Stats Lab 6

Jon Ashbrock

February 28, 2018

1 Introduction

After today's lab, the student will:

1. Be able to plot and interpret the residuals of a model
2. Know how to index lists with logical statements
3. Be able to build multivariate regression models in **R**
4. Be able to build arbitrary polynomial regression models in **R**
5. Understand the concept of "over-fitting" the model to the given data and the bias-variance trade-off

2 Plotting the Residuals of a Model

Recall that the residuals are defined to be the difference between the predicted and observed data. To determine the "goodness" of a model, it can be helpful to plot the residuals in order to identify patterns. In the following discussion, $\hat{y}$ will denote the predicted y value for a given x. Thus, given a data set $\{(x_i, y_i)\}_{i=1}^n$, and a model $\hat{y} = f(x)$ we could plot the residuals as follows (note, the following is pseudo-code):

For each x_i, compute the predicted value $\hat{y}_i = f(x_i)$.
Compute the error vector $y_i - \hat{y}_i$
Plot(x, errors)

How to use residuals to analyze a model: The residuals tell you first and foremost how large the error is. Large error means that a model may not be great. Another way to use the residuals is to look for patterns when plotting the residuals against your predictive variable. If there is a clear pattern in this plot, then there is a relationship in our data which we have not accounted for. In our case, we will use a higher order polynomial when we see patterns. In general the residuals could suggest a trigonometric function or exponential function based on the shape. When a model is **as complex as it needs to be** the residuals should look like random "noise". It is usually best to select a model which is the simplest model which has no pattern in the residuals. This will be discussed more later in the over-fitting section.

Discussion 1. *Example of residuals on exponential*

3 Indexing lists with logical operators

Now that we have seen how to filter outliers by hand in **R**, we can now use the logical indexing feature of **R** to do this more quickly. For instance if I want to filter the countries with large population out of a list, I can simply do: which will remove those population values which are greater than 1000.

```
pop <- pop[pop<1000]
```

I can also filter other vectors by logical operations involving another vector. Therefore I can filter the military data to get rid of those data points in the military list if their corresponding population value is an outlier. Therefore, to filter both lists I can do:

```
mil <- mil[pop<1000]
pop <- pop[pop<1000]
```

Discussion 2. *Why will this not work if we run the lines in the reverse order?*

4 Multivariable Regression

Sometimes we may expect our output variable to be a function of more than one input variable. For this part we will be trying to predict a dependent variable y as a linear function of n independent variables $x_1, \ldots x_n$. Thus we will compute a linear function of the x_i to predict y:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n \tag{1}$$

As in the case of last class (where $n = 1$), we pick the parameters $\beta_0, \ldots \beta_n$ that minimize the error on our given sample. The theory is much the same in this case and the parameters have the same interpretation as in the previous case. Therefore this section will be devoted to the implementation in **R**.

Luckily, we can use the lm function to do the multivariable regression. We simply need to use the correct formula. If our variables are named Y and $x1, \ldots xn$ in our program, then the formula could look like:

$$formula = Y \sim x1 + x2 + \ldots xn$$

Note that if there is no ambiguity in where to get the data from, we do not need to include the "data =" variable in the lm function. That is, if we have lists called $x1, x2, \ldots xn$ and one called Y then simply typing $lm(formula = Y \sim x1 + \ldots xn)$ will return the multivariate model.

5 Polynomial Regression Models

Polynomial regression models can be realized using the multi-variable linear regression model. For example, given X to predict Y and we want to use a quadratic model, the following method will work. Create a new variable (perhaps called $X2$) where the values in $X2$ are the squares of the corresponding values in X. This can be done with $X2$<- ^ X^2. Then do a multi-variable linear regression using the variables X and $X2$.This will give a linear model in the variables x and x^2 which is a quadratic model. All of the above analysis now applies.

6 Over-Fitting in Models

When building predictive models, we are trying to predict future observations of that variable. Data always has inherent noise built in so it should not be

the most important thing for our model to fit the previously observed data perfectly. Otherwise, we are trying to fit a model to random noise. What is more important is that the model predicts new data accurately. Put another way, we want our model to predict the underlying phenomenon absent of any noise. If our underlying phenomenon is linear, logarithmic, or quadratic, for example, then it does not make sense to use a high degree polynomial to model the relationship even if the noise causes this polynomial to fit our data better (with a higher R^2) than a line.

Therefore, much care should be taken in constructing polynomial regression models of high degree. These types of models suffer from the "bias-variance" problem. Higher complexity models have lower bias (fit the existing data better) but have higher variance (predict new data worse). Anyone who wants to learn more about this can feel free to come talk to me as this is quite close to my actual research area.

The important takeaway from this section is to use the simplest model that fits the data reasonably well. To determine whether or not it fits the data reasonably well, one should look at the plot of the residuals and confirm that there are no "patterns" to the residuals. That is, the model we use should cause the residuals to appear to be random numbers which are independent of the x value.