Optimization Project - III

Nonlinear Programming

1	0	0	1	0
0	1	1	0	0
0	0	1	1	0
0	0	0	0	1

By Jesper Li, Surya Prasad Reddy P, Yashpreet Kaur

Background

In predictive analytics, variable selection is always a common problem in regression. Too many variables will lead to an overly complex model with high variance, and less than needed variables will lead to too simple models with high bias. The optimal number of variables will give us optimal mean squared error, resulting in a better model. In the past, direct variable selection using optimization was not efficient since it was computationally difficult. Instead, lots of indirect variable selection methods, such as Lasso and ridge, were commonly used. Recently, with the huge development of optimization software, direct variable selection has become operable. In this project, our goal is to directly select variables for regression by solving mixed integer quadratic programs and compare the result with the one from Lasso.

Key Insights

Before we jump into the coding part, it is important to talk about some terminologies. For direct variables selection, we actually choose the features that give us the least mean square error in the training dataset. The formula is as following:

$$\min_{\beta} \sum_{i=1}^{n} (\beta_0 + \beta_1 x_{i1} + \cdots + \beta_m x_{im} - y_i)^2$$

Our purpose is to find which and what beta we use so that we can minimize the MSE. In order to get this work, we need to use the optimization software to help us.

After directly selecting the variables, we will use the library lassocv to do the indirect variables selection and compare the results. The Lasso version of the loss function will be the following:

$$\min_{\beta} \sum_{i=1}^{n} (\beta_0 + \beta_1 x_{i1} + \cdots + \beta_m x_{im} - y_i)^2 + \lambda \sum_{j=1}^{m} |\beta_j|$$

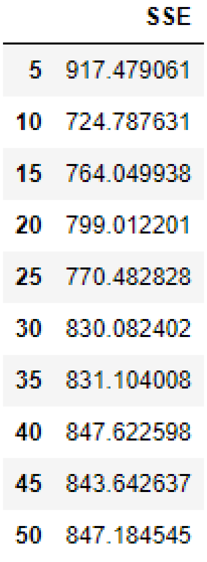
By adding a penalty part, the model will select variables during training the data.

Direct Variable Selection

For the objective, we cannot put the loss function directly into it; we need to use some linear algebra tricks to transform it to a matrix. After transformation, it will be in this form:

$$\min_{\beta, z} \beta^T (X^T X)\, \beta + (-2\, y^T X)\, \beta$$

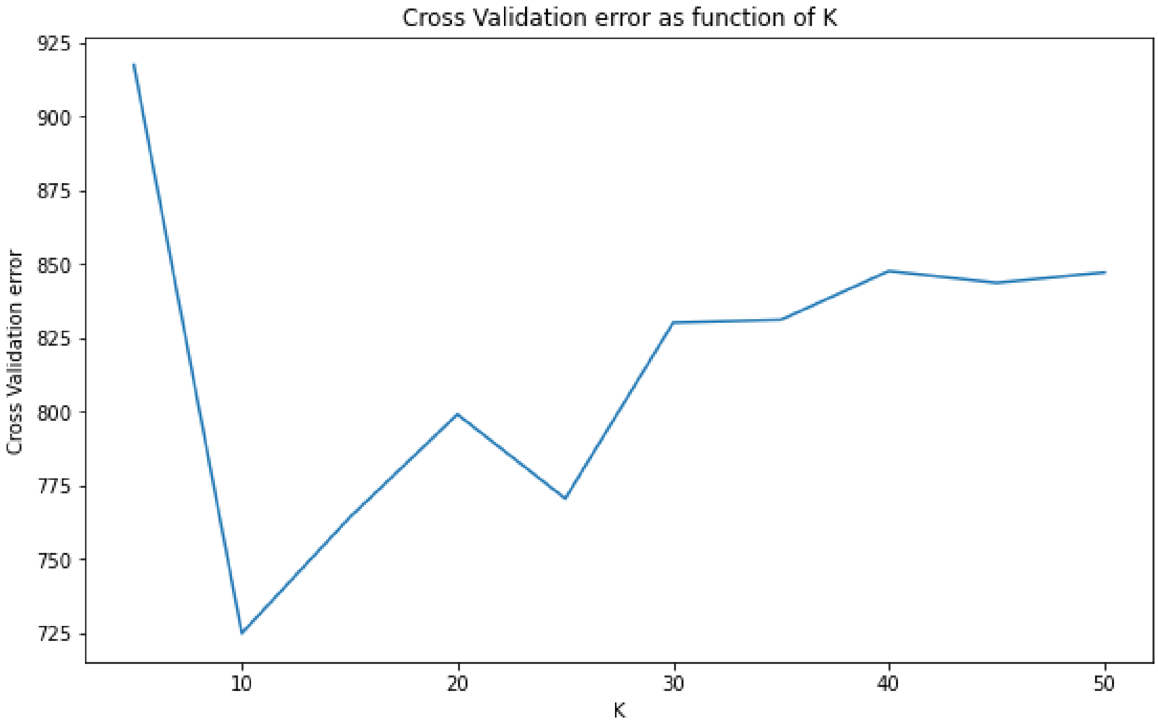
The objective will be constructed in two parts: the quadratic part and the linear part. In here, X will be our data input matrix, beta will be our decision variables, and y will be data output. In addition, we also introduce the z variables which will control the number of betas we select; the z variables will be binary variables, and it equals 1 if the corresponding beta is being selected, and 0 otherwise. Since our decision variables contain 50 new variables z (one for each beta, and we don’t need to control the beta 0 since it is the constant and will always be exist), we need to modify the input matrix to be 101 * 101 where the top left corner is equal to the quadratic part, and the linear term to be 101 * 1 contains 50 zero in the end. There are two types of constraints we need to consider: 100 constraints for beta, and a constraint on z. We want our beta equals to 0 when z equal to 0, and beta can be any number within the boundary of M when z equal to 1. Here, we use the big M technique, and the big M we choose is 20. The last constraint is that the number of z has to be less or equal to the number we choose.



	SSE
5	917.479061
10	724.787631
15	764.049938
20	799.012201
25	770.482828
30	830.082402
35	831.104008
40	847.622598
45	843.642637
50	847.184545

To avoid our choice of the optimal beta number by chance, we also did a 10-fold cross validation. We randomly split our training dataset into 10 groups and use these 10 groups of data to run the optimization process separately for each k values we choose. And then, we sum each validation’s set’s sum of squared errors. After loop through all the k values from 5 to 50 with an interval of 5, we compare the sum of sum squared errors, and choose the k values with the least sum of sum squared errors. The results are the following:

We can see that when k equals 10, we have the smallest SSE with 724.79. It implies that our cross validation suggests that we should choose 10 betas for our data, so that it can give us the best result.



Finally, we use 10 as the parameters for our MIQP on the test data, and the SSE error is 116.827.

Indirect Variable Selection -- Lasso

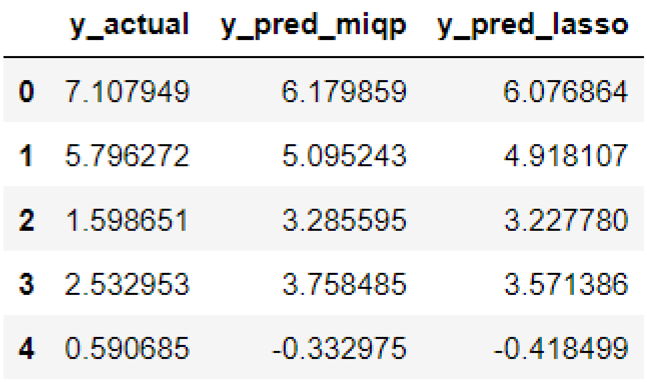
The objective function for the indirect variable selection using lasso regression is

$$\min_{\beta} \sum_{i=1}^{n} (\beta_0 + \beta_1 x_{i1} + \cdots + \beta_m x_{im} - y_i)^2 + \lambda \sum_{j=1}^{m} |\beta_j|,$$

We have run a 10-fold cross fold validation on the values of the alpha using scikit learn. Based on this, the best value of alpha, the regularization constant is 0.76. With this parameter, the number of non-zero beta coefficients is found to be 17 among a total of 51 beta coefficients and the sum of squared errors for the test data is 117.48.

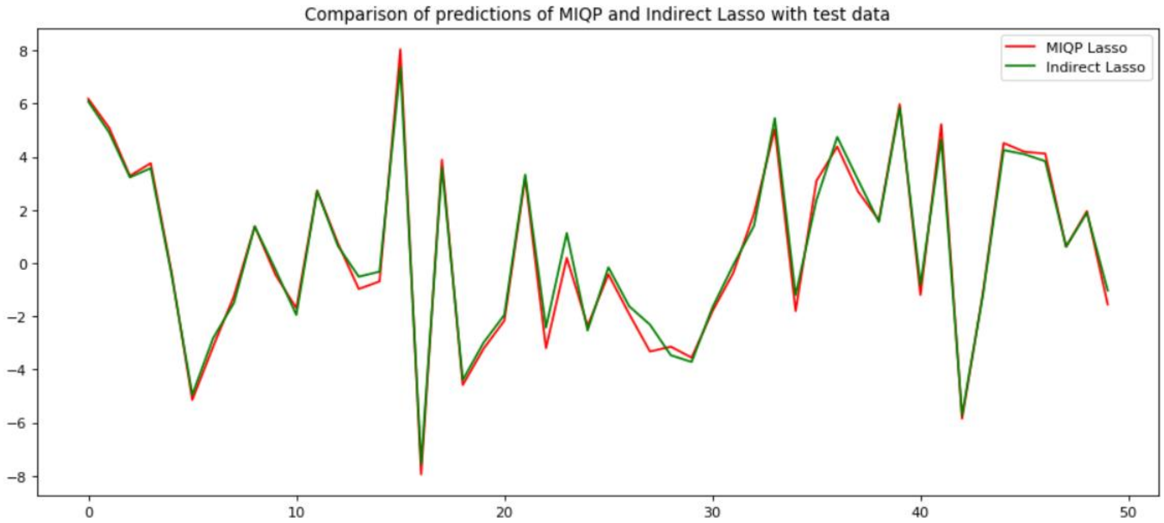
Recommendations

Let’s compare the performance of both the direct and indirect selection models of variable selection for Lasso regression.



	y_actual	y_pred_miqp	y_pred_lasso
0	7.107949	6.179859	6.076864
1	5.796272	5.095243	4.918107
2	1.598651	3.285595	3.227780
3	2.532953	3.758485	3.571386
4	0.590685	-0.332975	-0.418499

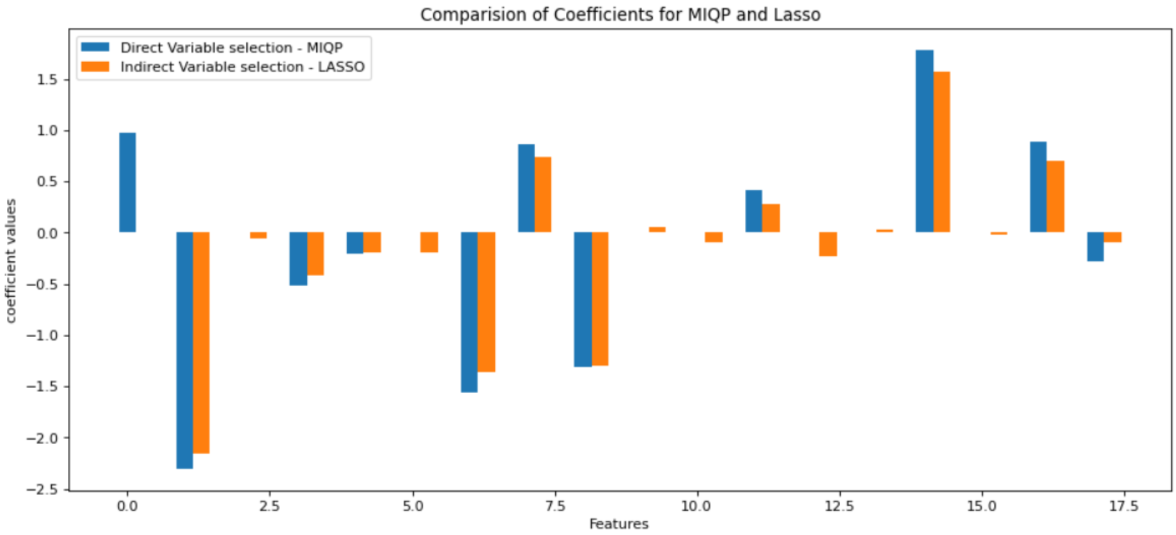
The plot of the predicted values of both the models is shown below



The best value of alpha is found to be 0.76 in indirect variable selection. Based on this the number of non zero coefficients is 17. However the value of K, i.e. the number of non zero coefficients for direct variable selection is found to be 10.

	Direct Variable Selection: MIQP	Indirect Variable Selection: Lasso
No of non zero coefficients	10	17
Testing data: sum of squares error	116.83	117.48

The plot of the beta coefficients of both the models is shown below.



Based on our analysis, we observed that the computation time for the direct variable selection using MIQP is much higher than that of indirect variable regression. Further, the computation time for direct variable selection will increase with increase in the number of the features.

There is no doubt that direct variable selection takes more time than indirect variable selection, but, as we saw above, there is actually a difference between the two methods SSE. Therefore, we suggest that we should evaluate costs and profits before deciding which method to use for a business-related project; if the incremental profits using the direct method exceed the costs, we should use the direct method and vice versa. For example, for a $1 billion project, the direct method may take a few days to get results, but a small difference in accuracy could result in a multi-million-dollar benefit. Therefore, for relatively small projects, we recommend that companies should continue to use Lasso regression for indirect variable selection because clients are already familiar with it and there are many built-in libraries available for optimization and scalability and use direct variable selection for larger projects when the additional cost is relatively small compared to the increase in profit.