Nonlinear regression and Cross-validation

Statistical Learning

Carranza-Alarcón Yonatan-Carlos[1]

[1]Université de technologie de Compiègne

Outline

1. Cook's Distance

2. A "perfect" linear regression versus a Non-linear regression

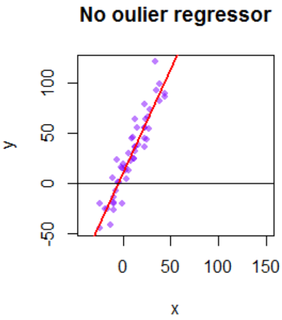
3. Nested Cross-validation

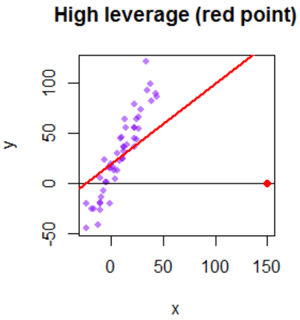
Overview

Cook's Distance

Data points with large residuals (outliers) and/or high leverage may distort the outcome and accuracy of a regression.

$$D_i = \frac{\sum_{j=1}^{n}\left(\widehat{y}_j - \widehat{y}_{j(i)}\right)^2}{ps^2}, \quad \text{where} \quad s^2 = \frac{\mathbf{e}^\top \mathbf{e}}{n-p}$$





If Cook's distance of the observation i is bigger, so this one influences in the estimation of β.

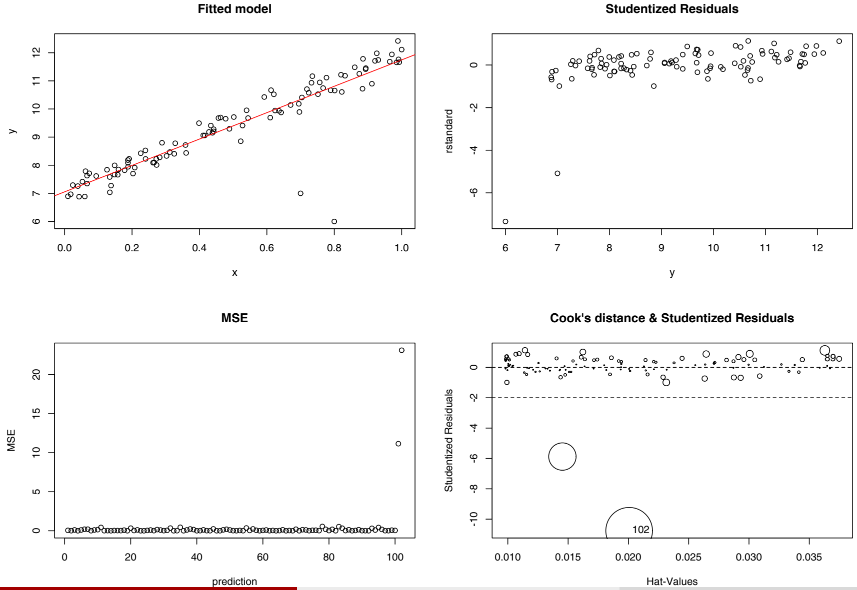
Linear regression - Outlier

Given the following simulated data set $\mathcal{D} = \{(x_i, y_i)\}$, with 2 outlier points:

$$y_i = 5x_i + 7 + \epsilon, \quad x_i \sim \mathcal{U}(0,1), \quad \epsilon \sim \mathcal{N}(0, \sigma = 0.3)$$
$$\mathcal{D} = \mathcal{D} \cup \{(0.7, 7), (0.8, 6)\}$$

```r
# linear simulation + outlier
x <- runif(100)
y <- 5*x + 7 + rnorm(100, sd = 0.3)
# outlier points
x <- c(x, 0.7, 0.8)
y <- c(y, 7, 6)
plot(x, y, main="Fitted model")
fit.linear <- lm(y~x)
summary(fit.linear)
abline(fit.linear$coefficients[1], fit.linear$coefficients[2], col="red")
plot(y,rstandard(fit.linear),ylab='rstandard',main="Studentized Residuals")
plot((y-fitted(fit.linear))^2, ylab='MSE', xlab="prediction", main="MSE")
influencePlot(fit.linear, main="Cook's distance & Studentized Residuals")
```

Exploring training data set



Overview

Linear regression

Theoretical linear model

Let us consider the two following theoretical linear model:

$$\mathcal{D}_1: \; y_i = 4 + 5\sin(x_i) + \epsilon_i, \; x_i \sim \mathcal{U}(0,10), \epsilon \sim \mathcal{N}(0, \sigma = 1) \quad \text{(Nonlinear)}$$

$$\mathcal{D}_2: \; y_i = 4 + 5 * x_i + \epsilon_i, \; x_i \sim \mathcal{U}(0,10), \epsilon \sim \mathcal{N}(0, \sigma = 3) \quad \text{(Linear)}$$

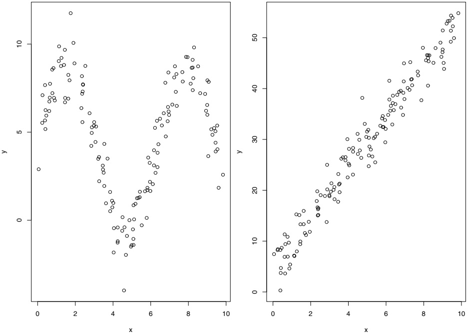
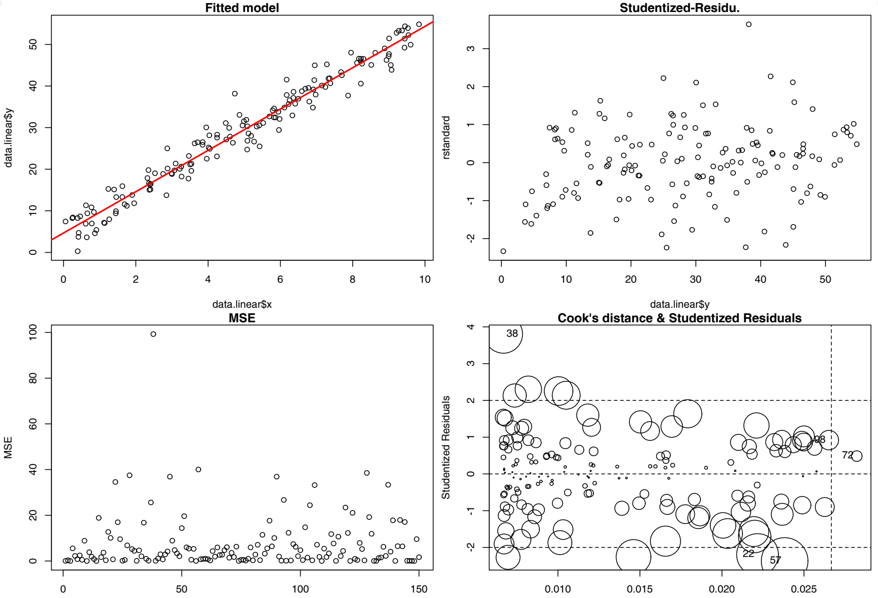
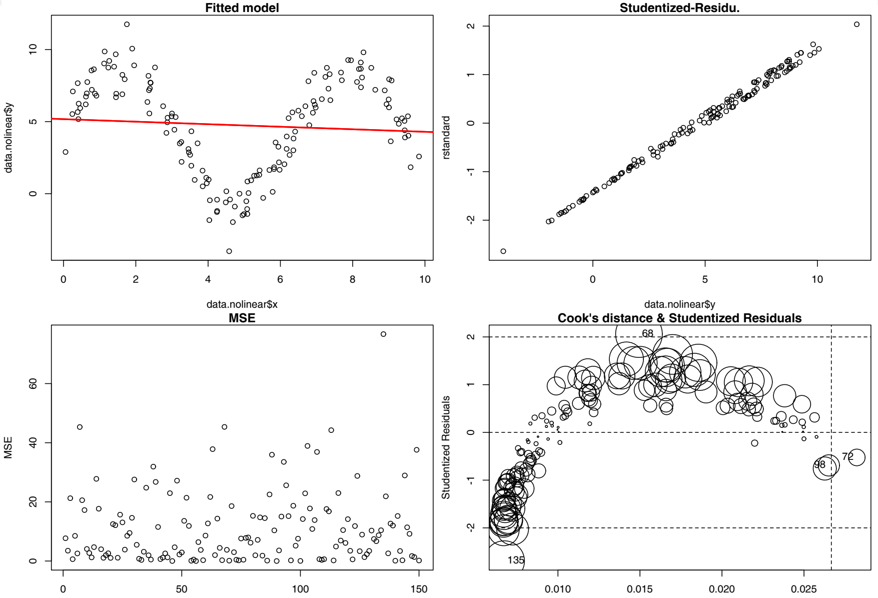


Figure: Nonlinear (left: $\mathcal{D}_1$) and linear (right: $\mathcal{D}_2$) data generated.

Exploring linear regression



Exploring non-linear regression



Polynomial regression model

Given $\boldsymbol{y_i}, \boldsymbol{x_i}, \beta_0 \in \mathbb{R}$ and $\beta_* \in \mathbb{R}$, we may consider the following models:

$$\boldsymbol{y} = \beta_0 + \boldsymbol{x_i}\beta_1 + \boldsymbol{x_i}^2\beta_2 \tag{Quadratic model}$$

$$\boldsymbol{y} = \beta_0 + \boldsymbol{x_i}\beta_1 + \cdots + \boldsymbol{x_i}^6\beta_k \tag{Polynomial model of degree 6}$$

$$\boldsymbol{y} = \beta_0 + \boldsymbol{x_i}\beta_1 + \ln(\boldsymbol{x_i})\beta_2 \tag{Logarithm model}$$

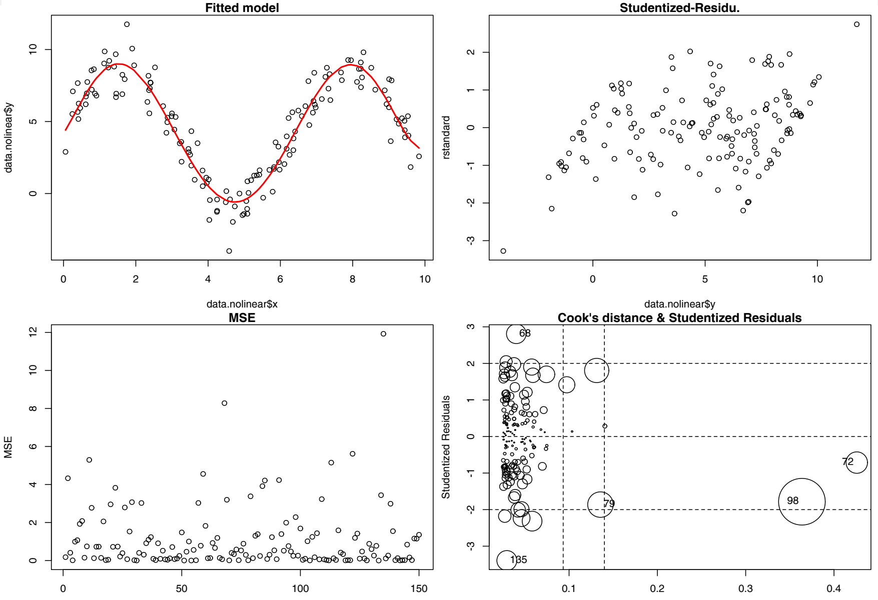
$$\boldsymbol{y} = \beta_0 + \boldsymbol{x_i}\beta_1 + \exp(\boldsymbol{x_i})\beta_2 \tag{Exponential model}$$

$$\ldots \tag{Infinity Combinations}$$

I would like to use the polynomial model of degree 6, i.e. (in R):

```r
fit.nonlinear <- lm(y~ 1 + poly(x, 6, raw=T), data=data)
```

Polynomial regression model



Overview

Nested and non-nested Cross-validation

Hyper-parameter

Tuning a hyper-parameter of the statistical model.

Estimation

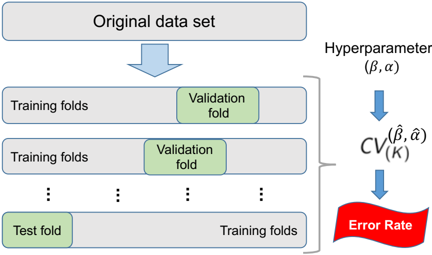
Estimation of the parameters of the statistical model.

Comparing

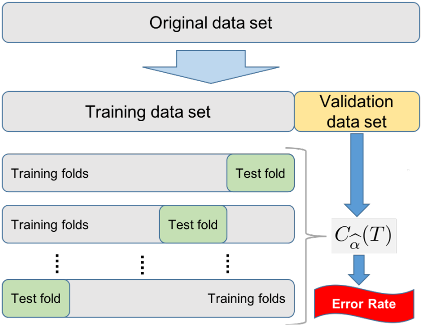
Comparing performance of different statistical models.

Nested and non-nested Cross-validation

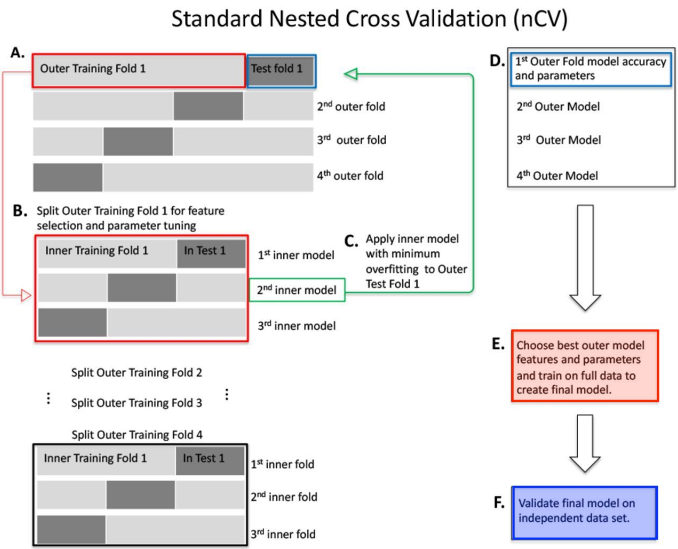
Non-nested cross-validation



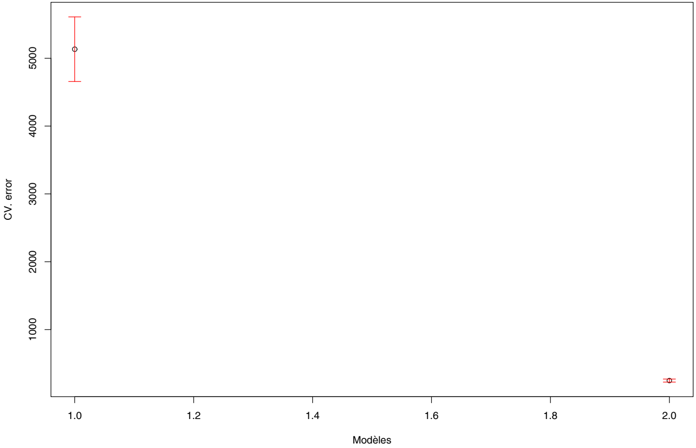
Nested cross-validation



Nested Cross-validation[1]



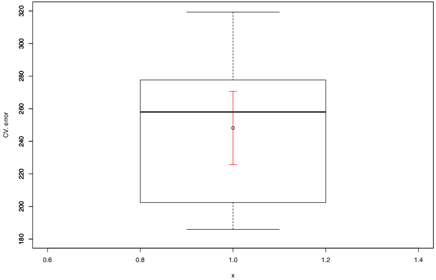
Regularized regression vs K-nearest-neighbor



KNN(left) and Regression elastic-net(right)

Conclusion

- We always choose the statistical model that has the lowest mean error and and the lowest variability.
- Boxplot $\neq$ Interval confidence of the error of cross-validation.



References

Saeid Parvandeh et al. "Consensus features nested cross-validation". In: *Bioinformatics* 36.10 (2020), pp. 3093–3098.