column_names = ['V' + str(i) for i in range(num_columns)]
F1_target_variable.columns = new_column_names

#contour map of original data:F1 target

rows, cols = F1_target_variable.shape

X, Y = np.meshgrid(np.arange(cols), np.arange(rows))
Z = F1_target_variable.values

grid_x, grid_y = np.mgrid[0:266, 0:266]
grid_z = np.zeros((266, 266))
contour = plt.contourf(X,Y,Z, cmap='viridis')
plt.colorbar(contour, label='Interpolated Value')
plt.xlabel('X')
plt.ylabel('Y')
plt.title('Original Data Contour Plot')
plt.legend()
plt.show()

#3D contour map

fig = plt.figure(figsize=(10, 6))
ax = fig.add_subplot(111, projection='3d')

contour = ax.plot_surface(X, Y, Z, cmap='viridis', edgecolor='none')

fig.colorbar(contour, shrink=0.5, aspect=5, label='Value')

ax.set_title('Original Data 3D Contour Plot')
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Value')

plt.show()

#Autocorrelation sampling test

sample_size = 40000
```

```
num_samples = 10
k = 10

autocorr_results = []

for i in range(num_samples):
    df_sampled = dic_df_F1_target.sample(n=sample_size, random_state=i)

    coordinates = df_sampled[['x', 'y']].values
    knn_graph = kneighbors_graph(coordinates, k, mode='connectivity',
include_self=True)

    w = weights.W.from_sparse(knn_graph)

    w.transform = 'r'

    values = df_sampled['aij'].values

    #calculate Moran's I
    moran = Moran(values, w)
    moran_I = moran.I
    moran_p = moran.p_sim

    #calculate Geary's C
    geary = Geary(values, w)
    geary_C = geary.C
    geary_p = geary.p_sim

    autocorr_results.append({
        'Sample': i + 1,
        'Moran\'s I': moran_I,
        'Moran P-value': moran_p,
        'Geary\'s C': geary_C,
        'Geary P-value': geary_p
    })


#Resample

sample_ratio = 0.01
np.random.seed(0)

coordinates = [(d['x'], d['y']) for d in dic_list_F1_target]
```

```
sample_indices = np.random.choice(len(coordinates), int(sample_ratio * len(coordinates)), replace=False)
sample_coordinates = [coordinates[i] for i in sample_indices]

train_data = [d for d in dic_list_F1_target if (d['x'], d['y']) in sample_coordinates]
test_data = [d for d in dic_list_F1_target if (d['x'], d['y']) not in sample_coordinates]

for d in train_data:
    d['type'] = 'sample'

X_train = np.array([[d['x'], d['y']] for d in train_data], dtype=float)
y_train = np.array([d['aij'] for d in train_data], dtype=float)
X_test = np.array([[d['x'], d['y']] for d in test_data], dtype=float)
y_test = np.array([d['aij'] for d in test_data], dtype=float)

#Compare the differences in ordinary kriging differences

ok_spherical = OrdinaryKriging(X_train[:, 0], X_train[:, 1], y_train, variogram_model='spherical')

ok_linear = OrdinaryKriging(X_train[:, 0], X_train[:, 1], y_train, variogram_model='linear')

ok_gaussian = OrdinaryKriging(X_train[:, 0], X_train[:, 1], y_train, variogram_model='gaussian')

def calculate_rmse(y_true, y_pred):
    return mean_squared_error(y_true, y_pred, squared=False)

z_spherical, _ = ok_spherical.execute('points', X_test[:, 0], X_test[:, 1])
z_linear, _ = ok_linear.execute('points', X_test[:, 0], X_test[:, 1])
z_gaussian, _ = ok_gaussian.execute('points', X_test[:, 0], X_test[:, 1])

#calculate RMSE
rmse_spherical = calculate_rmse(y_test, z_spherical)
rmse_linear = calculate_rmse(y_test, z_linear)
rmse_gaussian = calculate_rmse(y_test, z_gaussian)

#
Question 2
#Bivariate global Moran index

def create_spatial_weights(df):
    coordinates = [(d['x'], d['y']) for d in df]
    w = lp.weights.DistanceBand.from_array(coordinates, threshold=1.5)
    return w

w = create_spatial_weights(dic_list_F1_target)

def calculate_bivariate_moran(target, collaborative_variable, w):

    target_flat = target.values.flatten()
    collaborative_variable_flat = collaborative_variable.values.flatten()

    moran = Moran_BV(target_flat, collaborative_variable_flat, w)
    return moran.I, moran.p_sim

morans = []
name_list                                                                =
[F1_collaborative_variable1,F1_collaborative_variable2,F1_collaborative_variable3,
F1_collaborative_variable4]
for var in name_list:
    moran_I, p_value = calculate_bivariate_moran(F1_target_variable, var, w)
    morans.append((moran_I, p_value))

moran_I_list = []

for i, (moran_I, p_value) in enumerate(morans):
    print(f"F1 Collaborative Variable {i + 1} - Bivariate Moran's I: {moran_I},
p-value: {p_value}")
    moran_I_list.append(moran_I)

# Choose the largest bivariate Moran index
best_index = np.argmax([moran_I for moran_I, _ in morans])
best_moran_I, best_p_value = morans[best_index]
print(f"F1 Best Collaborative Variable: {best_index + 1} - Bivariate Moran's I:
{best_moran_I}, p-value: {best_p_value}")

#Collaborative kriging algorithm

sample_ratio = 0.05

# Construct a joint covariance matrix
n_sampled = len(sample_coordinates)
n_unsampled = len(X_test)
total_vars = 3

distances = np.linalg.norm(X_train[:, np.newaxis] - X_train[np.newaxis, :], axis=2)
```

```
joint_cov_matrix = np.zeros((total_vars * n_sampled, total_vars * n_sampled))
joint_cov_matrix[:n_sampled, :n_sampled] = model_target.covariance(distances)
joint_cov_matrix[:n_sampled, n_sampled:2*n_sampled] = cross_corr_1 *
model_target.covariance(distances)
joint_cov_matrix[:n_sampled, 2*n_sampled:] = cross_corr_4 *
model_target.covariance(distances)
joint_cov_matrix[n_sampled:2*n_sampled, :n_sampled] = cross_corr_1 *
model_target.covariance(distances)
joint_cov_matrix[n_sampled:2*n_sampled, n_sampled:2*n_sampled] =
model_co_var1.covariance(distances)
joint_cov_matrix[n_sampled:2*n_sampled, 2*n_sampled:] = cross_corr_1 *
cross_corr_4 * model_target.covariance(distances)
joint_cov_matrix[2*n_sampled:, :n_sampled] = cross_corr_4 *
model_target.covariance(distances)
joint_cov_matrix[2*n_sampled:, n_sampled:2*n_sampled] = cross_corr_1 *
cross_corr_4 * model_target.covariance(distances)
joint_cov_matrix[2*n_sampled:, 2*n_sampled:] =
model_co_var4.covariance(distances)

# Construct covariance vectors
cov_vector = np.zeros((total_vars * n_sampled, n_unsampled))
for i in range(n_unsampled):
    h = np.linalg.norm(X_train - X_test[i], axis=1)
    cov_vector[:n_sampled, i] = model_target.covariance(h)

    cov_vector[n_sampled:2*n_sampled, i] = cross_corr_1 *
model_target.covariance(h)

    cov_vector[2*n_sampled:, i] = cross_corr_4 * model_target.covariance(h)

cond_values = np.hstack([y_train, F1_collaborative_variable1_values_sampled,
F1_collaborative_variable4_values_sampled])

weights = np.linalg.solve(joint_cov_matrix, cov_vector)

predicted_values = np.dot(weights.T, cond_values)


dic_list_F1_target_estimated_ck = dic_list_F1_target.copy()

for i, idx in enumerate([i for i in range(len(dic_list_F1_target_estimated_ck)) if
(dic_list_F1_target_estimated_ck[i]['x'], dic_list_F1_target_estimated[i]['y']) not in
sample_coordinates]):
        dic_list_F1_target_estimated[idx]['aij'] = predicted_values[i]
        dic_list_F1_target_estimated[idx]['type'] = 'interpolated'

df_interpolated_ck = pd.DataFrame(dic_list_F1_target_estimated_ck)

print(dic_list_F1_target_estimated_ck[:5])

print(df_interpolated_ck.head())

#Choose at least two methods

#Deep Kriging algorithm
for i, sample_ratio in enumerate(sample_ratios):
        np.random.seed(0)
        sample_indices = np.random.choice(len(X), int(sample_ratio * len(X)),
replace=False)
        X_train = X[sample_indices]
        y_train = y[sample_indices]

        scaler = StandardScaler()
        X_train_scaled = scaler.fit_transform(X_train)

        model = Sequential([
                Dense(64, activation='relu', input_shape=(4,)),
                Dense(128, activation='relu'),
                Dense(64, activation='relu'),
                Dense(1)
        ])
        model.compile(optimizer=Adam(learning_rate=0.01), loss='mse')
```

Question 4

```
def find_corresponding_values(target_df, v_df):
        merged_df = target_df.merge(v_df[['x', 'y', 'aij']], on=['x', 'y'], how='left')
        merged_df.columns = ['aij_target','x','y','aij_variable']
        return merged_df

dic_df_F2_target_with_v1_values = find_corresponding_values(dic_df_F2_target,
dic_df_F2_v1)
dic_df_F2_target_with_v2_values = find_corresponding_values(dic_df_F2_target,
dic_df_F2_v2)
dic_df_F2_target_with_v3_values = find_corresponding_values(dic_df_F2_target,
dic_df_F2_v3)
```

```
dic_df_F2_target_with_v4_values = find_corresponding_values(dic_df_F2_target,
dic_df_F2_v4)


#Bivariate global Moran index

def create_spatial_weights(df):
    coordinates = [(d['x'], d['y']) for d in df]
    w = lp.weights.DistanceBand.from_array(coordinates, threshold=1.5)
    return w

w = create_spatial_weights(dic_list_F2_target)

def calculate_bivariate_moran(target, collaborative_variable, w):
    target_flat = target['aij']
    collaborative_variable_flat = collaborative_variable['aij']

    moran = Moran_BV(target_flat, collaborative_variable_flat, w)
    return moran.I, moran.p_sim

morans = []
name_list =
[dic_df_F2_v1_corresponded,dic_df_F2_v2_corresponded,dic_df_F2_v3_correspond
ed,dic_df_F2_v4_corresponded]
for var in name_list:
    moran_I, p_value = calculate_bivariate_moran(dic_df_F2_target, var, w)
morans.append((moran_I, p_value))

moran_I_list = []

for i, (moran_I, p_value) in enumerate(morans):
    print(f"F1 Collaborative Variable {i +1} - Bivariate Moran's I: {moran_I},
p-value: {p_value}")
    moran_I_list.append(moran_I)

best_index = np.argmax([moran_I for moran_I, _ in morans])
best_moran_I, best_p_value = morans[best_index]

from skgstat import Variogram, OrdinaryKriging

f2_target.columns = ['x', 'y'] + list(f2_target.columns[2:])

x = f2_target['x'].values
y = f2_target['y'].values
```

```
z = f2_target['Target Property'].values

coordinates = np.column_stack((x, y))

variogram = Variogram(coordinates, z)

grid_x, grid_y = np.mgrid[0:266, 0:266]
grid_points = np.vstack([grid_x.ravel(), grid_y.ravel()]).T

kriging = OrdinaryKriging(variogram)
kriged_values = kriging.transform(grid_points)

kriged_grid = kriged_values.reshape((266, 266))


#Deep Kriging algorithm
sample_size = 1000

def build_model(hp):
	model = Sequential()
	model.add(Dense(units=hp.Int('units', min_value=32, max_value=512, step=32),
	                    input_shape=(X_train.shape[1],),
	                    kernel_regularizer=tf.keras.regularizers.l2(hp.Float('l2_reg',
min_value=1e-5, max_value=1e-1, step=1e-2)),
	                    activation='relu'))
	model.add(Dropout(rate=hp.Float('dropout_1', min_value=0.0, max_value=0.5,
step=0.1)))
	model.add(BatchNormalization())

	model.add(Dense(units=hp.Int('units_2', min_value=32, max_value=512,
step=32),


kernel_regularizer=tf.keras.regularizers.l2(hp.Float('l2_reg_2', min_value=1e-5,
max_value=1e-1, step=1e-2)),
	                    activation='relu'))
	model.add(Dropout(rate=hp.Float('dropout_2', min_value=0.0, max_value=0.5,
step=0.1)))
	model.add(BatchNormalization())

	model.add(Dense(1, activation='linear'))
	model.compile(loss='mse', optimizer='adam', metrics=['mae', 'mse'])
	return model
```

```
tuner = RandomSearch(
    build_model,
    objective='val_mse',
    max_trials=5,
    executions_per_trial=1,
    directory=f'my_dir_{sample_size}',
    project_name=f'deep_kriging_tuning_{sample_size}'
)
```