Project 1: Boston Median Home Value Linear Model Emma Taylor

11571537

Introduction:

The task of this project is to find the best model with which to predict the variable “medv” (meaning median home value for owner-occupied homes in thousands) using the data set ‘Boston’, which is a data set that contains information on homes and neighborhoods in the suburbs of Boston. The most basic approach to this task would be to create a model using OLS (ordinary least squares) linear regression that developed regression coefficients and therefore its predictions by minimizing the RSS (residual sum of squares). This OLS model has the benefit of being unbiased because the RSS is based off random error terms, but the lack of bias means that it can be highly variable. Modern statistics aims to create models with low bias, but also with low variance and accurate predictions. In order to remedy this problem and find an optimal model according to these goals, we’ll use best subset selection and shrinkage methods that produce a better model either by reducing the number of predictors in the model, reducing the variance in the regression coefficients themselves, or by both.

As described in Chapter 6 of *An Introduction to Statistical Learning: with Applications in R*, the OLS linear model has several places it can be improved in: prediction accuracy and model interpretability. For the model of median home value, we are interested in doing both. As previously discussed, our shrinkage methods and best subset selection will help us increase prediction accuracy, but these methods can also help us with mode interpretability. In the data set Boston, we are dealing with thirteen predictor variables and one response variable, meaning keeping each variable in the model will lead to a model that is very complicated and difficult to interpret. Thirteen regression coefficients, each measuring entirely different factors, is too many to find a clear connection between median home value and a comprehensible set of predictors. In order to make the model more interpretable, we want to reduce the number of variables included in the model. This can be done with best subset or one of the shrinkage methods, the LASSO. The variety of methods we will be trying each have their own benefits both in computation and output, but the final optimal model we choose will be the model with the smallest predicted test error in MSE (means squared error).

Beyond the textbook’s contributions and my analysis, academic statistics has done much study on the benefits of techniques including subset selection and shrinkage methods. Contemporary statisticians (Oyeyemi et al. 2015) have found shrinkage methods to produce more accurate models than OLS in the presence of data with multicollinearity, which is a possible problem in the ‘Boston’ data set, I predict especially with property tax variable ‘tax’ and number of rooms variable ‘rm’. Furthermore, statisticians (Hastie et al. 2017) have used methods beyond the scope of this course to compare methods of variable selection, including best subset selection and the LASSO. They find that neither method always dominates the other in best model selection, but that best subset selection does outperform LASSO when the data has a high signal-to-noise ratio. From these papers, we can see that the current literature supports the usage of these methods to produce models that better predict median home value from the Boston data set.

Methods and Results:

The first task is to split the Boston data into a training and a test set. Because the Boston data set has 506 observations and 14 variables, I felt comfortable using the 50/50 split that is described in the textbook. Creating a training and a test set requires that I randomly send half of the 506 rows to the training set and the other half to the test set, though they must be independent of each other in this case, so I have sampled for each subset without replacement. I now have a training set upon which I can use my model refinement techniques and a test set on which I can measure the usefulness of the results of those model refinement techniques.

The first model refinement technique that we are tasked with implementing is best subset selection. As mentioned in the introduction, best subset selection will result in a model in which we only use a subset of the variables. Best subset selection is a computationally intensive methods that creates a different model for each possible combination of variables in the model. Because the Boston data set has thirteen predictors, best subset selection will produce 8,192 possible models and then select which combination of predictors creates the best model for predicting median home value. In order to implement best subset selection, I use the library ‘leaps’ and the function ‘regsubsets’. I then used the ‘reg.summary’ function to summarize which variables best subset selects for each of the possible models by number of variables included (Appendix 1). After that, I plotted RSS and Adjusted R-squared over the number of variables included in the best subset models (Appendix 2). These plots showed that RSS was minimized and Adjusted R-squared was maximized when eleven variables were included, so I created a linear regression model including the eleven variables selected by best subset selection. The model included every variable except ‘age’ (proportion of owner-occupied units built prior to 1940) and ‘indus’ (proportion of non-retail business acres per town). To create model diagnostics, I used the ‘plot’ function to plot the diagnostic plots of the linear regression model (Appendix 3). Next, I performed hypothesis tests on three variables in my linear regression model, ‘rm’, ‘tax’, and ‘dis’. These three variables were all statistically significant, and their outputs are available in Appendix 4. For the final part of Task 2, I assessed the accuracy of the selected model by using the model to predict the median home values in the testing set using the ‘predict’ function and then calculating the MSE of these predictions using the ‘mean’ function. This model, found from best subset selection, has an MSE of 153.1221.

The next model refinement technique we are tasked with implementing in Task 3 is the LASSO regression. For the LASSO model, I began by loading the ‘glmnet’ library for the ‘glmnet’ function. I then created the x matrix using the ‘model.matrix’ function and a model of medv from Boston, as well as the y vector from Boston and medv. Next, I created a training set and a test set with a 50/50 split using the ‘sample’ function. I finally used the ‘glmnet’ function to create the LASSO model from the training set, using the default lambda grid. Then, I needed to use cross-validation to find the best lambda, so I used the ‘cv.glmnet’ function to find the lambda with the smallest CV error (0.0141035). To figure out which predictors had their coefficient set to 0 by the LASSO model and which variables were still included, I refit the entire model using ‘glmnet’ again and the ‘predict’ function to find the coefficients with using the best lambda. The model created by LASSO set ‘age’ equal to zero and left every other variable, but notably ‘indus’ was still extremely small (Appendix 5). To conduct hypothesis tests, I used the library ‘hdi’. I find again that ‘rm’, ‘chas’ (Charles River bordering), and ‘tax’ are statistically significant. Finally, to assess the error of the LASSO model, I use the ‘predict’ function and the ‘mean’ function to find the MSE of the LASSO model much in the same way I did for the best subset model. I find that the MSE is equal to 26.85999, smaller than the best subset model.

The final model refinement technique that we are tasked with in Task 4 is to create a ridge regression model. Creating a ridge regression model is very similar to creating a LASSO model since they are both shrinkage models built from the ‘glmnet’ package, so much of my code is very similar. I created my ridge model using the ‘glmnet’ function and the x matrix and y vector I created for the LASSO model. However, instead of setting alpha=1 for LASSO, I set alpha=0 for ridge regression. Next, I used ‘cv.glmnet’ to find my best lambda for the ridge model, which is the lambda with the smallest CV error. The best lambda for the ridge regression is equal to 0.6546267, larger than the LASSO best lambda. The ridge regression model, unlikely the best subset model and the LASSO model, includes each predictor variable with the tuning parameter lambda applied. To view these new coefficients, I refit the model as I did with the LASSO using ‘glmnet’ and ‘predict’, the output of which is in Appendix 6. I conducted hypothesis tests the same way I as I did for the LASSO model, using the ‘hdi’ function. I found that ‘rm’ is still significant, ‘lstat’ (lower status of the population in percent) is significant, and ‘nox’ (nitrogen oxide particles in parts per million)’ is significant. For the assessment of the ridge regression model, I use the ‘predict’ function and the ‘mean’ function to determine the MSE of this model, which was determined to be 27.19152, slightly larger than the LASSO MSE but much smaller than the best subset model MSE.

Appendix of Results and Outputs:

Appendix 1:

A picture containing table

Description automatically generatedTable

Description automatically generated

The ‘\*’ symbols here indicate which variables are included for each model based on its number of variables included.

Appendix 2:

Chart, line chart

Description automatically generated

The point indicated on the Adjusted R-squared chart shows AR2 is maximized at 11 variables.

Appendix 3:

Chart, scatter chart

Description automatically generated

Appendix 4:

Text

Description automatically generatedText, letter

Description automatically generatedText

Description automatically generated

The p-values here are effectively 0, showing that these predictors are significant.

Appendix 5:

Table

Description automatically generated

We can see here that only one variable, ‘age’, is equal to 0, but many variables are quite small as a result of the shrinkage from the tuning parameter.

Appendix 6:

Table

Description automatically generated

No variables are equal to 0 under ridge regression, but many are quite small as a result of the shrinkage from the tuning parameter.

Task 5 and Discussion:

Task 5 asks us to consider the prediction accuracy of each of the three models and to choose a model that we prefer when considering prediction accuracy and model interpretability. The prediction accuracies of each model (best subset, LASSO, and ridge) are measured by the MSEs produced by each model, which respectively are: 153.1221, 26.85999, 27.19152. Because LASSO has the smallest MSE at 26.85999, we can see that it has the best prediction accuracy. However, in order to choose a model, we must also take into consideration model interpretability, often meaning we prefer a model with fewer prediction variables. However, the best subset, LASSO, and ridge models have 11, 12, and 13 predictor variables respectively, meaning choosing best subset over LASSO or ridge would not provide significant improvement in interpretability. Because of the superior prediction accuracy of the LASSO model and the small changes in model interpretability between the models, I would pick the LASSO model as my most preferred model.

From this project, I have concluded that the shrinkage methods were more applicable to the Boston data set with the goal of predicting median home value than the best subset selection method. Referring back to the 2017 paper by Hastie et al. from my introduction, LASSO was shown to outperform best subset selection when there was a low signal-to-noise ratio, so it is possible that this observation holds for the Boston data set as well. If I were to go further on this project and improve its analysis and results, I would do a more thorough review of the Boston data, possibly cleaning some outliers or high leverage points that could exist in the data as shown in the diagnostic plots. It is also possible that we could model median home value better by using non-linear modelling techniques, yet as we have not learned many of those yet, they were currently outside the scope of this project. The shrinkage methods used here and the best subset selection do improve on the OLS prediction accuracy, so these methods could be applied to other data sets that we have utilized in class, such as the College data set or the Credit data set. Because the Credit data set aims to predict credit default, it could be very beneficial for the interested statisticians to improve their prediction accuracy.

References:

Hastie et al. *Extended Comparisons of Best Subset Selection, Forward Stepwise Selection, and the Lasso*. arXiv:1707.08692v2

Oyeyemi et al. *On Performance of Shrinkage Methods – A Monte Carlo Study*. International Journal of Statistics and Applications 2015, 5(2): 72-76

Chapters 5 and 6 of *An Introduction to Statistical Learning: with Applications in R* by Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani

Code used in R:

Much of the code below was guided by the labs described in *An Introduction to Statistical Learning*.

library(MASS)

attach(Boston)

set.seed(1)

training=Boston[train,]

testing=Boston[-train,]

library(leaps)

regfit.full=regsubsets(medv~.,training,nvmax=13)

reg.summary=summary(regfit.full)

reg.summary

par(mfrow=c(2,2))

plot(reg.summary$rss ,xlab=" Number of Variables ", ylab=" RSS", type="l")

plot(reg.summary$adjr2 ,xlab =" Number of Variables ", ylab=" Adjusted RSq",type="l")

which.max (reg.summary$adjr2)

points (11, reg.summary$adjr2[11], col ="red",cex =2, pch =20)

lm.fit1=lm(medv~crim+zn+chas+nox+rm+dis+rad+tax+ptratio+black+lstat,data=training)

summary(lm.fit1)

plot(lm.fit1)

t.test(rm)

t.test(dis)

t.test(tax)

bestsub.pred=predict(lm.fit1,newx=x[testing,])

mean((medv-bestsub.pred)^2)

library(glmnet)

x=model.matrix(medv~.,Boston)

y=Boston$medv

set.seed(1)

train=sample(1:nrow(x),nrow(x)/2)

test=(-train)

y.test=y[test]

lasso.mod=glmnet(x[train,],y[train],alpha=1)

plot(lasso.mod)

set.seed(1)

cv.out=cv.glmnet(x[train,],y[train],alpha=1)

bestlam=cv.out$lambda.min

bestlam

out=glmnet(x,y,alpha=1)

lasso.coef=predict(out,type="coefficients",s=bestlam )[1:15,]

lasso.coef

install.packages("hdi", repos="http://R-Forge.R-project.org")

library(hdi)

hdi(rm)

hdi(tax)

hdi(dis)

lasso.pred=predict(lasso.mod,s=bestlam,newx=x[test,])

mean((lasso.pred-y.test)^2)

ridge.mod=glmnet(x[train,],y[train],alpha=0)

set.seed(1)

cv.out1=cv.glmnet(x[train,],y[train],alpha=0)

bestlam1=cv.out1$lambda.min

bestlam1

out1=glmnet (x,y,alpha =0)

predict(out1,type="coefficients",s=bestlam1)[1:15,]

hdi(rm)

hdi(lstat)

hdi(nox)

ridge.pred=predict(ridge.mod,s=bestlam1,newx=x[test,])

mean((ridge.pred-y.test)^2)

Contributions to this Project:

This project was completed solely by me, Emma Taylor.