Bias-Variance Trade-off

Microeconometrics and Application: Report 1

Simrit Rattan (27720199656249)
Elena Riccarda Ziege (27720199656247)

October 24, 2019

1 Statistical Background

To assess the performance of a certain model, the data set first needs to be seperated into a training and a test data set. The training data set is used to construct the model. Once a learning method has been chosen, it is important to evaluate its out-of-sample performance in terms of its prediction capability on independent test data. (cf. Hastie, et al., 2001, p. 219 ff.) The reason why the performance is not evaluated on the training data is because we are always interested in predicting the future, meaning new data, not the past. Measuring the quality of fit on the training data would only tell us how well the estimated model performs on the data used for estimation. (cf. Gareth, et al., 2013, p. 29 ff.) A very good prediction on the training data set could just mean that the model used some information given in the error term of the used training data set and not only the relation between the explanatory and the dependent variable (cf. Belloni, et al., 2014). We use the Mean Squared Error (MSE) to measure the prediction capability of a learning method in the following way:

$$MSE = E[(y - f(x)^2)] = \sigma^2 + [Biasf(x)]^2 + Var(y - f(x)) \tag{1}$$

(cf. Le Calonnec, 2017)

We denote y as our real function and *f(x)* as the estimation. From equation (1) we see that the MSE is decomposed into three parts: the bias of *f(x)*, the variance of the difference between y and *f(x)* and the irreducible error, also called noise, which occurs because the real function is not directly available. Regardless of how good our model is, we can never predict the data perfectly because of the noise in the data. (cf. Gareth, et al., 2013, p. 29 ff.)

The bias is the expected difference between the real function and our estimation of it. In general the bias decreases as the model complexity increases.

$$Bias(f(x)) = E[(f(x) - y] \tag{2}$$

(cf. Sherrer, 2018)

The variance is the amount by which our *f(x)* would change if we estimated it using different training data (cf. Gareth, et al., 2013, p. 29 ff.). It is the difference of our prediction *f(x)* and the true values y in our data set (cf. Bayesie, 2019). The variance of our estimate should not vary too much between different sets of training data, thus a low variance is preferable (cf. Gareth, et al., 2013, p. 29 ff.). But with increasing model complexity the variance increases too (cf. Gareth, et al., 2013, p. 29 ff.).

$$Var(y - f(x)) = E[(y - (f(x))^2] - E[y - f(x)]^2 \tag{3}$$

(cf. Bayesie, 2019)

Our aim would be to minimize the MSE over an independent test sample, which is also known as the generalization error. When the model complexity increases, the test MSE will first decrease, but after a while it will increase again. The training error on the other hand will decrease with an increase in the model complexity, which is why it is not a good measure for the quality of fit. We need to find the degree of complexity which minimizes the test error. (cf. Hastie, et al., 2001, p. 219 ff.)

The test MSE can be high due to two reasons:

- Overfitting occurs when the relation between the training data and the model is too close. This means that the model gets used to the patterns in the training data and thus can't generalize well on other data, which results in a high variance.

- Underfitting is when the model can't describe the training data as well as the test data, because it did not manage to find a relation between x and y with the little information it got from the training data. Underfitting means that our bias is high. (cf. Le Calonnec, 2017)

The aim is to choose a model that simultaneously obtains a low variance and bias. This is called the Bias-Variance Trade-off. (cf. Gareth, et al., 2013, p. 29 ff.)

2 Empirical Implementation

2.1 Model comparison

2.1.1 Reading and preparing the data

```r
setwd("~/Documents/Master/3. Semester/Microecon/Report 1")
data <- read.csv("Credit.csv", header=T)
```

For our analysis we use the dependent variable *Limit*, which reflects a person's credit limit and which we regress on the explanatory variable *Income*, the person's yearly income. We split our data set into a training data set consisting of 60 percent of the observations and a test data set consisting of the other 40 percent. We will use the training data set to construct different models and afterwards compare these models using the test data set. We will compare a linear regression with two more complex models and one model using a transformed variable. At first we will compare them using the adjusted R^2 and the Akaike information criterion (AIC). As the adjusted R^2 describes the proportion of the variation explained by the model, we desire a high value, as opposed to the AIC, where a low value indicates a model with a small test error. (cf. Gareth, et al., 2013, p. 210 ff.)

```r
set.seed(2019)
idx.train <- caret::createDataPartition(y = data$Limit, p = 0.6, list = FALSE)
train <- data[idx.train, ]
test <-  data[-idx.train, ]
```

2.1.2 Linear regression

At first, we estimate a linear regression model regressing a person's credit limit on her income. We can observe that there is a statistically significant, positive relation between income and a person's credit limit in this model, as would be expected. It has an adjusted R^2 of 0.6261 and an AIC of 4168.105.

```r
linear <- lm(Limit ~ Income, data=train)
summary(linear)
```

```
##
## Call:
## lm(formula = Limit ~ Income, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2676.14 -1132.74    15.86  1281.31  2464.88
```

```
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  2435.20     148.09   16.45   <2e-16 ***
## Income         52.69       2.63   20.03   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 1417 on 238 degrees of freedom
## Multiple R-squared:  0.6277, Adjusted R-squared:  0.6261 
## F-statistic: 401.3 on 1 and 238 DF,  p-value: < 2.2e-16
```

```r
AIC(linear)
```

```
## [1] 4168.195
```

2.1.3 Cubic model

Next we make the model a little bit more complex by using the cubic value of the person's income. Here we can see that, in contrast to the linear term, the quadratic and cubic terms of a person's income do not have a statistically significant relation with the credit limit in this model. Compared to the linear model, the AIC increases while the adjusted R^2 decreases. This indicates that the cubic model performs worse than the linear model.

```r
cubic <- lm(Limit ~ poly(Income, degree = 3), data=train)
summary(cubic)
```

```
## 
## Call:
## lm(formula = Limit ~ poly(Income, degree = 3), data = train)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -2821.41 -1151.90    14.45  1294.81  2501.34 
## 
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)    
## (Intercept)                4767.88      91.67  52.011   <2e-16 ***
## poly(Income, degree = 3)1 28391.04    1420.14  19.992   <2e-16 ***
## poly(Income, degree = 3)2  -859.29    1420.14  -0.605    0.546    
## poly(Income, degree = 3)3 -1163.30    1420.14  -0.819    0.414    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 1420 on 236 degrees of freedom
## Multiple R-squared:  0.6293, Adjusted R-squared:  0.6246 
## F-statistic: 133.6 on 3 and 236 DF,  p-value: < 2.2e-16
```

```r
AIC(cubic)
```

```
## [1] 4171.143
```

2.1.4 Polynomial model of degree 7

As a third approach we estimate a model containing a polynomial of degree 7 of the income variable. Again there is a positive relation between income and the credit limit in this model but still only a statistically significant relation between the linear term of the person's income and the credit limit. The polynomial model has a higher adjusted R^2 but also a higher AIC than the linear model. Thus it is inconclusive which model should be chosen.

```
polynomial7 <- lm(Limit ~ poly(Income, degree = 7), data=train)
summary(polynomial7)
```

```
##
## Call:
## lm(formula = Limit ~ poly(Income, degree = 7), data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2714.0 -1090.6    -5.5  1269.2  2593.8
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  4767.88      91.34  52.201   <2e-16 ***
## poly(Income, degree = 7)1 28391.04    1415.00  20.064   <2e-16 ***
## poly(Income, degree = 7)2  -859.29    1415.00  -0.607   0.5443
## poly(Income, degree = 7)3 -1163.30    1415.00  -0.822   0.4119
## poly(Income, degree = 7)4   514.82    1415.00   0.364   0.7163
## poly(Income, degree = 7)5   370.21    1415.00   0.262   0.7938
## poly(Income, degree = 7)6  1889.81    1415.00   1.336   0.1830
## poly(Income, degree = 7)7  2734.54    1415.00   1.933   0.0545 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1415 on 232 degrees of freedom
## Multiple R-squared:  0.6383, Adjusted R-squared:  0.6273
## F-statistic: 58.48 on 7 and 232 DF,  p-value: < 2.2e-16
```

```
AIC(polynomial7)
```

```
## [1] 4173.298
```

2.1.5 Level-log model

Now we transform the variable of interest, income, by logarithm. It can be seen that, as the income increases by one percent, the person's credit limit increases on average by 2495 units, which is quite a large effect. The logarithmic model performs worse than the linear model according to the AIC and the adjusted R-squared.

```
logarithmic <- lm(Limit ~ log(Income), data=train)
summary(logarithmic)
```

```
##
## Call:
## lm(formula = Limit ~ log(Income), data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3067.9 -1310.7    -1.9  1335.8  4435.2
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -4068.6      529.8  -7.679 4.13e-13 ***
## log(Income)   2495.1      146.9  16.989  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1562 on 238 degrees of freedom
## Multiple R-squared:  0.5481, Adjusted R-squared:  0.5462
## F-statistic: 288.6 on 1 and 238 DF,  p-value: < 2.2e-16
```

```r
AIC(logarithmic)
```

```
## [1] 4214.725
```

When we compare these different models using the adjusted R^2, the polynomial model would be chosen while the linear model would be chosen based on the AIC. But one has to keep in mind, that the adjusted R^2 and the AIC both prefer models using less parameters.

2.1.6 Prediction

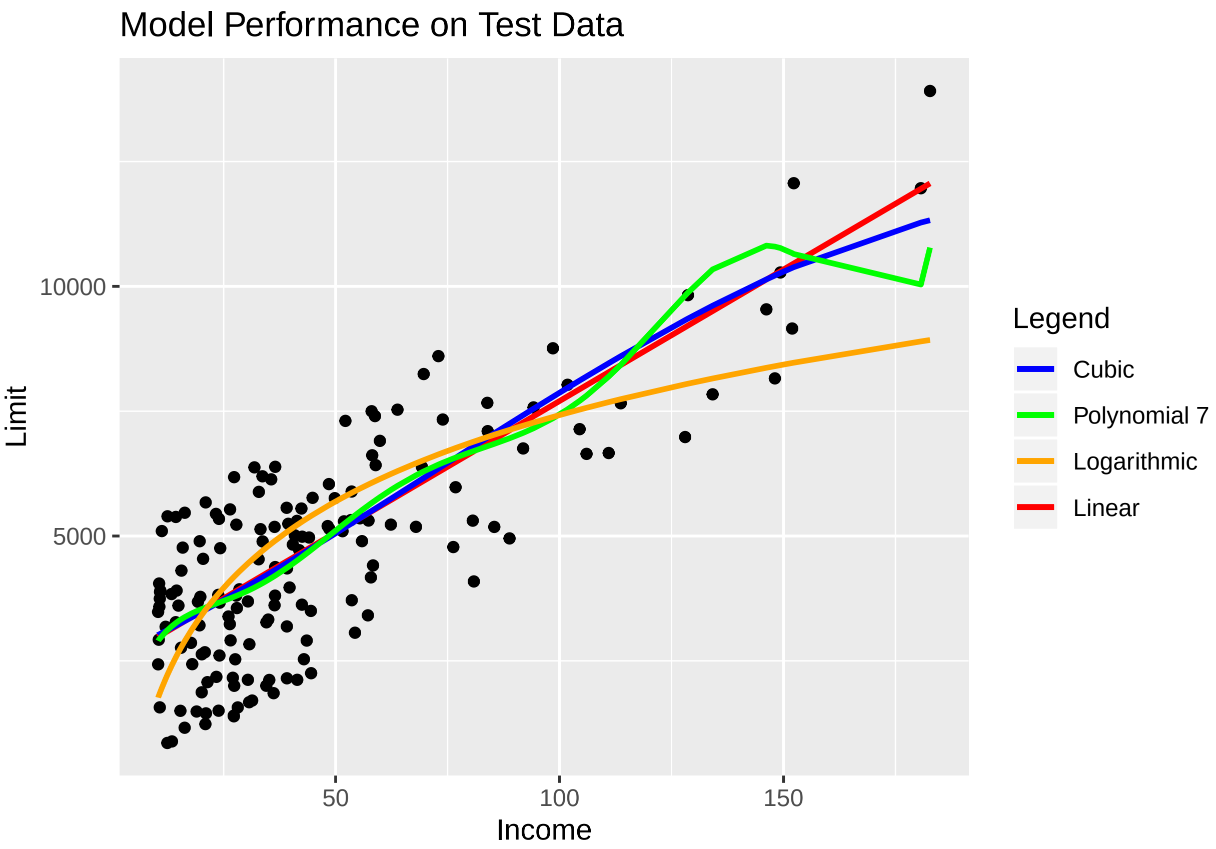
Now we predict the different models using the test data set to compare their performance.

```r
test$pred_linear <- predict(linear, newdata=test)
test$pred_cubic <- predict(cubic, newdata=test)
test$pred_polynomial7 <- predict(polynomial7, newdata=test)
test$pred_logarithmic <- predict(logarithmic, newdata=test)
```

2.1.7 Plot

Next we use the predictions of the different models we made on the test data set to plot them in a `ggplot()` of limit against income. In the plot we can compare how well the different models fit the test data set. The red line portrays the linear model, the blue one the cubic model, the green one the polynomial model of degree 7 and the orange line portrays the level-log model.

```r
library(ggplot2)
ggplot(test, aes(x = Income, y = Limit, colour =  ) ) +
  geom_point() +
  geom_line(aes(y = test$pred_linear, color="red"), size = 1) +
  geom_line(aes(y = test$pred_cubic, color="blue"), size = 1) +
  geom_line(aes(y = test$pred_polynomial7, color="green"), size = 1) +
  geom_line(aes(y = test$pred_logarithmic, color="orange"), size = 1) +
  scale_color_manual(name = "Legend",
                     values = c("red" = "red", "blue" = "blue", "green" = "green",
                                "orange" = "orange"),
                     labels = c("Cubic", "Polynomial 7", "Logarithmic","Linear"  )) +
  labs(x = "Income", y = "Limit", title="Model Performance on Test Data")
```



From the plot we can conclude that the transformed model does not describe the data well at higher values of income. The cubic and the polynomial model of degree 7 differ from the linear model as the income increases. The latter even showing a big variation as the income increases, which could be seen as a worse fit. You can see a deviation from the curve of the polynomial of degree 7 for very high income values that we cannot explain. This could be due to outliers in the income data.

2.2 Bias-Variance-Tradeoff

Now we compare the different models according to their test bias and their test variance and try to find out which model is the best in terms of the test MSE. As already mentioned, the bias should decrease and the variance should increase as we increase the degree of model complexity. Our aim is to then choose the model that minimizes the test MSE.

```r
models <- list(linear,
               cubic,
               polynomial7,
               logarithmic)
variance <- list()
bias <- list()
MSE <- list()
for (i in 1:length(models)) {
  currentPrediction <- predict(models[[i]],newdata=test)
  variance[i] <- var(test$Limit - currentPrediction)
  bias[i]<- mean(currentPrediction - test$Limit)
  MSE[i]<- mean((currentPrediction - test$Limit)^2)
}
```

```
output <- cbind(variance,bias, MSE)
colnames(output) <- c("Variance", "Bias", "MSE")
rownames(output) <- c("Linear", "Cubic", "Polynomial 7", "Logarithmic")
output
```

```
##              Variance Bias     MSE
## Linear       1951549  204.9069 1981338
## Cubic        1985948  201.9355 2014314
## Polynomial 7 2049349  188.1375 2071936
## Logarithmic  2613404  216.2628 2643840
```

We measure the performance of the linear model compared to two more complex and one transformed model. First of all, we look at the difference between the linear, the cubic and polynomial model of degree 7. As expected, the variance does increase with increasing complexity and the bias decreases. Due to the fact that taking the logarithm of a variable does not add to the complexity of the model, we do not see this kind of trend in the bias, even though the variance increases. In fact, the bias as well as the variance are more preferable in a linear model than in a logarithmic one. Finally, seeing the result makes it obvious that, for our given data set, the linear model minimizes the test MSE and thus should be chosen as the one best describing the relationship between a persons’s income and the credit limit. This could be because a linear model has a better generalizability and thus fits a different data set (our test data) better.

3 Literature

Bayesie, C., 2019. A deeper look at Mean Squared Error. [Online] (Available at: https://www.countbayesie.com/blog/2019/1/30/a-deeper-look-at-mean-squared-error([Accessed 17 10 2019].

Belloni, A., Chernozhukov, V. & Hansen, C., 2014. High-dimensional methods and inference on structural and treatment effects. Journal of Economic Perspectives, pp. 29-50.

Gareth, J., Witten, D., Hastie, T. & Tibshirani, R., 2013. An Introduction ot Statistical Learning with Applications in R. 1 Hrsg. Stanford: Springer.

Hastie, T., Tibshirani, R. & Friedman, J., 2001. The Elements of Statistical Learning - Data Mining, Inference and Prediction. 2 Hrsg. Stanford: Springer.

Le Calonnec, Y., 2017. CS229 Bias-Variance and Error Analysis. [Online], Available at: http://cs229.stanford.edu/section/error-analysis.pdf,[Accessed: 2019/10/13].

Sherrer, C., 2018. The Bias-Variance Decomposition. [Online], Available at: https://cscherrer.github.io/post/bias-variance/, [Accessed: 2019/10/17].