Project 3 Report

Haoliang (Jack) Hu, Mengjia Sun, Michael Balis
McCombs School of Business, University of Texas at Austin
BAX358: Optimization Methods Decision Making
Dr. Daniel Mitchell
April 16, 2023

I. Introduction

Variable selection is an integral part of predictive analytics as it improves the predictive power of models. There are a few different strategies used for the variable selection process, and for this project, we will compare the direct and indirect variable selection methods using LASSO and MIQP, respectively. For both methods, we will use the same two datasets, one for training and one for testing. Our target variable will be the y-values, and our features to select for the model will come from the 50 X variables.

II. Indirect Variable Selection - LASSO

The Least Absolute Shrinkage and Selection Operator, or LASSO, is a regression technique used for feature selection by "shrinking" data towards a central point as the mean. This shrinkage process prunes variables from the model, which prevents overfitting and minimizes the total loss.

When using the LASSO method, we imported the scikit-learn package to perform a 10-fold cross-validation on the training data set in order to estimate the performance of the model. We found the best value of λ to be around 0.0764, which is a tuning parameter that determines the amount of shrinkage. We then used that value of λ to fit a LASSO model on the entire training dataset to find the model's predicted y-values for the testing dataset. With these predictions, we compared them to the actual y-values from the testing dataset and evaluated the model using the mean squared error (MSE), which turned out to be around 2.35.

III. Direct Variable Selection - MIQP

Objective function

To perform direct variable selection, we first have to find the quadratic term (Q matrix) and linear term (c vector) that define our objective function. Our MIQP model's objective is to minimize the sum of squared error (SSE). By minimizing SSE, our model aims to find the best possible fit for the data points and reduce the amount of error between the predicted values and the actual values.

The objective function is:

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

Which after some linear algebra can be written as:

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

This means the Q matrix for our model is $X^T X$ and the c vector is -2$y^T X$. With the Q matrix and c vector, we are able to formulate our objective function easily by filling in the first (Q matrix) and second (c vector) arguments of the SetMObjective function.

Constraints

Other than the intercept and the betas, the output of our objective function also includes binary z variables that indicate whether a corresponding beta will be picked or not. To implement this concept, we use big M constraints to define upper and lower bounds for selected betas. Betas that are not selected will remain 0 because the corresponding z variables are zeros. We decided to define M as 80, which is bigger than all the coefficients we found using LASSO (all the coefficients found using the LASSO method turned out to be single digits). Our goal is to find the optimal k (the total number of non-zero z variables) that results in the lowest sum of squared errors.

The constraints can therefore be defined as:

$$s.t. -Mz_j \le \beta_j \le Mz_j \quad for\, j = 1, 2, 3, \dots, m$$

$$\sum_{j=1}^{m} z_j \le k$$

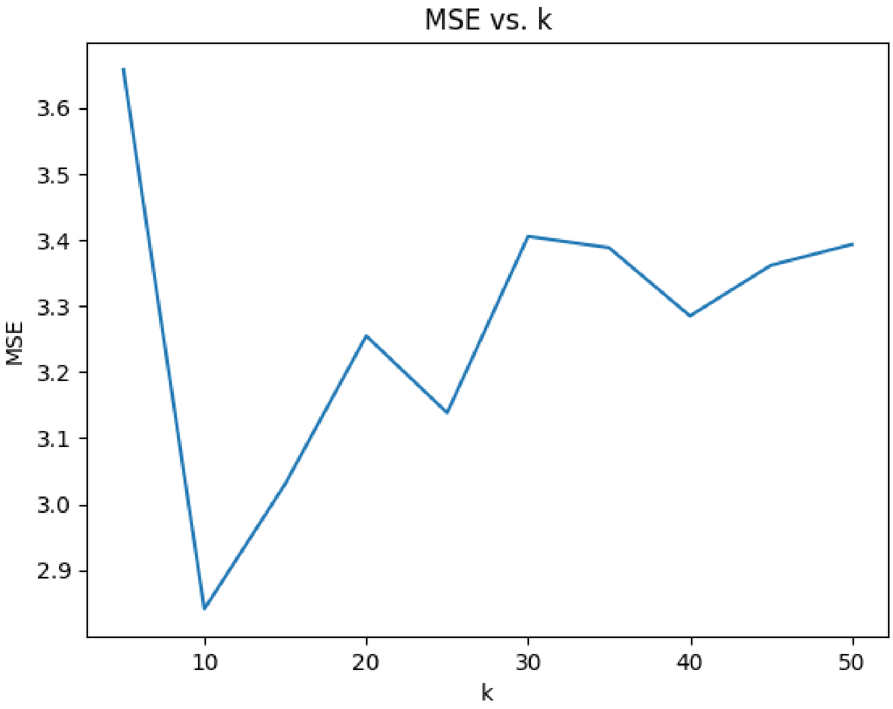
$$z_j \;\; are\ binary.$$

Finding the best k

After defining the objective function and the constraints, we are ready to use our MIQP model to test out different k values. We first created a list of k values (5,10,15,20,25,30,35,40,45,50) and then used a for loop to perform cross-validation each time we tested out a k value. For cross-validation, we first shuffled the data and split it into 10 folds. The model is then trained on 10-1 of the folds and validated on the remaining fold. This process is repeated 10 times, with each of the 10 folds used once as the validation set. The resulting MSE for each of the models trained on the 10-1 folds is then averaged to obtain an overall estimate of the model's performance. We decided to use MSE as the performance metric instead of SSE because, in general, MSE is preferred over SSE when comparing the performance of different models or evaluating the performance of a single model over multiple datasets. This is because MSE is normalized by the number of observations in the dataset, making it easier to compare models or evaluate performance across different datasets of different sizes.

After performing cross-validation on each of the k values, we found that the optimal k is 10, with an MSE of around 2.86 on the training set.

The MSE of each of the k values is shown below:

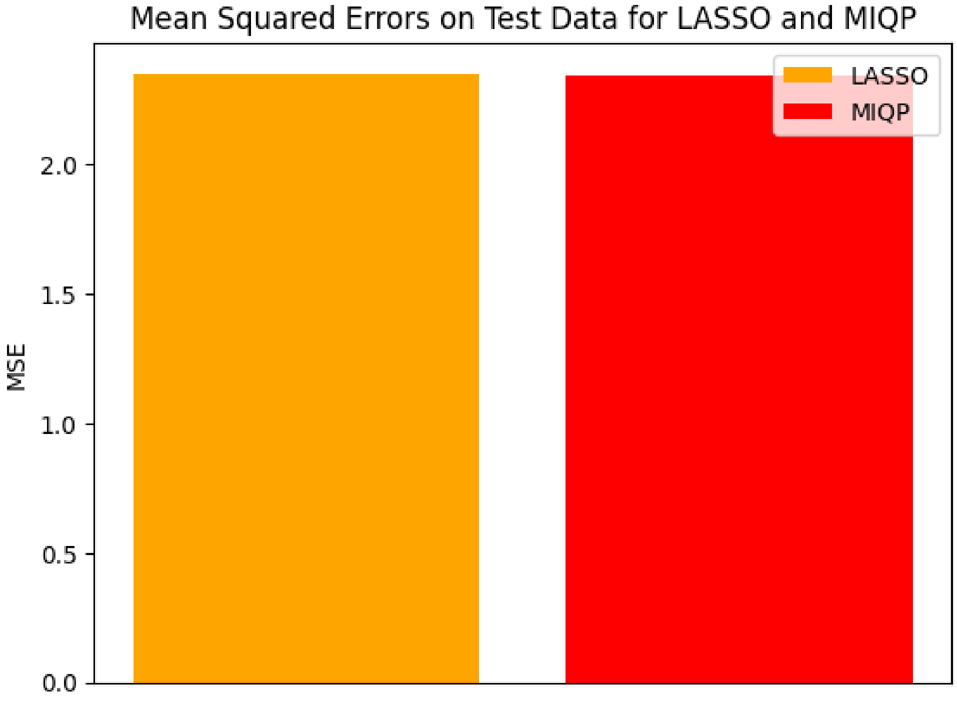


The graph above reaffirms our result that k = 10 leads to the lowest MSE.

Performance on out-of-sample data

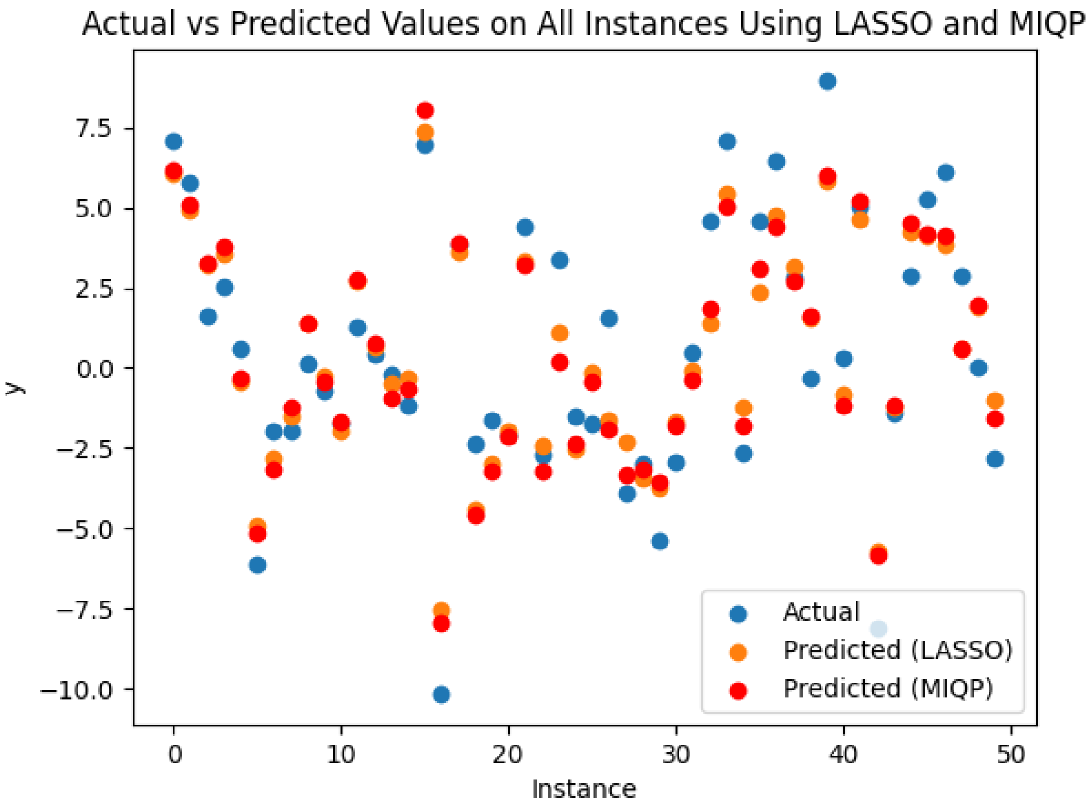
With the optimal k, k = 10, we fit our model on the entire training set and found the MSE on the out-of-sample (test set) data is around 2.34.

IV. Comparing the two methods



Although the computational time of direct variable selection has decreased with the advent of better solvers, the time and computational power required to perform MIQP means that this method still might not achieve a great ROI, depending on the usage of the model's predictions. If given more computational power and more time to run, MIQP will outperform LASSO by a wider margin.

Lasso regression tends to work better when the number of features is smaller than the number of samples because it can eliminate features from the model by setting their coefficients to zero. Additionally, it runs much faster than MIQP and yielded an extremely similar MSE (2.35 vs. 2.34), which was only about 0.5% worse than the MSE from the MIQP. Furthermore, the LASSO method is easier to code and simpler to explain to the average person, which may lead to better results as people are inherently distrusting of 'black box' models that they do not understand. Not only do people want to know the answer, they want to know how and why the answer came to be, so it can be argued that it is better to sacrifice 0.5% MSE in exchange for an easier and faster process.



The above visualization shows the predicted values, ŷ, for both the LASSO and MIQP models on all instances of the test data. While the MSE for the MIQP was marginally better, it is clear from the visualization that the models predicted values closer to each other's predictions than they did to the actual results.