Lecture 3 - Supervised Learning: Regression

Sondre Elstad

2022-06-25

Introduction

In the next two lectures, we will study a couple of very classic techniques in statistics: linear regression and logistic regression. Recall that linear regression is applied in the cases where we have a quantitative response variable. When estimating how different factors will affect wage levels, we are dealing with a quantitative response variable (wage level). We run into these kinds of problems all of the time. Indeed, the technique you are about to learn is indeed the most basic and, yet, most important technique in all of statistics. It is the basis for conceptual understanding of all other statistical models. Yet, it is one of the most interpretable and useful techniques available.

Single Regression

We will start with the case in which there is only one predictor X. Mathematically, we can formulate the assumed relationship between the predictor and the response as follows:

$$Y \approx \beta_0 + \beta_1 X.$$

The predictor for Y is $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$. Notice that this is a prediction for one specific datum, x_i. The residual for observation (or datum) i is $e_i = y_i - \hat{y}_i$. It is the difference between the actual response for observation i and the predicted response for observation i. We define the residual sum of squares (RSS) as

$$RSS = e_1^2 + e_2^2 + ... + e_n^2.$$

We want to minimize the prediction error and, in this case, that implies minimizing RSS. Put differently, we choose β_0 and β_1 to minimize RSS. This approach is called Ordinary Least Squares (OLS). We can show that the minimizing coefficients are

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2},$$

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x},$$

where $\bar{y} \equiv \frac{1}{n}\sum_{i=1}^n y_i$ and $\bar{x} \equiv \frac{1}{n}\sum_{i=1}^n x_i$. However, as we saw in the case of $E(\epsilon) = 0$, it is important to consider the variation around the mean. This is also true for statistical estimators such as $\hat{\beta}_0$ and $\hat{\beta}_1$. We measure the variation of an estimator using the standard error (which is really the square root of the variance of the estimator). It can be shown that the standard errors of the $\hat{\beta}_0$ and $\hat{\beta}_1$ are given by

$$SE(\hat{\beta}_0)^2 = \sigma^2 \left[\frac{1}{n} + \frac{\bar{x}^2}{\sum_{i=1}^n (x_i - \bar{x})^2} \right], \quad SE(\hat{\beta}_1)^2 = \frac{\sigma^2}{\sum_{i=1}^n (x_i - \bar{x})^2},$$

where $\sigma^2 = Var(\epsilon)$. σ^2 comes from the irreducible error. It might be irreducible, but we are able to estimate it from our data. It is given by the formula $RSS = \sqrt{\frac{RSS}{n-2}}$.

Based on these standard errors, we are able to define confidence interval. Assume that we estimate $\hat{\beta}_0$ and $\hat{\beta}_1$ using 10 different training samples (of course, with the same features and response variables). A 95% confidence interval is defined as the interval which will include the true values β_0 and β_1. We formulate these intervals as follows:

$$\hat{\beta}_i \pm 2 \cdot SE(\hat{\beta}_i), \quad i \in \{0, 1\}$$

Note that prediction intervals are always wider than confidence intervals. The reason is that they account for the irreducible error ϵ. Closely related to these confidence intervals are the notion of hypothesis testing. Very often, we are interested in uncovering whether there exists a statistically significant relationship between predictors and the response variable. By 'statistically significant relationship' we mean a relationship which is sufficiently 'far from' 0 to warrant a causal interpretation. To operationalize this, consider the following hypothesis test:

$$H_0 : \text{There is no relationship between } X \text{ and } Y$$

versus the alternative hypothesis

$$H_a : \text{There is some relationship between } X \text{ and } Y.$$

We can reformulate this in mathematical terms as follows:

$$H_0 : \beta_1 = 0$$

versus

$$H_a : \beta_1 \neq 0.$$

Testing this hypothesis entails computing a t-statistic:

$$t = \frac{\hat{\beta}_1 - 0}{SE(\hat{\beta}_1)},$$

This measures the distance between $\hat{\beta}_1$ and 0 in number of standard deviations. The t-statistic is based on a corresponding t-distribution. For n greater than 30, this closely resembles the standard normal distribution $N(0, 1)$. The simple form of the t-distribution implies that we have the ability to incur a lot of information from it. In fact, assuming the t-distribution, we can compute the probability of observing as large a value for $\hat{\beta}_1$ as we do. This probability is called the p-value. If the p-value is large, this implies that our estimate is nothing special (in that the probability of observing such a value is rather large). However, if $p < 0.05$, we reject the null hypothesis at the 5% confidence level. In other words, if $p < 0.05$, 0 will not be included in the 95% confidence interval.

So, we have our significant predictors. But saying that the predictor exerts a 'statistically significant' influence on the response variable is not necessarily the same as saying that the predictor exerts a large influence on the response variable. Put differently, even though the predictor is statistically significant, it might still explain a rather small portion of the overall variance of Y. It makes sense to use the predictor to predict Y, but there might be any variables that it would make sense to predict Y with. We need a better assessment of the fit of the predictive model. We can use R^2 for this purpose:

$$R^2 = \frac{TSS - RSS}{TSS} = 1 - \frac{RSS}{TSS}$$

where $TSS = \Sigma(y_i - \bar{y})^2$ is the total sum of squares. $TSS - RSS = ESS$ (explained sum of squares). An R^2 close to 1 indicate that the predictor explains a lot of the overall variation in Y.

Multiple Regression

In general we might consider a suite of p predictors. A multiple linear regression models looks like

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_p X_p + \epsilon,$$

where X_j is the jth predictor and β_j measures the relationship between the jth predictor and the response. Just as for the single regression case, we can test hypotheses concerning the predictors:

$$H_0 : \beta_1 = \beta_2 = ... = \beta_p = 0$$

versus the alternative

$$H_a : \text{at least one } \beta_j \text{ is non-zero.}$$

However, we can no longer rely on the t-statistic. If p is large, this implies a large number of repeated testing. This increases the likelihood of errant conclusions. We need a test which allows us to test all the parts of the above hypothesis simultaneously. The F-statistic gives us what we need:

$$F = \frac{(TSS - RSS)/p}{RSS/(n-p-1)}.$$

Under the linear model assumption, we could show (we're not going to, though) that

$$E\left\{\frac{RSS}{n-p-1}\right\} = \sigma^2$$

and, provided that H_0 is true,

$$E\{(TSS - RSS)/p\} = \sigma^2.$$

However, if the opposite H_a is true, we have $E\{(TSS - RSS)/p\} > \sigma^2$, implying $F > 1$. To sum up: If H_0 is true, F is likely to be close to 1. If H_a is true, $F >$ is likely. We could restrict our test to a chosen subset q of predictors:

$$H_0 : \beta_{p-q+1} = \beta_{p-q+2} = ... = \beta_p = 0.$$

In this case, we fit a restricted model in which all these q predictors are set to zero. We obtain the restricted RSS, RSS_0, and compute

$$F = \frac{(RSS_0 - RSS)/q}{RSS/(n-p-1)}.$$

Issues of fit are just as crucial, if not more, for multiple regression models as for single regression models. R^2 must be non-decreasing in the number of predictors (including more information cannot imply that we explain less of the variation in Y). In fact, R^2 is increasing in the number of predictors. There are alternative measures we could look at, such as adjusted R^2, but we will not discuss these here. It suffices to say that the linear regression models minimizes the expected forecast error by maximizing R^2. In the multiple regression setting, we can show that $R^2 = Corr(Y, \hat{Y})^2$. By choosing the model that maximizes R^2, we maximize this correlation.

R Lab (ISL 2017)

For some of the next lectures (including this one), we will use R for playing around with the ideas encountered in each lecture. Please note that these R Labs draw heavily from the R Labs in ISL 2017 (for all intents and purposes it is copy-paste) and, as such, I take no credit for the codes or for the cases.

We load the MASS package, which is a very large collection of data sets and functions

```
library(MASS)
library(ISLR2)
```

```
##
## Attaching package: 'ISLR2'
```

```
## The following object is masked from 'package:MASS':
##
##     Boston
```

We will start by using the lm() function to fit a simple linear regression model, with medv as the response and lstat as the predictor. The basic syntax is lm(y~x, data), where y is the response, x is the predictor, and data is the data set in which these two variables are kept.

```
lm.fit <- lm(medv ~ lstat, data = Boston)
attach(Boston)
lm.fit <- lm(medv ~ lstat)
```

If we type lm.fit, some basic information about the model is output.

```
lm.fit
```

```
##
## Call:
## lm(formula = medv ~ lstat)
##
## Coefficients:
## (Intercept)        lstat
##       34.55        -0.95
```

```
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ lstat)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.168  -3.990  -1.318   2.034  24.500
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 34.55384    0.56263   61.41   <2e-16 ***
## lstat       -0.95005    0.03873  -24.53   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.216 on 504 degrees of freedom
## Multiple R-squared:  0.5441, Adjusted R-squared:  0.5432
## F-statistic: 601.6 on 1 and 504 DF,  p-value: < 2.2e-16
```

The predict() function can be used to produce confidence intervals and prediction intervals for the prediction of medv for a given value of lstat.

```
predict(lm.fit, data.frame(lstat = (c(5, 10, 15))),
        interval = "confidence")
```

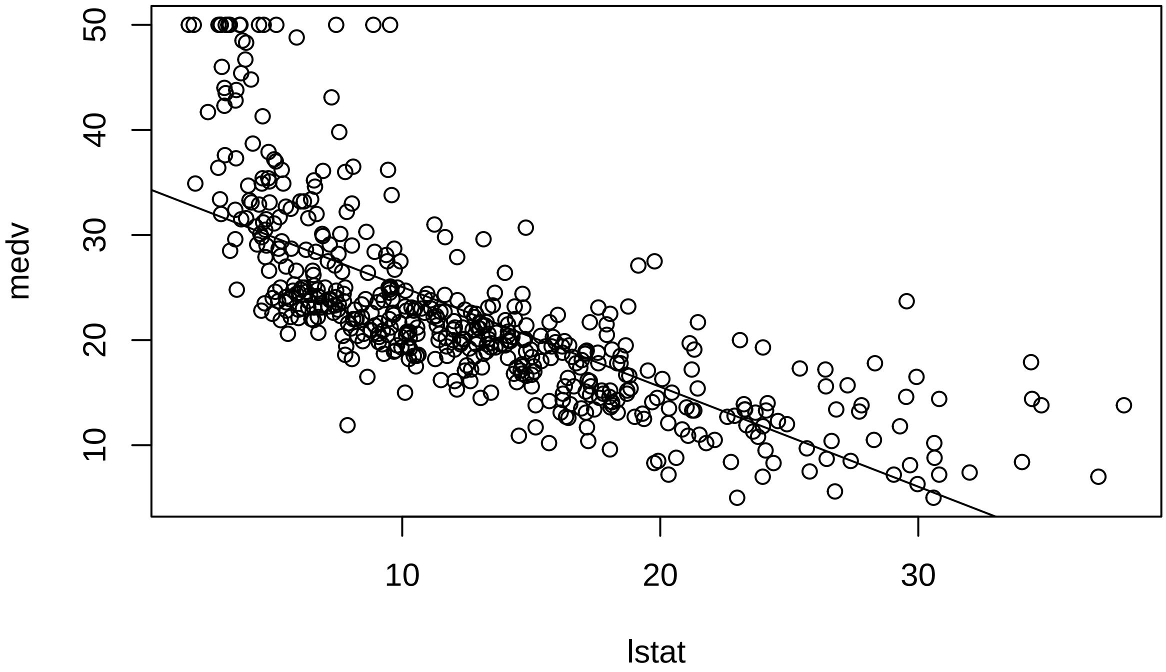
```
##        fit      lwr      upr
## 1 29.80359 29.00741 30.59978
## 2 25.05335 24.47413 25.63256
## 3 20.30310 19.73159 20.87461
```

```
predict(lm.fit, data.frame(lstat = (c(5, 10, 15))),
        interval = "prediction")
```

```
##        fit       lwr      upr
## 1 29.80359 17.565675 42.04151
## 2 25.05335 12.827626 37.27907
## 3 20.30310  8.077742 32.52846
```

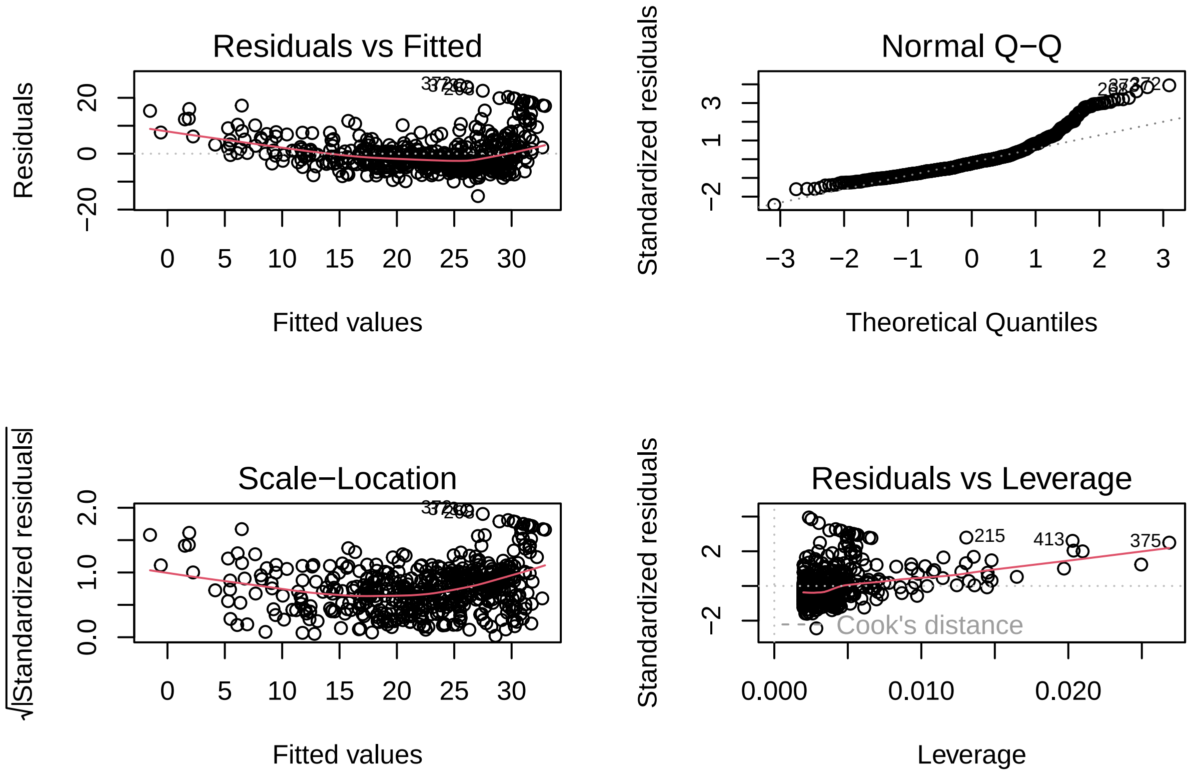
We will now plot medv and lstat along with the least squares regression using the plot() and abline() functions.

```
plot(lstat, medv)
abline(lm.fit)
```



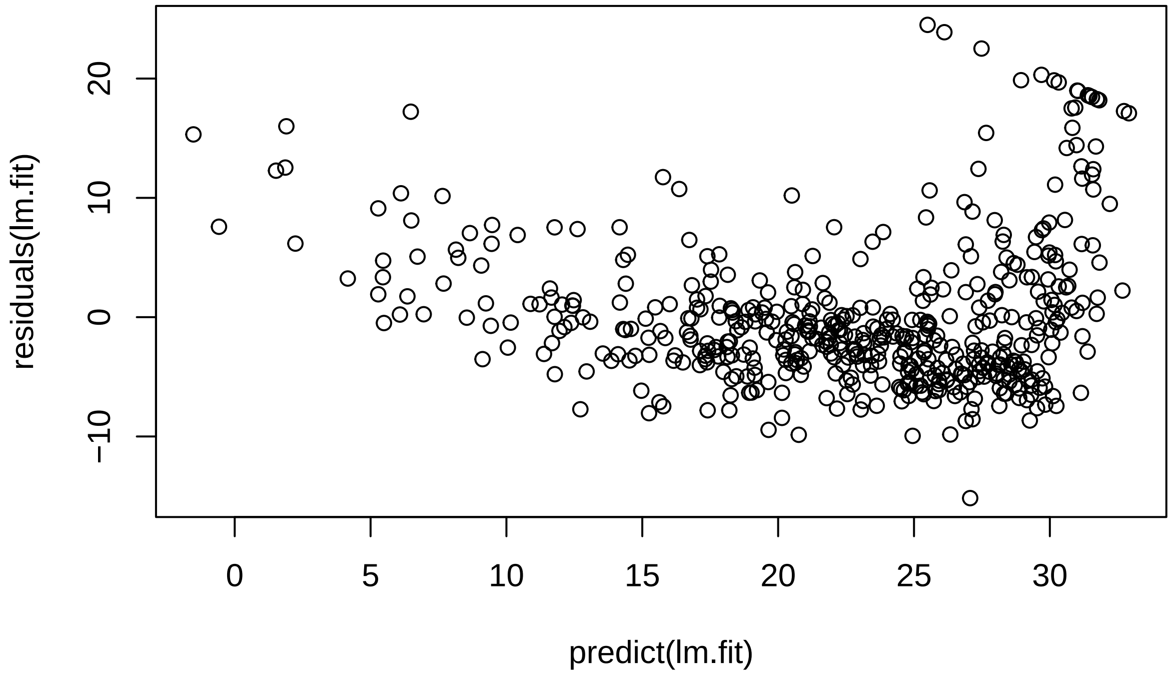
Four diagnostic plots are automatically produced by applying the plot() function directly to the output from lm(). However, it is often convenient to view all four plots together. We can achieve this by using the par() and mfrow() functions, which tell R to split the display screen into separate panels so that multiple plots can be viewed simultaneously. For example, par(mfrow=c(2, 2)) divides the plotting region into a 2 × 2 grid of panels.

```
par(mfrow = c(2, 2))
plot(lm.fit)
```



Alternatively, we can compute the residuals from a linear regression fit using the residuals() function.

```
plot(predict(lm.fit), residuals(lm.fit))
```



In order to fit a multiple linear regression model using least squares, we again use the lm() function.

```
lm.fit <- lm(medv ~ lstat + age, data = Boston)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ lstat + age, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.981  -3.978  -1.283   1.968  23.158
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 33.22276    0.73085  45.458  < 2e-16 ***
## lstat       -1.03207    0.04819 -21.416  < 2e-16 ***
## age          0.03454    0.01223   2.826  0.00491 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.173 on 503 degrees of freedom
## Multiple R-squared:  0.5513, Adjusted R-squared:  0.5495
## F-statistic:   309 on 2 and 503 DF,  p-value: < 2.2e-16
```

The following syntax results in a regression using all predictors except age.

```
lm.fit <- lm(medv ~ . - age, data = Boston)

summary(lm.fit)
```

```
##
## Call:
## lm(formula = medv ~ . - age, data = Boston)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -15.1851  -2.7330  -0.6116   1.8555  26.3838
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  41.525128   4.919684   8.441 3.52e-16 ***
## crim         -0.121426   0.032969  -3.683 0.000256 ***
## zn            0.046512   0.013766   3.379 0.000785 ***
## indus         0.013451   0.062086   0.217 0.828577
## chas          2.852773   0.867912   3.287 0.001085 **
## nox         -18.485070   3.713714  -4.978 8.91e-07 ***
## rm            3.681070   0.411230   8.951  < 2e-16 ***
## dis          -1.506777   0.192570  -7.825 3.12e-14 ***
## rad           0.287940   0.066627   4.322 1.87e-05 ***
## tax          -0.012653   0.003796  -3.333 0.000923 ***
## ptratio      -0.934649   0.131653  -7.099 4.39e-12 ***
## lstat        -0.547409   0.047669 -11.483  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 4.794 on 494 degrees of freedom
## Multiple R-squared:  0.7343, Adjusted R-squared:  0.7284
## F-statistic: 124.1 on 11 and 494 DF,  p-value: < 2.2e-16
```