Regression Analysis of Automobile Mileage

Illahi Khan, Travis Lovejoy, Rin Tran

March 13, 2013

Abstract

With hydrocarbon-based fuels rising to record highs across the globe, automobile users are becoming more aware of their fuel consumption. By identifying the various aspects of automobile design that most positively correlate to miles per gallon (M.P.G.), automobile manufacturers can better understand which trends are a positive force in M.P.G. and produce more fuel efficient cars. In our study, we analyze data collected by the University of California, Irvine, Machine Learning Repository. Our methodology to determine the aspects of automobile design that most accurately predict M.P.G. included constructing a correlation coefficient matrix via the various variables provided by the data set. Using this correlation coefficient matrix along with some other less-important determinants, we concluded that displacement, weight, and horsepower most accurately predicted M.P.G. We constructed a linear regression model using these three variables which accurately predicts mpg as verified by our calculated 95% confidence intervals.

Part I

Introduction

Rising fuel costs across the globe coupled with increased awareness of the environmental impacts of burning hydrocarbons have led drivers across the world to limit or watch their fuel consumption. Thus, a potential new automobile customer is more inclined to search for automobiles that yield a relatively high miles per gallon value. Therefore, it is important to analyze the various aspects of an automobile that most strongly contribute to an automobile's miles per gallon. Thus, a regression model would help automobile manufacturers better understand the tradeoffs required for better mileage which could result in faster introduction of low mileage cars onto the market. Our goal was to find a relatively accurate linear regressoin model for predicting mpg using the data that was available to us.

Part II
Terminology

Throughout this paper, we use certain terms that may prove to be ambigious or unnecessarily long when used in repetition. Here we attempt to clear any ambiguities and/or assign acronyms. The source of our data set is provided by the University of California, Irvine, Machine Learning Repository. This source will hereafter be referred to in the paper as the "UCI Machine Learning Repository." The data set provided by the UCI Machine Learning Repository has various attributes labeled as: mpg, cylinders, displacement, horsepower, weight, acceleration, model year, origin, and car name. These attribute names will be taken to mean their commonly accepted definitions relating to automobiles. The standard unit for measuring mileage of a vehicles in the U.S.A., miles per gallon, will be referred to as m.p.g., mpg, or mileage. Mathematical and statistical equations will be defined as they arise.

Part III
Correlation Coefficient Matrix

Using the statistical programming language R, we were able to construct a correlation coefficient matrix. Here note that correlation refers to the Pearson product-moment correlation coefficient which is a certain type of correlation calculation indicating the linear dependence between two predictor variables. The Pearson product-moment correlation coefficient for a sample population is calculated using the following forumula :

$$r = \frac{\sum_{i=1}^{n}(X_i - X)(Y_i - Y)}{\sqrt{\sum_{i=1}^{n}(X_i - X)^2} \cdot \sqrt{\sum_{i=1}^{n}(Y_i - Y)^2}}.$$

The correlation coefficient matrix is a matrix of the Pearson product-moment correlations between different variables (this is also called an intercorrelation matrix). The R programming language provides a library function for determining the correlation coefficient matrix. However, some modifications to the data set were required in order for the aforementioned function to work correctly. This was because the data set provided by the source from UCI Machine Learning Repository possessed gaps in some data variables. For example, the horsepower of a "Ford Pinto" (a type of automobile produced by the Ford Motor Company) is listed as "?". Thus, when the standard correlation coefficient matrix function was used in R, an error indicating that not all of the variables were numeric was thrown. We alleviated this problem by simply deleting the lines with incomplete data as necessary (admittedly, a quite crude solution). Since only horsepower values were missing for various associated automobiles, we only needed to remove the lines that were missing in horsepower when calculating

its correlation coefficient. This does not effect the correlation coefficient between horsepower and mpg because the lines containing no information about horsepower were very few in comparison to the lines containing horsepower information. The correlation coefficient matrix (intercorrelation matrix) is as follows:

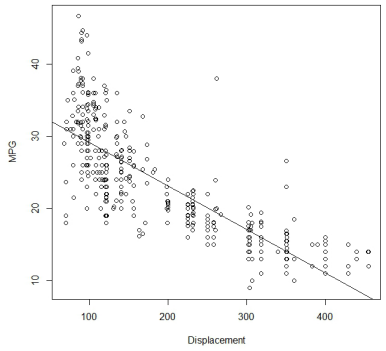
$$\begin{bmatrix} 1.000 & -0.778 & -0.805 & -0.778 & -0.832 & 0.423 & 0.581 & 0.565 \\ -0.778 & 1.000 & 0.951 & 0.843 & 0.898 & -0.505 & -0.346 & -0.569 \\ -0.805 & 0.951 & 1.000 & 0.897 & 0.933 & -0.544 & -0.370 & -0.615 \\ -0.778 & 0.843 & 0.897 & 1.000 & 0.865 & -0.689 & -0.416 & -0.455 \\ -0.832 & 0.898 & 0.933 & 0.865 & 1.000 & -0.417 & -0.309 & -0.585 \\ 0.423 & -0.505 & -0.544 & -0.689 & -0.417 & 1.000 & 0.290 & 0.213 \\ 0.581 & -0.346 & -0.370 & -0.416 & -0.309 & 0.290 & 1.000 & 0.182 \\ 0.565 & -0.569 & -0.615 & -0.455 & -0.585 & 0.213 & 0.182 & 1.000 \end{bmatrix}$$

Note that, from left to right and top to bottom, the variables are mpg, cylinders, displacement, horsepower, weight, acceleration, modelyear, and origin, respectively.

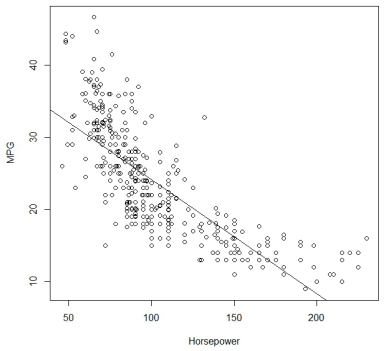
Part IV

Determining Predictor Variables

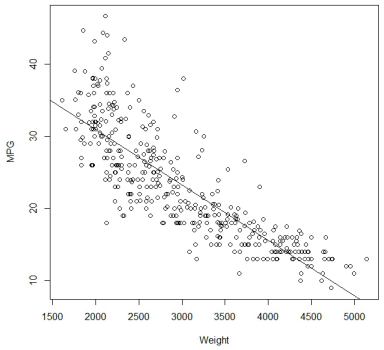
The above correlation coefficient matrix shows that the three variables most correlated to m.p.g. are displacement, weight, and horsepower. For further analysis, we studied the plots of these potential predictor variables.



The graph for displacement and mpg shows what the correlation coefficient implies. As displacement increases, mpg generally decreases. The graph shows perhaps a negative exponential distribution. However, a line, as shown in the graph, closely approximates the relationship with a fair amount of data points lying fairly close to the line and distributed evenly above and below the line.



Similarly, the graph for horsepower and mpg shows what the correlation coefficient implies. As horsepower increases, mpg generally decreases. Similar to the graph above. The graph shows perhaps a negative exponential distribution. However, a line, as shown in the graph, closely approximates the relationship with a fair amount of data points lying fairly close to the line and distributed evenly above and below the line.



Again similarly, the graph for weight and mpg shows what the correlation coefficient implies. As weight increases, mpg generally decreases. Similar to the graphs above, the graph shows perhaps a negative exponential distribution. However, a line, as shown in the graph, closely approximates the relationship with a fair amount of data points lying fairly close to the line and distributed evenly above and below the line.

We concluded that these three predictor variables were sufficient for our regression model. We also noted that displacement and weight were very correlated. This would prove useful when determining our linear model.

Part V
Linear Model

With the three chosen predictor variables we constructed multiple linear models using R's in-built library-provided linear modeling function (lm). Note that R's lm() function fits a linear model for given data sets supplied to it as arguments. We tested various models, all including displacement, weight, and horsepower as well as one in which we considered cylinders, as well. By analyzing summary information using R's model summary function (summary), we found that the model that was most accurate was the model that incorporated displacement, weight, and height. We further found that a more accurate model occurred when we paired displacement and weight because of their intercorrelation. The error produced from this model was very small in comparison to the other models and the multiple R squared value was at a very decent level. Since the multiple R squared value is a measure of how well the data points fall on the regression line of the model and the multiple R squared value for our linear regression model was 0.7462, we determined that our model was fairly accurate. Our regression model is given as follows:

$$Y = 57.87 - 0.07566 \cdot X_1 + 0.009043 \cdot X_2 + 0.06793 \cdot X_3 + 0.00002163 \cdot (X_1 \cdot X_3),$$

where Y is mpg, X_1 is displacement, X_2 is weight, and X_3 is horsepower.

Part VI
Confidence Interval

Using R's prediction function (predict), we were able to determine the confidence interval of our prediction. The confidence interval was determined at 95% and was found to be the following for the first three values of the independent variables (displacement, weight, and horsepower):

prediction	lower bound	upper bound
17.38925	16.55307	18.22542
14.73844	13.69466	15.78222
16.17922	15.12653	17.2319

This showed that our predictions fell within the 95% confidence interval of the actual value of mpg and thus further solidified our model's predictive power.

Part VII
Conclusion

Therefore, we concluded that the best variables for predicting mpg were displacement, weight, and horsepower. Using our linear regression model, these three variables can reliably predict mpg and thus our model can be used by automobile manufacturers to produce cars in ways that reduce mileage. A car manufacturer who wishes to entertain a certain demographic need not sacrifice mpg when taking into account other factors. For example, a manufacturer that markets as a high-horsepower car manufacturer can produce a more fuel efficient car at the same horsepower as their other cars by decreasing displacement and weight. The manufacturer can even increase horsepower and mileage by decreasing weight and displacement to certain values that can be found with our linear model.

Appendix

A Code

The following is code that was used in the making of this paper. Note that the code is in the R statistical programming language and requires a R compiler to run.

```
colNames = c("mpg", "cylinders", "displacement", "horsepower",
"weight", "acceleration", "modelyear", "origin", "carname")

carData <- read.table("auto-mpg.data", col.names = colNames)

### Delete the attribute "carname" along with its data as it
### serves no purpose in this project
carData = carData[-9]

summary(lm(carData$mpg ~ carData$displacement
+ carData$horsepower+ carData$weight))

### Delete all rows with empty data
for(i in 1:398){
        if(carData[i,4] == "?")
                carData = carData[-i,]
}

### Fix "horsepower" attributes into numeric values
hpV = as.vector(carData$horsepower)
```

```
hpV2 = as.integer(hpV)
carData$horsepower = hpV2

corMatrix = cov(carData)

### Looking at the intercorellation matrix, we can extract
### important attributes that will help us compute mpg.
### Because displacement and weight have a high correlation
### value, we fix the equation so that they modify each
### other.
red2 = lm(mpg ~ displacement*weight + horsepower
, data = carData)
full = lm(mpg ~ displacement + cylinders + weight +
horsepower + acceleration + modelyear, data = carData)

anova(red1,full)

### A 95% confidence interval is given by
CI = predict(red2,data.frame(displacement = 307,
horsepower = 130, weight = 3504), interval = "confidence")
CI[2]
### and
CI[3]

### Test whether our equation is within the 95%
### confidence interval
for(i in 1:6){
        print(predict(red2,data.frame(displacement =
        carData$displacement[i], horsepower =
        carData$horsepower[i], weight = carData$weight[i]),
        interval = "confidence"))
}

head(carData)
```

B References

- Matloff, Norm. From Algorithms to Z-Scores: Probabilistic and Statistical Modeling in Computer Science. Davis: University Of California, 2012. Print.
- J. L. Rodgers and W. A. Nicewander. Thirteen ways to look at the correlation coefficient. The American Statistician, 42(1):59–66, February 1988. Print.

- "R: Fitting Linear Models." R: Fitting Linear Models. N.p., n.d. 13 Mar. 2013. <http://stat.ethz.ch/R-manual/R-devel/library/stats/html/lm.html>. Web.

C Credits and Contact

While this paper truly was a group effort that resulted in a very educational experience for all of us, we are required to list contributions made by each group memeber. They are listed below, alphabetically by last name.

Illahi Khan (ibkhan@ucdavis.edu) - code and write-up and ideas/inputs
Travis Lovejoy (talovejoy@ucdavis.edu) - math and ideas/inputs
Rin Tran (rvtran@ucdavis.edu) - code and ideas/inputs