UM-SJTU Joint Institute
PROBABILISTIC METHODS IN ENGINEERING
VE401

TERM PROJECT 2

PHOTOLITOGRAPHY OVERLAY FOR PATTERNING OF INTEGRATED CIRCUITRY

Group 18

Dong Yifan	518370910211
Gracia Stefani	517370990022
Jin Haoxiang	518370910215
Li Minhao	518370910148

Date: July 31 2020

Abstract

Multiple-linear regression plays a significant role in our life. For instance, the overlay error in photolitography can be describes using a multiple-linear regression model. In this project, we try to analyze various models using method of multiple-linear regression tool and tries to improve the base model. Then we try to use pytorch, an open source package for machine learning, to implement a neural network for fitting the data. The testing results show that the model is good at preventing overfit. However, the PRESS is larger than that of the linear regression model.

Key Words: Multiple linear regression, Pytorch, Neural network

Contents

I Introduction

Photolitography is a process in which a pattern is created on a wafer by light exposure through a protective mask layer. This method can be used to create integrated circuits (*dies*). One problem related to photolitography is that multi-layer imprinting of dies is always accompanied by some amount of shifting, i.e. the layers are not perfectly aligned.

We denote the nominal coordinates of each control point in a die as $(X+x, Y+y)$. We refer to (X, Y) for the position of the die and (x, y) for the nominal coordinate of each die relative to its position. The *overlay error* o_x, o_y expresses the deviation from the nominal coordinates. The actual position for each control point can then be expressed as:

$$x_{\text{actual}} = X + x + o_x(X, Y, x, y),$$
$$y_{\text{actual}} = Y + y + o_y(X, Y, x, y).$$

By modelling the overlay errors, one will be able to adjust the processing machines such that the errors can be corrected. For the next sections, we will analyze some models relating to the overlay errors and try to find an optimal model for them. For the following discussions, we have used the experimental data provided by Dragan Djurdjanovic, Professor at the Walker Department of Mechanical Engineering at the University of Texas at Austin.

II The Basic Model

The base model for this project is

$$o_x(X, Y, x, y) = \alpha_0 + \alpha_1 X + \alpha_2 Y + \alpha_3 XY + \alpha_4 X^2 + \alpha_5 Y^2 + \alpha_6 x + \alpha_7 y + \alpha_8 xy + \alpha_9 x^2 + \alpha_{10} y^2$$
$$o_y(X, Y, x, y) = \beta_0 + \beta_1 X + \beta_2 Y + \beta_3 XY + \beta_4 X^2 + \beta_5 Y^2 + \beta_6 x + \beta_7 y + \beta_8 xy + \beta_9 x^2 + \beta_{10} y^2$$

Using the data provided, we performed a nonlinear fit in Mathematica to obtain the coefficients of the base model given above. We obtained the following values for the coefficients of $o_x(X, Y, x, y)$ model:

	Estimate	Standard Error	P-value
α_0	-0.262423	0.121	0.0306269
α_1	0.000652114	0.00055	0.238926
α_2	0.00127669	0.00056	0.0243581
α_3	-0.000056137	8.54×10^{-6}	3.12098×10^{-10}
α_4	-0.0000462102	7.87×10^{-6}	1.4253×10^{-8}
α_5	-0.0000131142	7.71×10^{-6}	0.0903008
α_6	0.0665979	0.00644	$6.0945\ \times10^{-21}$
α_7	-0.0345959	0.00373	1.23034×10^{-17}
α_8	-0.000352896	0.000564	0.532278
α_9	0.0029393	0.00126	0.0209764
$\alpha_1 0$	0.00105152	0.000359	0.00369793

Table 1: Coefficients for the x overlay errors of the base model

Using this model with the coefficients in table III, we obtain the coefficient of determination $R^2 = 0.5776$. This means that 0.5776 of total the variation in o_x is explained by the model, which can be considered as not good. There is a hint that the model for o_x above is not a good model. Moreover, we have also calculated the PRESS statistic to be 109.135.

Similarly, we also obtained the following values for the coefficients of $o_y(X, Y, x, y)$ model:

	Estimate	Standard Error	P-value
β_0	0.706481	0.116706	5.44671×10^{-9}
β_1	0.00264355	0.000534222	1.41241×10^{-6}
β_2	0.00358417	0.000544968	2.99956×10^{-10}
β_3	3.04558×10^{-6}	8.26423×10^{-6}	0.712808
β_4	5.51767×10^{-6}	7.61008×10^{-6}	0.469132
β_5	-9.48762×10^{-6}	7.45802×10^{-6}	0.204561
β_6	0.0163108	0.00623137	0.00942216
β_7	0.0131395	0.00361407	0.000339551
β_8	0.000787553	0.000545719	0.150289
β_9	-0.00335037	0.00122337	0.00663376
β_{10}	0.00265074	0.000346898	5.18725×10^{-13}

Table 2: Coefficients for the y overlay errors of the base model

Using this model with the coefficients in table III, we obtain the coefficient of determination $R^2 = 0.3892$, which indicates that this model is not very good. We have also calculated the PRESS statistic to be 101.619. The two presses are all smaller than the orignal one.

III Simplification on the Model

The base model for the project is given as

$$o_x(X,Y,x,y) = \alpha_0 + \alpha_1 X + \alpha_2 Y + \alpha_3 XY + \alpha_4 X^2 + \alpha_5 Y^2 + \alpha_6 x + \alpha_7 y + \alpha_8 xy + \alpha_9 x^2 + \alpha_{10} y^2$$
$$o_y(X,Y,x,y) = \beta_0 + \beta_1 X + \beta_2 Y + \beta_3 XY + \beta_4 X^2 + \beta_5 Y^2 + \beta_6 x + \beta_7 y + \beta_8 xy + \beta_9 x^2 + \beta_{10} y^2$$

And we would like to simplify the model and eliminate some unnecessary terms in this model.

The method we applied is the *Backward Elimination Procedure* mentioned in the slides of VE401. We first calculated out the Rsquared values of the models with one less term, and then compared them and chose the model with the biggest Rsquared value. Then we performed the *T-Test for Model Sufficiency*: We assumed the null hypothesis $H_0 : \beta_j = 0$ for $j = 0, 1, \cdots, p$, and rejected it at significance level α if

$$T_{n-p-1} = \frac{b_j}{S\sqrt{\xi_{jj}}}$$

satisfies $|T_{n-p-1}| > t_{\alpha/2,n-p-1}$

The significance level we chose is 5%. Using the data provided, we did the *Backward Elimination Procedure* with the help of Mathematica and found out that for the model

$$o_x(X,Y,x,y) = \alpha_0 + \alpha_1 X + \alpha_2 Y + \alpha_3 XY + \alpha_4 X^2 + \alpha_5 Y^2 + \alpha_6 x + \alpha_7 y + \alpha_8 xy + \alpha_9 x^2 + \alpha_{10} y^2$$

the terms that we can eliminate are xy, X and Y^2. We obtained the following values for the coefficients of the simplified $o_x(X,Y,x,y)$ model:

	Estimate	Standard Error	P-value
α_0	-0.38551	0.0975524	0.000101822
α_2	0.00123127	0.000564233	0.030057
α_3	-0.000056411	8.55344×10^{-6}	2.64782×10^{-10}
α_4	-0.000056411	7.38931×10^{-6}	6.51577×10^{-8}
α_6	0.0645879	0.00627652	7.7761×10^{-21}
α_7	-0.0345166	0.00373851	1.37758×10^{-17}
α_9	0.0032819	0.00125462	0.00946029
α_{10}	0.00109171	0.000359052	0.00262116

Table 3: Coefficients for the x overlay errors of the simplified model

The coefficient of determination of the simplified model is $R^2 = 0.5694$, which does not vary a lot from the determination coefficient $R^2 = 0.5776$ of the original model. The new press is 108.116.
And for the model

$$o_y(X, Y, x, y) = \beta_0 + \beta_1 X + \beta_2 Y + \beta_3 XY + \beta_4 X^2 + \beta_5 Y^2 + \beta_6 x + \beta_7 y + \beta_8 xy + \beta_9 x^2 + \beta_{10} y^2$$

the terms that we can eliminate are XY, X^2, xy and Y^2. We obtained the following values for the coefficients of the simplified $o_y(X, Y, x, y)$ model:

	Estimate	Standard Error	P-value
β_0	0.670236	0.0829282	2.99662×10^{-14}
β_1	0.00265161	0.000535508	1.37815×10^{-6}
β_2	0.00352008	0.000545031	5.71269×10^{-10}
β_6	0.0164336	0.00624096	0.00900116
β_7	0.0137084	0.00361212	0.000186386
β_9	-0.00315451	0.00121126	0.00977312
β_{10}	0.00268	0.000346837	2.89295×10^{-13}

Table 4: Coefficients for the y overlay errors of the simplified model

The coefficient of determination of the simplified model is $R^2 = 0.3758$, which does not vary a lot from the determination coefficient $R^2 = 0.3892$ of the original model. The new press is 100.124.

IV Improved Model

In the previous second, we have eliminated a few terms to make our model simplified but not changed too much. In this section, we have tried a lot of times to add a few terms and improve the model in a better way.

For o_x, our new model contains 14 terms

$$\begin{aligned} o_y(x, X, y, Y) = {} & \alpha_0 + \alpha_2 Y + \alpha_3 XY + \alpha_4 X^2 + \alpha_5 x + \alpha_6 y + \alpha_7 x^2 + \alpha_8 y^2 \\ & + \alpha_9 e^{|X|/10} + \alpha_{10} e^{|Y|/10} + \alpha_{11} e^{-x} + \alpha_{12} e^{-y} + \alpha_{13} e^{x} + \alpha_{14} e^{y} \end{aligned}$$

We obtain that the $R^2 = 0.6202$ and press is 101.468, which is much smaller than the previous model. The following chart for estimate of each coefficient:

α_0	-0.4014
α_2	0.001533
α_3	-0.00006086
α_4	-0.00005497
α_5	0.03224
α_6	-0.05105
α_7	0.006755
α_8	0.007645
α_9	2.523×10^{-7}
α_{10}	-6.519×10^{-7}
α_{11}	-0.0001427
α_{12}	-2.22×10^{-6}
α_{13}	-0.00001529
α_{14}	-3.201×10^{-7}

Table 5: Coefficients for improved model in o_x

For o_y, our new model contains 15 terms

$$o_y(x, X, y, Y) = \beta_0 + \beta_1 X + \beta_2 Y + \beta_3 x + \beta_4 y + \beta_5 x^2 + \beta_6 y^2 + \beta_7 e^{|x|} + \beta_8 e^{|y|} + \beta_9 e^{|X|/10} + \beta_{10} e^{|Y|/10} + \beta_{11} e^{X/10} + \beta_{12} e^{Y/10} + \beta_{13} \frac{x}{y} + \beta_{14} \frac{y}{x}$$

We obtain that the $R^2 = 0.5864$ and the press is 71.734, which is far less than the value of the previous model. The following chart for estimate of each coefficient:

β_0	0.3112
β_1	0.00301
β_2	0.00656
β_3	0.0422
β_4	0.0217
β_5	0.0144
β_6	0.00741
β_7	-0.000173
β_8	-1.035×10^{-7}
β_9	1.137×10^{-7}
β_{10}	2.796×10^{-6}
β_{11}	-3.201×10^{-7}
β_{12}	-4.083×10^{-6}
β_{13}	0.1704
β_{14}	0.005554

Table 6: Coefficients for improved model in o_y

V Neural Network Model

Previously, we use multilinear regression model to fit the data by minimizing error sum of squares. In this section, a neural network is established to find an optimal model.

A Forward Propagation

Firstly, we are going to show how neural network gives us a prediction. Multilinear regression model's estimators are represented by an n dimension vector while neural networks' estimators are represented by several matrixes with real coefficients. For instance, the following figure shows a neural network.

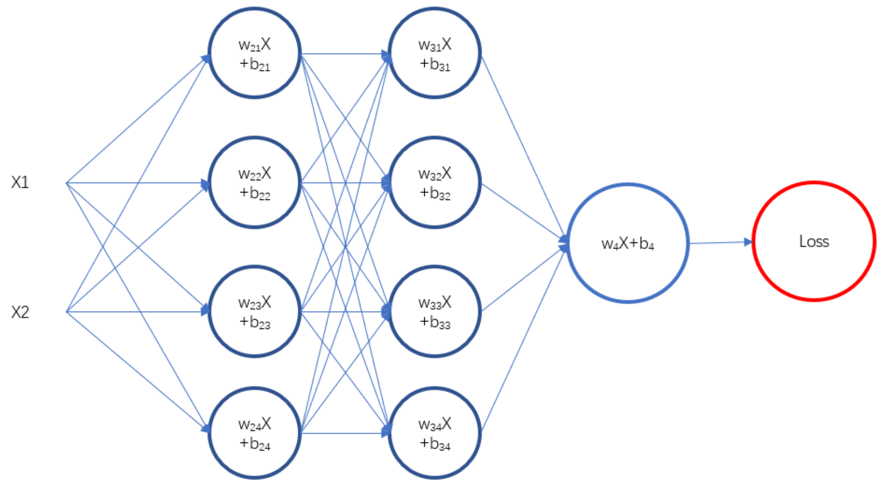


Figure 1: Neural network representation

Each circle, called cell, represents an intermediate real number. It is calculated by multiplying weight vector w_{ij} and input vector X, then adds a real number b_{ij}. The weight vector is a row vector and input vector is a column vector. Each column of cells form a layer and the input vector X is simply result of previous layer where X_i is the i_{th} result. From the view of layer, each layer's weight W is an $n \times m$ matrix, where $W_{i.}$ represents the i^{th} row vector. n represents the number of cells and m represents the dimension of input vector or previous layer's number of cells. Also, each layer's bias B is an n dimension column vector. In this example, the variable X_1 and X_2 form the 2 dimension input vector for the second layer. After passing through the second layer, it outputs a 4 dimension column vector to next layer and finally the red circle turns the 4 dimension column vector to a real number. Use W_{ij}^k, B_i^k,Y^k to represent k^{th} layer's weight matrix, bias vector, and output vector. Here ij is the entry of each element. The first layer is input layer and hence it doesn't have weight or bias. The process of forward propagation(from left to right in this figure) can be mathematically represented by:

$$\begin{pmatrix} Y_1^1 \\ Y_2^1 \end{pmatrix} = \begin{pmatrix} X_1 \\ X_2 \end{pmatrix}$$

$$\begin{pmatrix} Y_1^2 \\ Y_2^2 \\ Y_3^2 \\ Y_4^2 \end{pmatrix} = \begin{pmatrix} W_{11}^2 & W_{12}^2 \\ W_{21}^2 & W_{22}^2 \\ W_{31}^2 & W_{32}^2 \\ W_{41}^2 & W_{42}^2 \end{pmatrix} \times \begin{pmatrix} Y_1^1 \\ Y_2^1 \end{pmatrix} + \begin{pmatrix} B_1^1 \\ B_2^1 \\ B_3^1 \\ B_4^1 \end{pmatrix}$$

$$\begin{pmatrix} Y_1^3 \\ Y_2^3 \\ Y_3^3 \\ Y_4^3 \end{pmatrix} = \begin{pmatrix} W_{11}^3 & W_{12}^3 & W_{13}^3 & W_{14}^3 \\ W_{21}^3 & W_{22}^3 & W_{23}^3 & W_{24}^3 \\ W_{31}^3 & W_{32}^3 & W_{33}^3 & W_{34}^3 \\ W_{41}^3 & W_{42}^3 & W_{43}^3 & W_{44}^3 \end{pmatrix} \times \begin{pmatrix} Y_1^2 \\ Y_2^2 \\ Y_3^2 \\ Y_4^2 \end{pmatrix} + \begin{pmatrix} B_1^3 \\ B_2^3 \\ B_3^3 \\ B_4^3 \end{pmatrix}$$

$$Y^4 = \begin{pmatrix} W_{11}^4 & W_{12}^4 & W_{13}^4 & W_{14}^4 \end{pmatrix} \times \begin{pmatrix} Y_1^3 \\ Y_2^3 \\ Y_3^3 \\ Y_4^3 \end{pmatrix} + B^4$$

B Loss Function

After knowing how neural network gives us a prediction, we also need to determine how far is the output away from real measurements. We use pytorch, an open source machine learning tool, to implement the algorithm and it has provided us with several ways of calculating loss. In this project, we get the loss by calculating the mean of error sum of squares. The function is called "MSELoss" in pytorch.

C Back Propagation

The linear regression model gets the parameters with one "propagation". However, the neural network improves its estimation by trying to change the parameters in order to decrease the loss for many times. The process of changing parameters is called back propagation because it can only be executed after getting the prediction and loss. Pytorch offers many API of optimizer which can do back propagation for us. Although each optimizer employs different strategies, the most common way is to calculate the partial derivative of loss with respect to each layer's weight and bias and then add a product of a constant α, called learning rate, and the partial derivative to parameters. Use the previous example. Denote the loss as L and loss function as $f(x)$, which maps Y^4 into a real number. The forward propagation can be simplified as:

$$L = f(Y^4), \qquad Y^i = W^i \times Y^{i-1} + B^i$$

The derivative of L with respect to Y^4 can be viewed as a fixed function of Y^4 because the loss function is determined as soon as the model is established. In this example, Y^4 is a real number while the last layer can also have more than one cell. Then x becomes an n dimension vector where n is the number of last layer's cells. Assume that the loss function is $L = Y^4$ for simplicity. By applying the chain rule, we can get the partial derivative of L with respect to each layer's weight and bias. Then by subtracting the product of learning rate and partial derivative, we can update the parameters.

$$W^4 := W^4 - \alpha \times {Y^3}^{\mathrm{T}}, \qquad B^4 := B^4 - \alpha$$

$$W^3 := W^3 - \alpha \times {W^4}^{\mathrm{T}} \times {Y^2}^{\mathrm{T}}, \qquad B^3 := B^3 - \alpha \times {W^4}^{\mathrm{T}}$$

$$W^2 := W^2 - \alpha \times {W^3}^{\mathrm{T}} \times {W^4}^{\mathrm{T}} \times {Y^1}^{\mathrm{T}}, \qquad B^2 := B^2 - \alpha \times \begin{pmatrix} W^4_{11} \times (\sum_{n=1}^4 W^3_{n1}) \\ W^4_{12} \times (\sum_{n=1}^4 W^3_{n2}) \\ W^4_{13} \times (\sum_{n=1}^4 W^3_{n3}) \\ W^4_{14} \times (\sum_{n=1}^4 W^3_{n4}) \end{pmatrix}$$

The intuition of subtracting such a product is that the partial derivatives represent the slope at that point. If a parameter's partial derivative is positive, then decrease it can make the loss smaller and vice versa.

D Finding The Right Model

There is much flexibility on selecting the neural network model. We can change the structure of neural network, learning rate, optimizer and loss function. We don't focus on the choice of optimizers because their main strategies of doing back propagation are similar and most of their differences lie in accelerating the training process, which is not a concern in this project. Also, as mentioned before, we only use the mean of error sum of squares as the loss. In this way, both neural network and linear regression model aim at reducing the same loss and their results are comparable.

Firstly, a neural network with 4 layers is used. The first layer is input layer and it consists of four variables X, x, Y, y. The second layer has 10 cells and the third layer has 10 cells and the fourth layer has one output cell. After the model is trained for 1000 times, learning rate at 0.001, it shows a convergence of error sum of squares.

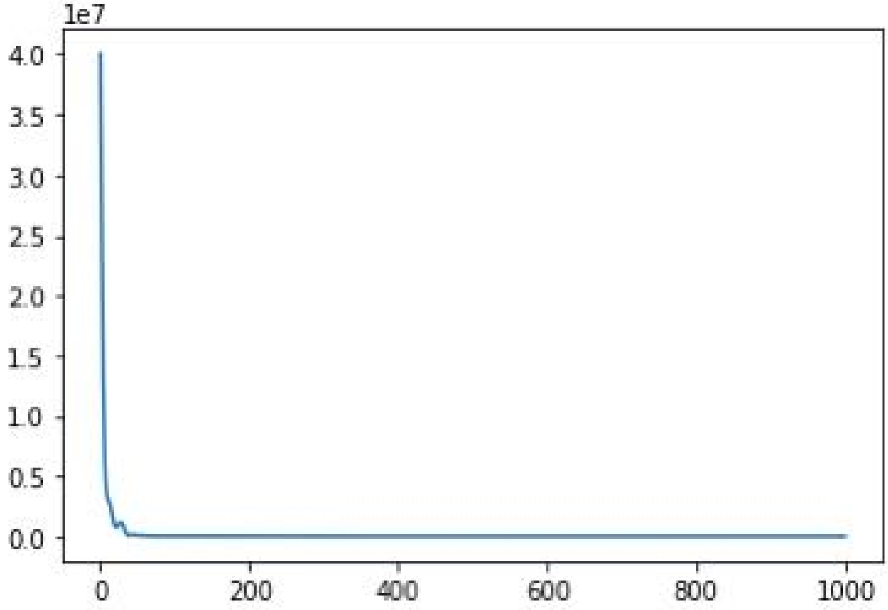
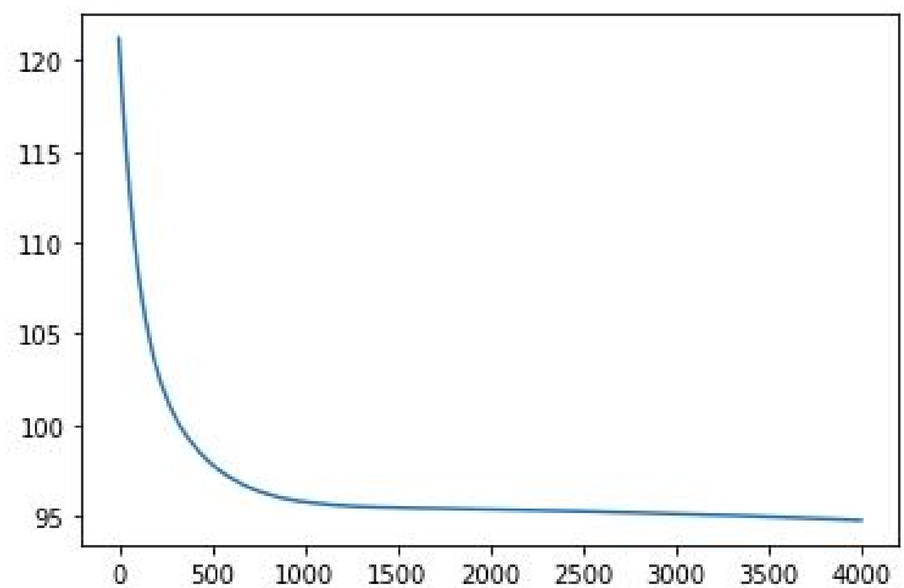


Figure 2: SSE vs. training times

The Y-axis represents error sum of square and the X-axis represents the trails. We can see that the model converges successfully. Based on this training, we continue to train for another 4000 times in order to have a closer look at convergence.



epoch.jpg

Figure 3: SSE vs. training times

We can see that after 2000 times of training, the loss almost reaches minimum. Hence it is reasonable to choose 0.001 as the learning rate and train the model for 2000 times. However, since the weights and biases are randomly initialized between -1 and 1, there is possibility that the they are coincidentally close to proper parameters. We may change training times further based on prediction results. Using this model to predict

ox we get PRESS of 143.0757.

press.jpg

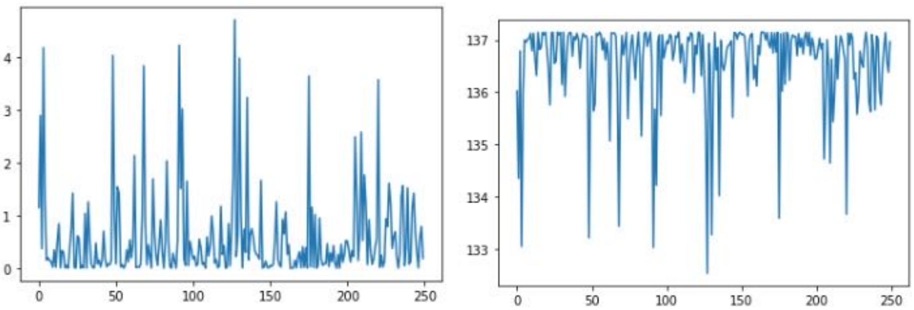


Figure 4: Prediction result

The first figure shows the component of PRESS, namely the square of difference between the prediction of x^{th} set of data and its measurement value. The x^{th} training excludes the x^{th} set of data. The second figure shows the SSE of each trial, using 249 sets of data. We can find evidence that the model is lack of fit because according to previous result, we expect the SSE to be roughly 100 while the experimental SSE is around 136. We improve the model's ability to fit data by adding more layers, variables and training times. The 10 variables of base model are used and three extra layers with 20 cells, 40 cells and 20 cells are added after the second layer. Also, we train 3000 times per trial to ensure the convergence. In this way, there are more parameters for the model to update. Finally, we find that the PRESS equals 117.5327.

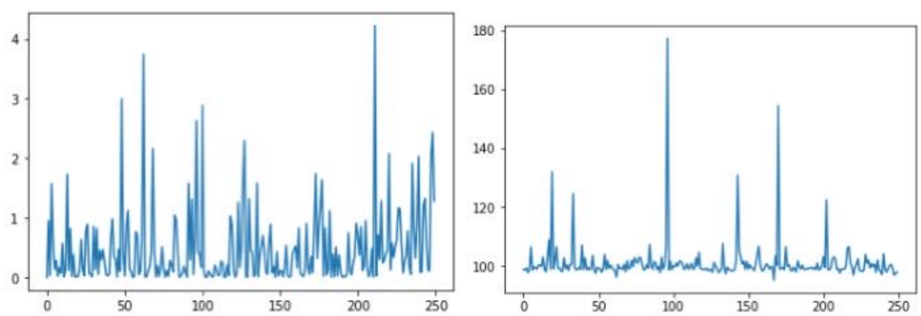


Figure 5: Prediction result for o_x with improved model

Similarly, from first figure we can see that the components of PRESS are mostly smaller than one. The second figure shows that the improved model has a better ability of fitting data because most SSE are smaller than 100.

Using the same model for fitting o_y we find that the PRESS is 111.2552.

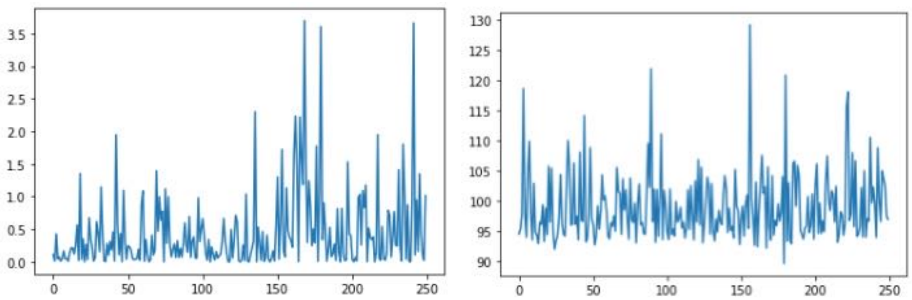


Figure 6: Prediction result for o_y using the improved model

To verify whether the improved model is necessary for fitting oy, we use the first simple version to calculate PRESS, which turns out to be 119.6242.

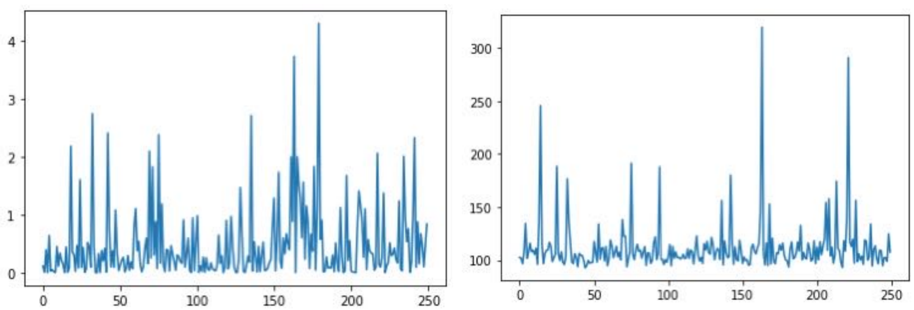


Figure 7: Prediction result for o_y using simple model

From the numerical comparison between PRESS, we can't find evidence that a more sophisticated model leads to a better prediction for oy because their PRESS are close to each other. However, since PRESS of the improved model is smaller, we consider it to be better than simple model.

E Test For Overfit

To test whether our model is overfit, we randomly select 200 data for training. After getting the trained model, we insert other 50 data to calculate the testing SSE. If the model is seriously overfitting, then the SSE of 50 data should be very large and causes the total SSE to increase a lot. The testing SSE of ox is 21.2928 with training SSE 78.7544. The testing SSE of o_y is 25.6653 with training SSE 76.6826. Since the both total SSE are only around 100, similar to the SSE which we previously get with full training data, there is no evidence that the model is overfitting.

VI Conclusion

The purpose of this paper was to give us an application of how to apply the knowledge of multi-linear regression to the model in daily life.

First, we use the basic model to derive the formula for each o_x and o_y. The RSquared value shows that they are not very appropriate models. We then tried to simplify the base model using the backward elimination procedure. We did this by investigating the RSquared value of the models with one term less than the currently considered model and choose the one with the biggest RSquared value to proceed with the investigation. We finally found that we can simplify the model of o_x in the order of xy, X, Y^2 with significance level 5%, and the model of o_y in the order of XY, X^2, xy, Y^2 with significance level 5%. The

RSquared of the simplified model is $R^2 = 0.569381$ for o_x and $R2 = 0.375834$ for o_y, which are both quite close to that of the original models. After the simplification, we tried to add a few approximate terms to each model, mainly focusing on the exponential term such as $\exp(|X|/10)$ and $\exp(-x)$.

Lastly, we used neural networks to predict o_x and o_y. After improving the model based on the prediction results, we decided to use a neural network with 6 layers. They have 10, 20, 40, 20, 10, 1 cell respectively and all of them have bias. The PRESS for o_x is 117.5327 and the PRESS for o_y is 119.6242. There is a large difference in the PRESS of the of o_x of the simplified model and improved model . However, there isn't much evidence that the improved model works better for predicting o_y. Eventually, the data are divided into training data and testing data to examine whether the model is overfitting. From the result we can find evidence that the model is not overfitting for both o_x and o_y because the total SSE are close to the SSE obtained from full data.

References

[1] Hohberger, H., Ve401 Probabilistic Methods in Engineering, Term Project 2 Guidelines, Summer 2020. UM-SJTU Joint Institute, Shanghai Jiao Tong University, Shanghai. Author: `http://umji.sjtu.edu.cn/faculty/horst-hohberger`