A World of Events

A Blog for Event and Data Analytics

Demystifying Linear Regressions as a Tool for Inference

Linear regressions are one of the simplest algorithms for predicting quantitative responses. In fact, some people may even consider it dull when compared to other advances approaches, like Support Vector Machines. However, I find that not only linear regressions provide good average prediction results, but, more importantly, its simplicity and transparency makes it an ideal tool for trying to understand the data in itself, that is, the relationship between the predictor (explanatory) variables and the responses. In this sense, linear regressions are better suited as a mechanism for inferring the data, rather predicting it.

The best way of arguing for this is by going through an example. As usual, let's use R, and load the mtcars data-set (1974 Motor Trend US magazine comprising of the fuel consumption and 10 aspects of automobile design and performance for 32 automobiles):

```
> head(mtcars)
                  mpg cyl disp hp drat wt    qsec vs am gear carb
 Mazda RX4         21.0 6  160 110 3.90 2.620 16.46 0 1  4    4
 Mazda RX4 Wag     21.0 6  160 110 3.90 2.875 17.02 0 1  4    4
 Datsun 710        22.8 4  108 93  3.85 2.320 18.61 1 1  4    1
 Hornet 4 Drive    21.4 6  258 110 3.08 3.215 19.44 1 0  3    1
 Hornet Sportabout 18.7 8  360 175 3.15 3.440 17.02 0 0  3    2
 Valiant           18.1 6  225 105 2.76 3.460 20.22 1 0  3    1
```

Next, let's create a simple linear regression model for this data. We will use just some of the predictor variables that seem more important, like number of cylinders (cyl), horsepower (hp), weight (lb/1000), 1/4 mile time (qsec), and number of gears (gear). We construct the linear regression model by specifying the response variable (mpg) and the predictor variables (cyl + hp + wt + qsec + gear). This is calling fitting the model to the training data. Here is an example:

```
> fit = lm(mpg ~ cyl + hp + wt + qsec + gear, data=mtcars)
> summary(fit)

Call:
lm(formula = mpg ~ cyl + hp + wt + qsec + gear, data = mtcars)

Residuals:
Min 1Q Median 3Q Max
-3.3969 -1.5852 -0.5171 1.0712 5.5914

Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept) 26.96517 15.13161 1.782 0.08643 .
cyl -0.45775 0.83952 -0.545 0.59023
hp -0.01808 0.01671 -1.082 0.28923
wt -3.41354 1.02454 -3.332 0.00259 **
qsec 0.38753 0.55312 0.701 0.48975
gear 0.72536 1.13460 0.639 0.52821
—
Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.576 on 26 degrees of freedom
Multiple R-squared: 0.8468, Adjusted R-squared: 0.8173
F-statistic: 28.74 on 5 and 26 DF, p-value: 8.227e-10
```

The sheer amount of information is overwhelming, which tends to make people ignore the details. As we shall see, all the data in the fitted model are equally important, and need to be considered. Let’s take this in parts.

First, let’s assess the accuracy of the fitted model by using the metrics located at the end of the summary:

```
Residual standard error: 2.576 on 26 degrees of freedom
Multiple R-squared: 0.8468, Adjusted R-squared: 0.8173
```

Both the residual standard error (RSE) and the R-squared metrics represent how far are the estimated responses from the actual responses (when considering the training data). The RSE value of 2.57 indicates that on average the estimated response deviates in 2.57 mpg from the actual response.

For example, let’s apply this to the first row of the data, which is for ‘Mazda RX4’:

```
cyl hp wt qsec gear -> mpg
6 110 2.620 16.46 4 -> 21.0
```

If we apply the coefficients generated by the fitted model to the first row of ‘Mazda RX4", we get the estimated mpg of 22.5669. The actual response should have been 21.0 (as it can be seen in the data itself), hence its residual (deviation) is 1.5669 (i.e. 22.5669 – 21.0), which is pretty close to the RSE of

2.57.

If you missed it, how did we exactly come up with the estimated mpg response of 22.57? This is an example of a prediction, and really just means applying the values of the predictor variables (i.e. cyl, hp, wt, qsec, gear) to the estimated coefficients provided by the regression model (i.e. -0.45, -0.01, -3.41, 0.38, -0.72). In the case of 'Mazda RX4', the values are respectively 6 (cyl), 110 (hp), 2.62 (wt), 16.46 (qsec), and 4 (gear). Hence, the calculation becomes _estimated-MazdaRX4-mpg = 6 * -0.45775 + 110 * -0.01808 + 2.62 * -3.41354 + 16.46 * 0.38753 + 4 * 0.72536 + 26.96517_. The last number is the intercept and represents the baseline when all other predictor variables are zero. It is like saying that the mpg would be 26.96 should the car have no cyl, hp, wt, etc. In this particular case, it obviously makes no sense, but in other cases, it does actually help you establish a baseline. Of course, there is an automated way of predicting in R, which yields exactly the same value as the above calculation:

```
> predict(fit, data.frame(cyl=c(6), hp=c(110), wt=c(2.62), qsec=c(16.46), gear=c(4)))
22.5669
```

Now that you understand the concept of residuals, the Residuals summary in the beginning of the model summary should also make sense:

```
Min      1Q      Median  3Q     Max
-3.3969 -1.5852 -0.5171  1.0712 5.5914
```

This is saying that in the worst case there is a deviation of 5.59 in the mpg response, and that most of the estimates deviate between -1.5 to 1.07 mpg from the actual responses.

Let's get back to the RSE. The issue with RSE is that it is a relative metric. Is a residual of 2.57 good or bad? To answer this question, you need to consider its unit, which in this case is mpg, and the general context of the problem. I don't know much about cars, so it is hard to say if a 2.57 error is a large error or something that can be ignored. To answer this, let's look at the R-squared. The R-squared is a proportion, it measures how well the model fits the data by comparing the residual variance to the total variance of the training data. In other words, it verifies if the variance of the estimated response is related to the fitted model or is it already inherent in the response before the regression is performed. The R-squared ranges from 0 to 1. A value close to 1 means that there is a good fit, and conversely a value close to 0 means that perhaps a linear model is not a good model for the data, and some other approach should be tried. In this example, the value of 0.81 indicates that we have the right model, but there is some room for improvements.

Finally, the F-statistics and p-value are used to determine if there is correlation between the predictor variables and the responses. For easy of reference, here are their values in our example:

```
F-statistic: 28.74 on 5 and 26 DF, p-value: 8.227e-10
```

What we are trying to establish is if the variance in the estimated response is just a matter of chance, or does it really relate somehow to the predictor variables? In mathematical terms, if there is no correlation, then it means that the coefficients all tend to be zero (i.e. if _y_ = _coef_ * _x_, and if y doesn't vary with changes to x, then coef must be equal to zero), and the variances seen in the estimated responses are related to the standard deviation itself of the actual responses. In this case, F-statistics becomes the ratio of the standard deviation (or rather the square) by itself and is equal to 1 or a small number close to 1. If there is correlation, then F-statistics is some other higher number. The p-value is the probability of getting this F-statistics by chance. If it is a very small value, generally below .05, it means that the likelihood of just being unlucky and getting this same value is very low, and therefore unlikely.

To summary it, we are looking for:

○o R-squared close to 1, and
○o F-statistics higher than 1, and
○o p-value very low (i.e. < 0.05)

If this is the case, then it means that there is some (linear) correlation between the predictor variables and the response and the linear regressions is likely doing a good job of fitting the training data.

Having established that the linear regression model is good, next let's look at what it gives us. We do this by considering the predictor variable coefficients and its parameters. Let's start with the wt variable:

```
      Estimate  (coef) Std. Error   t value   Pr(>|t|)
   hp -0.01808          0.01671     -1.082    0.28923
```

This is saying that (holding all other variables constant) a change to the motor horsepower causes a -0.018 change to miles per gallon consumption. In math terms, this is equivalent to *mpg = -0.018 * hp + others*. This in itself is very interesting information. It tells us that if we were to increase the motor's hp by 100 for the Mazda RX4, then its mpg will decrease from 21.0 to 19.2! This gives us a great tool for inference and analysis.

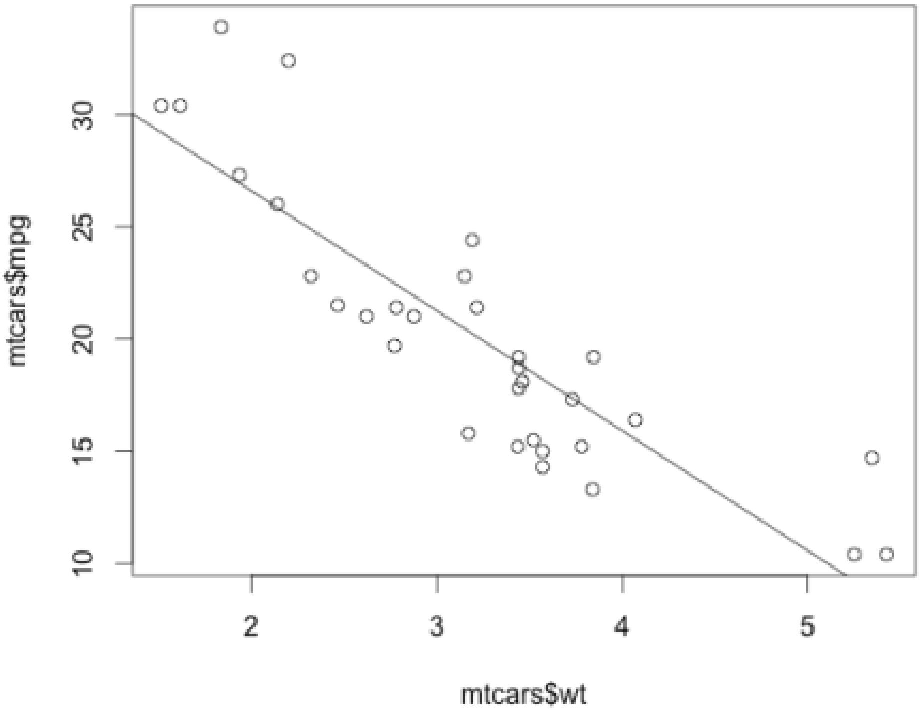
However, we must be careful, and again consider the full set of information provided before taking full measures. For example, the model tells us that there is a chance of error for this coefficient of 0.016. In other words, the correct coefficient could vary, in average, by 0.016 from the starting value of -0.018. The t-value tells us that the standard error is high in terms of the coefficient value we have. This is correct, the error is almost as much as the coefficient itself, hence the t-value close to 1. As rule of thumb, we are looking for high t-values. The expectation is that the coefficient varies in the range of [coef - 2 * SE, coef + 2 * SE]. This represents a confidence level of 95%. If you consider that the t-distribution is similar to a normal distribution, and that in a normal distribution, 95% of the values are within 2 standard deviation, then it makes sense to think that you have a 95% confidence that the coefficient value varies within two standard errors. In this case, it means that the hp coefficient can go from -0.05 to 0.014, which is not very assuring. Further, the Pr value says that there is 28% chance of us having gotten this coefficient by chance, that is, rather than because of correlation between hp and mpg. Again, this is not very assuring, we would generally like a Pr of less than 5%, that is, Pr < 0.05.

Let's look for other predictors that do better. Luckily, R simplifies this for us by placing increasing levels of '*' next to those variables that seemly correlate better. In our case, this is the wt predictor variable:

```
       Estimate   Std. Error   t value   Pr(>|t|)
   wt  -3.41354   1.02454      -3.332    0.00259 **
```

The model tells us that there is a 0.2% chance of wt and mpg NOT being correlated, which is very low. And the t-value is also reassuringly high. Hence, it seems safe enough for us to assume that an increase in weight in average causes a decrease of 3.4 in mpg. Again, very powerful and useful information!

To confirm this finding, let's plot the values of wt versus mpg for the training data, and then draw a line representing the regression model of *mpg = intercept + coefficient * wt*:



(http://adcalves.files.wordpress.com/2014/04/screen-shot-2014-04-17-at-8-42-09-am.png)

As it can be seen, these two variables are indeed highly correlated and linear in nature.

There is a lot of material in this article, however there are really just two important take-aways.

First, linear regression is like the white-box testing in machine learning, and what it lacks in accuracy, it more than compensates in transparency, allowing us to infer about the data, rather than just do black-box crystal-ball like predictions.

Second, don't ignore the details, they are there for a reason, and need to be considered.

This entry was posted on Thursday, April 17th, 201About these ads (http://en.wordpress.com/about-Learning, R. You can follow any responses to this erthese-ads/) h the RSS 2.0 feed. You can leave a response, or trackback from your own site.

The Contempt Theme.
Create a free website or blog at WordPress.com.

Follow