TP04 - Correction

Regression in Machine Learning

Carranza-Alarcon Yonatan-Carlos[1]

[1]UMR CNRS 7253 Heudiasyc
Université de technologie de Compiègne

December 16, 2019

Outline

Overview

Regression problem

Problem: Let $\mathcal{D} = \{(\boldsymbol{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y} \subseteq \mathbb{R}^p \times \mathbb{R}\}_{i=1}^n$ be a training data.

Given the regression model:

$$y_i = \beta_0 + \beta^T \boldsymbol{x}_i + \epsilon_i \iff \boldsymbol{y} = \mathbf{1}\beta_0 + X\boldsymbol{\beta} + \boldsymbol{\epsilon}, \text{ where } \forall i, \epsilon_i \sim \mathcal{N}(0, \sigma^2)$$

Vector Form

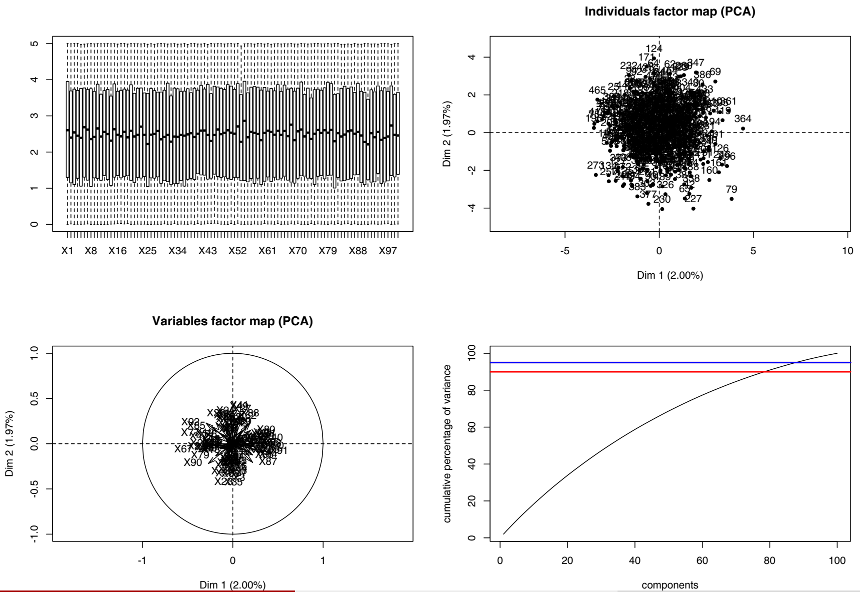
$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \vdots \\ \beta_p \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

Goal: We aim to fit the following regression model

$$y_i = \beta_0 + \beta\Phi(\boldsymbol{x}_i) + \epsilon_i$$

where $\Phi(\cdot)$ may be linear function, quadratic function, product of different nonlinear functions .....

Exploring training data set



Simple linear regression

```
> reg.fit <- lm(y~., data=data)
> summary(reg.fit)
Call:
lm(formula = y ~ ., data = data)

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   -0.84662   19.43296  -0.044 0.965272
X1            -0.24752    0.77686  -0.319 0.750182
X2            -0.95624    0.79786  -1.199 0.231432
.....
X95            3.09699    0.78918   3.924 0.000102 ***
X96           -0.90510    0.79252  -1.142 0.254116
X97          -23.52413    0.77522 -30.345  < 2e-16 ***
X98            1.02083    0.79911   1.277 0.202181
X99           -7.45701    0.81183  -9.185  < 2e-16 ***
X100           6.58068    0.82907   7.937 2.11e-14 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 22.99 on 399 degrees of freedom
Multiple R-squared:  0.9526,^^IAdjusted R-squared:  0.9407
F-statistic: 80.17 on 100 and 399 DF,  p-value: < 2.2e-16
```

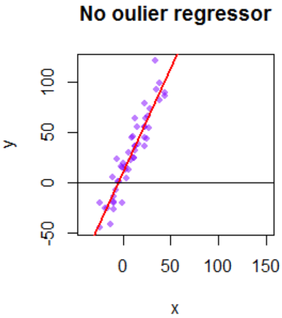
(X) * Degree significance (not enough to select a covariable)**

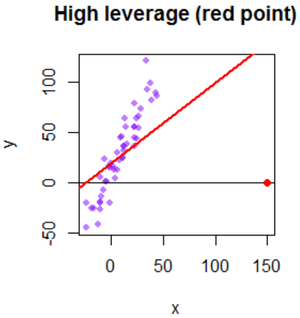
Overview

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.

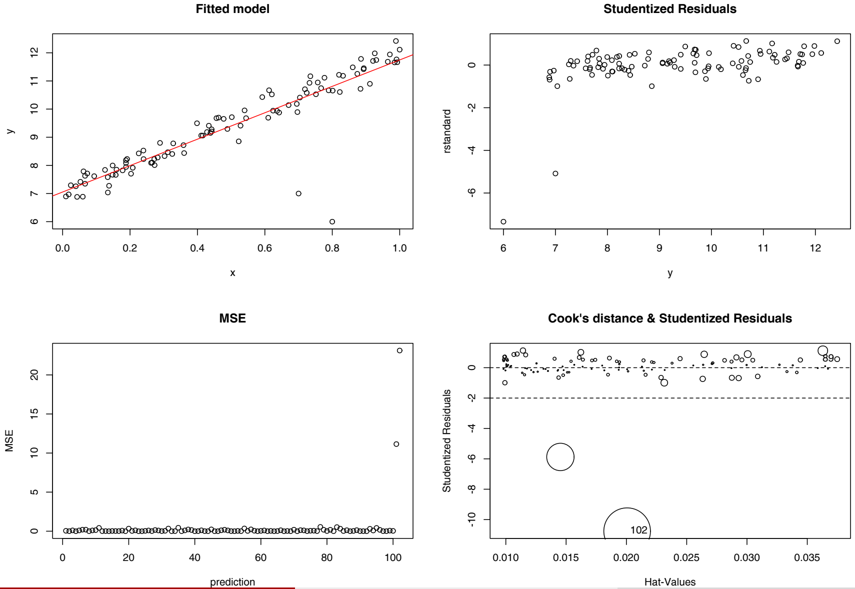
Linear regression - Outlier

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



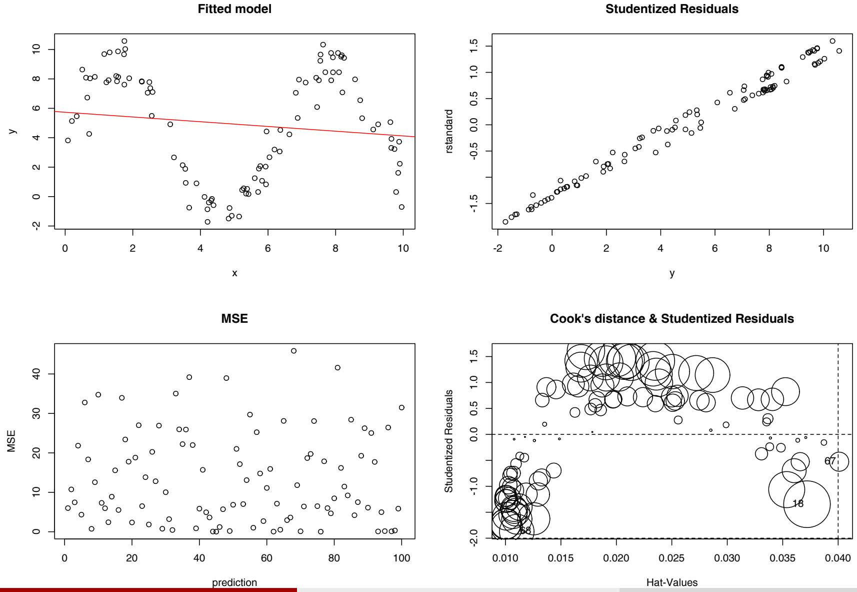
Non-Linear regression

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

$$y_i = 5\sin(x_i) + 4 + \epsilon, \quad x_i \sim \mathcal{U}(0, 10), \quad \epsilon \sim \mathcal{N}(0, \sigma^2 = 1)$$

```
x <- runif(100, min= 0, max=10)
y <- 5*sin(x) + 4 + rnorm(100)
plot(x, y, main="Fitted model")
fit.nonlinear <- lm(y~x)
summary(fit.nonlinear)
abline(fit.nonlinear$coeff[1], fit.nonlinear$coeff[2], col="red")
plot(y, rstandard(fit.nonlinear), ylab='rstandard', main="Studentized-Residu.")
plot((y-fitted(fit.nonlinear))^2, ylab='MSE', xlab="prediction", main="MSE")
influencePlot(fit.nonlinear, main="Cook's distance & Studentized Residuals")
```

Exploring training data set



Overview

Poly-Orthogonal vs Poly-NonOrthogonal

- Polynomial Orthogonal

$$y = \beta_0 + P_1(x)\beta_1 + P_2(x)\beta_2 + \cdots + P_p(x)\beta_{p+1}$$

  where: $\forall n \neq m \; P_m(x) \perp P_n(x)$, or also, $cor(P_m(x), P_n(x)) \approx 0$.

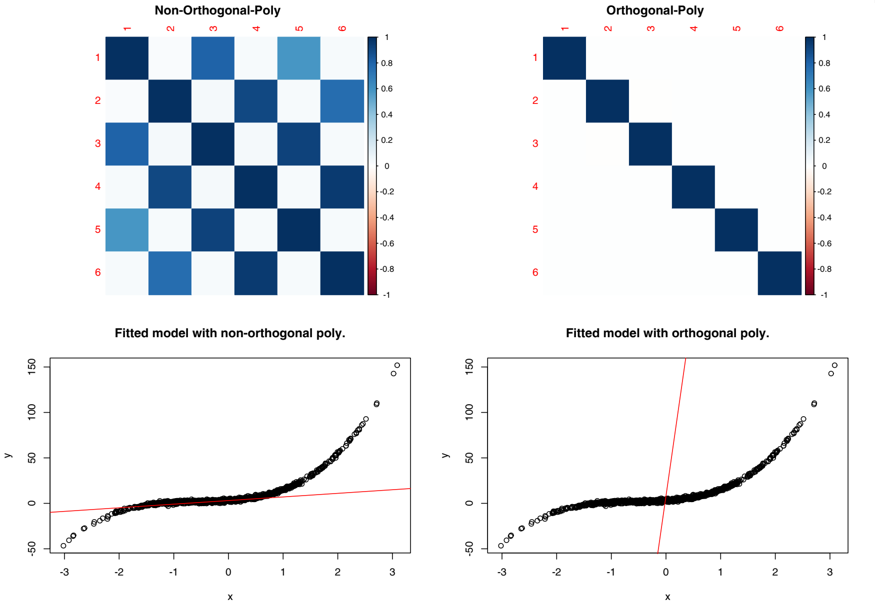
  1. Gram–Schmidt
  2. Legendre polynomials
  3. Hermite polynomials
  4. ....

- Polynomial Non-Orthogonal

$$y = \beta_0 + x\beta_1 + x^2\beta_2 + \cdots + x^p\beta_{p+1}$$

```r
library(corrplot)
# differences con option raw
x <- rnorm(1000)
raw.poly <- poly(x,6,raw=T)
orthogonal.poly = poly(x,6)
corrplot(cor(raw.poly),method="color",title="Non-Orthogonal-Poly")
corrplot(cor(orthogonal.poly),method="color",title="Orthogonal-Poly")
```

Correlation of two models (orthogonal or not)



Fitted a simple linear model

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

$$y_i = 3 + 4 * x + 5 * x^2 + 3 * x^3 + \epsilon, \quad x_i \sim \mathcal{N}(0, 1), \quad \epsilon \sim \mathcal{N}(0, \sigma^2 = 1)$$

```
# fitted value (two cases)
y <- 3 + 4*x + 5*x^2 + 3*x^3 + rnorm(1000)
raw.mod <- lm(y~poly(x,6,raw=T))
orthogonal.mod <- lm(y~poly(x,6))
plot(x, y, main="Fitted model with non-orthogonal poly.")
abline(raw.mod$coefficients[1], raw.mod$coefficients[2], col="red")
plot(x, y, main="Fitted model with orthogonal poly.")
abline(orthogonal.mod$coefficients[1],orthogonal.mod$coefficients[2])
sum(fitted(raw.mod)-fitted(orthogonal.mod))

# -1.831868e-14 (almost 0)
```

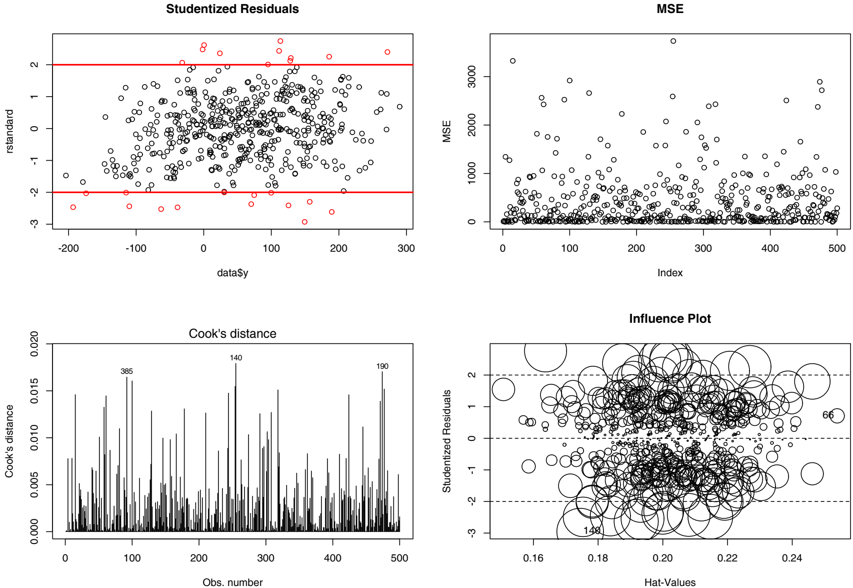
☞ Predictions are equal for the two simple linear models (OLS)!!. Why?.
☞ The Ridge and LASSO methods have a different behavior with each one (Orthogonal and Non-Orthogonal)

Overview

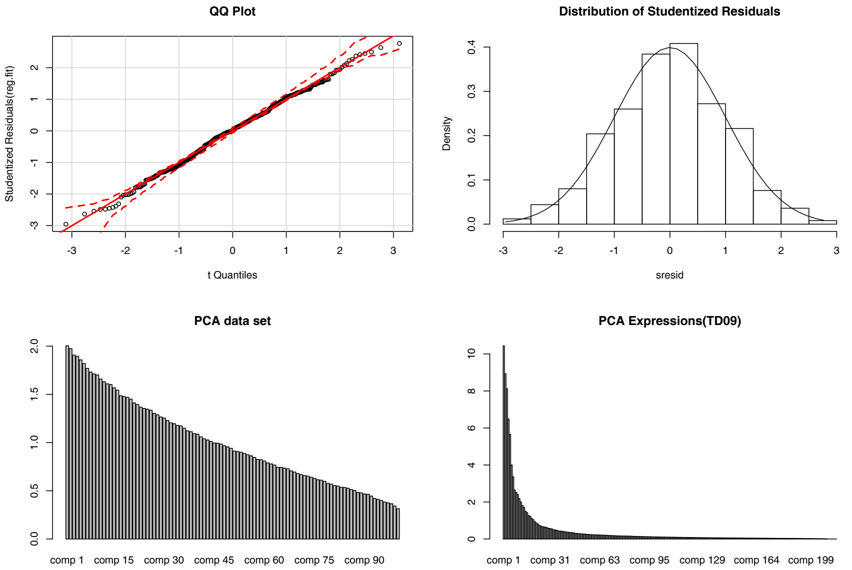
First view of the simple fitted regression model

```r
library(car)
library(MASS)
# Regression simple
reg.fit <- lm(y~., data=data)
summary(reg.fit)
plot(data$y,rstandard(reg.fit),ylab='rstandard',main="StudentizedResiduals",
     col=ifelse(abs(rstandard(reg.fit))> 2, 'red', 'black'))
abline(h = -2, col="red", lwd=2)
abline(h = 2, col="red", lwd=2)
plot((data$y - fitted(reg.fit))^2, ylab='MSE', main="MSE")

# Cook's Distance plot (identify D values > 4/(n-k-1))
cutoff <- 4/((nrow(data)-length(reg.fit$coefficients)-2))
plot(reg.fit, which=4, cook.levels=cutoff)

# Cook's Distance and Studentized Residuals
influencePlot(reg.fit, main="Influence Plot",
              sub="Circle size is proportial to Cook's Distance")

# Analysis of residuals information
qqPlot(reg.fit, main="QQ Plot") # qq plot for studentized residuals
sresid <- studres(reg.fit)
hist(sresid, freq=FALSE, main="Distribution of Studentized Residuals")
xfit <- seq(min(sresid), max(sresid), length=40)
yfit <- dnorm(xfit)
lines(xfit, yfit)
```

Exploring training data set



Exploring training data set



Overview

Set of different transformations

Given $\boldsymbol{y_i}, \beta_0 \in \mathbb{R}$ and $\boldsymbol{x}, \beta_* \in \mathbb{R}^p$

$$\boldsymbol{y} = \beta_0 + \boldsymbol{x_i}\beta_1 \quad \text{(Base model)}$$

$$\boldsymbol{y} = \beta_0 + \boldsymbol{x_i}\beta_1 + \boldsymbol{x_i}^2\beta_2 \quad \text{(Quadratic model)}$$

$$\boldsymbol{y} = \beta_0 + \boldsymbol{x_i}\beta_1 + \cdots + \boldsymbol{x_i}^k\beta_k \quad \text{(Polynomial model of degree k)}$$

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

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

$$\boldsymbol{y} = \beta_0 + \boldsymbol{x_i}\beta_1 + \ln(\boldsymbol{x_i})\beta_2 + \exp(\boldsymbol{x_i})\beta_3 + \cdots + \boldsymbol{x_i}^k\beta_{k+2} \quad \text{(Mixed model)}$$

$$\boldsymbol{y} = \beta_0 + \boldsymbol{x_i}\ln(\boldsymbol{x_i})\beta_1 + \cdots + \exp(\boldsymbol{x_i})\boldsymbol{x_i}^k\beta_k \quad \text{(Crazy model)}$$

$$\ldots \quad \text{(Infinity Combinations)}$$

Does the size of the parameter β influence on the MSE?

Biais versus Variance (Regression model)

MSE and Bias-Variance decomposition:

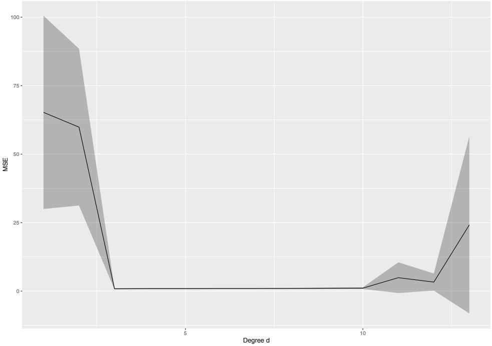
$$\mathbb{E}\left[(y-\widehat{f}_k)^2 \,\middle|\, X=x\right] = \underbrace{\underbrace{\mathbb{V}ar[\widehat{f}_k|X=x]}_{\text{Variance of } \widehat{f}_k} + \left(\underbrace{\mathbb{E}\left[\widehat{f}_k\right]-f}_{\text{Bias of } \widehat{f}_k}\right)^2}_{\text{Erreur réductible}} + \underbrace{\sigma^2}_{\text{Irreductible Error}}$$

$$= \text{Bias}(\widehat{f}_k)^2 + \mathbb{V}ar(\widehat{f}_k) + \sigma_\epsilon^2$$

Given our following model with $\beta = (\beta_0, \beta_1, \beta_2, \ldots, \beta_m) \in \mathbb{R}^{m+1}$ and functions $\phi = (\phi_1, \ldots, \phi_m)$,$\forall i, \phi_i : \mathcal{X} \to \mathbb{R}$

$$y_i = \beta_0 + \phi_1(x_{1,i})\beta_1 + \phi_2(x_{2,i})\beta_2 + \cdots + \phi_m(x_{m,i})\beta_m \qquad \text{(Model)}$$

Example (Polynomial linear model of degree d)



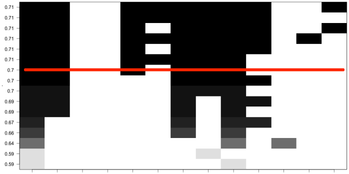
$$y_i = \beta_0 + \phi_1(x_{1,i})\beta_1 + \phi_2(x_{2,i})\beta_2 + \cdots + \phi_m(x_{m,i})\beta_m \qquad \text{(Model)}$$

☞ If $m \to \infty$ (i.e. bigger) $\implies$ $\text{Bias}(\widehat{f}_k)^2 \to 0$ and $\mathbb{V}ar(\widehat{f}_k) \to \infty$

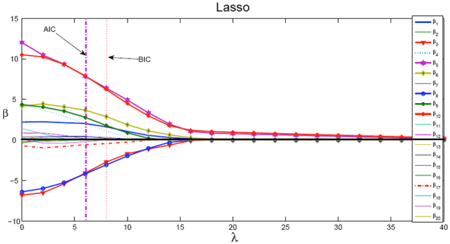
☞ If $m \to 0$ (i.e. smaller) $\implies$ $\text{Bias}(\widehat{f}_k)^2 \to \infty$ and $\mathbb{V}ar(\widehat{f}_k) \to 0$

RegSubset versus LASSO variable selection

1. Forward and Backward stepwise selection



2. LASSO regression method

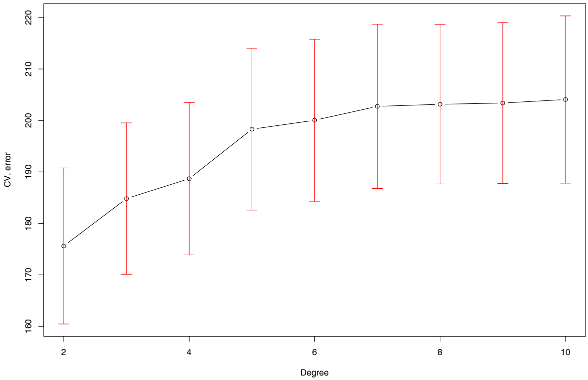


Overview

Non-orthogonal polynomial regression (with LASSO)

We start with a non-orthogonal polynomial regression model:

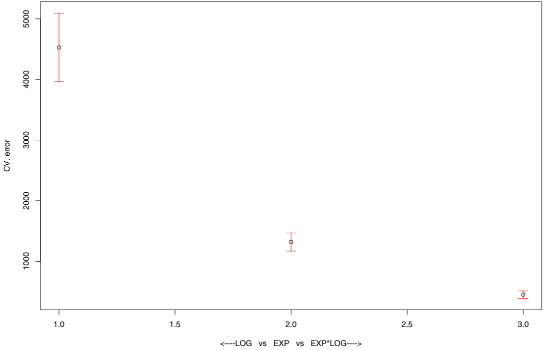
$$y_i = \beta_0 + \boldsymbol{x_i}\beta_1 + \cdots + \boldsymbol{x_i}^k\beta_k$$



☞ Best degree to minimise MSE: $k = 2$

Logarithm + Exponential + ... regression (with LASSO)

$$y_i = \beta_0 + \boldsymbol{x_i}\beta_1 + \ln(\boldsymbol{x_i})\beta_2 \qquad \text{(Log. model)}$$
$$y_i = \beta_0 + \boldsymbol{x_i}\beta_1 + \exp(\boldsymbol{x_i})\beta_1 \qquad \text{(Exp. model)}$$
$$y_i = \beta_0 + \boldsymbol{x_i}\beta_1 + \exp(\boldsymbol{x_i}) * \ln(\boldsymbol{x_i})\beta_2 \qquad \text{(Exp*Log model)}$$



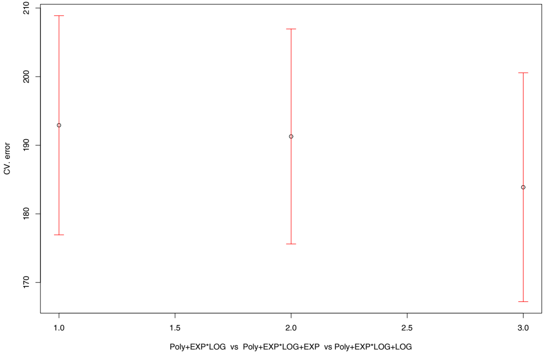
☞ Best model to minimse MSE: $\exp * \log$

Exp*log + Polynomial + others regression (with LASSO)

$$y_i = \beta_0 + \boldsymbol{x_i}\beta_1 + \boldsymbol{x_i}^2\beta_2 + \exp(\boldsymbol{x_i}) * \ln(\boldsymbol{x_i})\beta_3 \qquad \text{(Poly+Exp*Log)}$$

$$y_i = \beta_0 + \boldsymbol{x_i}\beta_1 + \boldsymbol{x_i}^2\beta_2 + \exp(\boldsymbol{x_i}) * \ln(\boldsymbol{x_i})\beta_3 + \exp(\boldsymbol{x_i})\beta_4$$

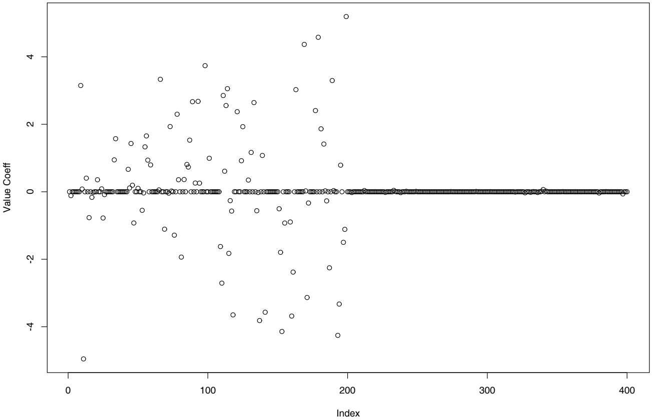
$$y_i = \beta_0 + \boldsymbol{x_i}\beta_1 + \boldsymbol{x_i}^2\beta_2 + \exp(\boldsymbol{x_i}) * \ln(\boldsymbol{x_i})\beta_3 + \log(\boldsymbol{x_i})\beta_4$$



☞ It is necessary to check all combinations?

Analysis coefficients of the best model (fitted all dataset)

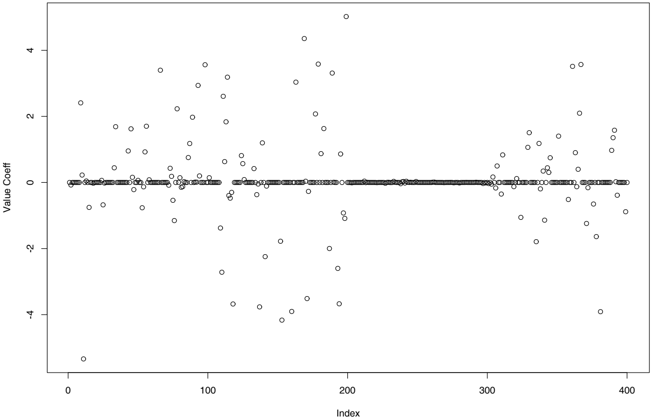
$$y_i = \beta_0 + \boldsymbol{x_i}\beta_1 + \boldsymbol{x_i}^2\beta_2 + \exp(\boldsymbol{x_i}) * \ln(\boldsymbol{x_i})\beta_3 + \exp(\boldsymbol{x_i})\beta_4$$



☞ Last 200 parameters are almost equal to zero $\Longrightarrow$ $\exp(\boldsymbol{x_i}) * \ln(\boldsymbol{x_i})\beta_3$ and $\exp(\boldsymbol{x_i})\beta_4$ do not contribute in the model.

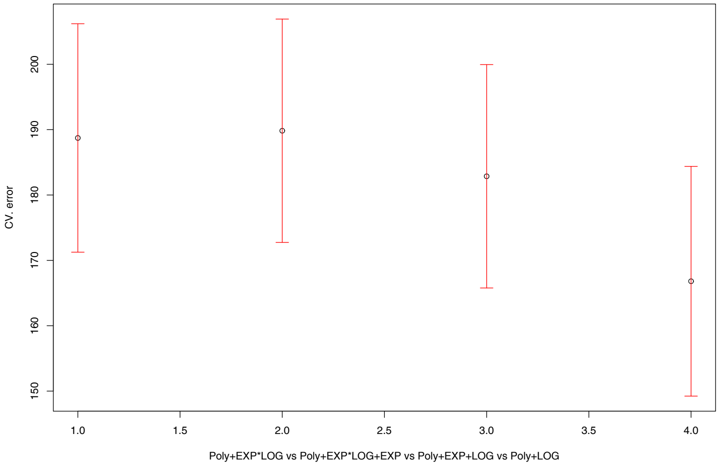
Analysis coefficients of the best model (fitted all dataset)

$$y_i = \beta_0 + \boldsymbol{x_i}\beta_1 + \boldsymbol{x_i}^2\beta_2 + \exp(\boldsymbol{x_i}) * \ln(\boldsymbol{x_i})\beta_3 + \log(\boldsymbol{x_i})\beta_4$$



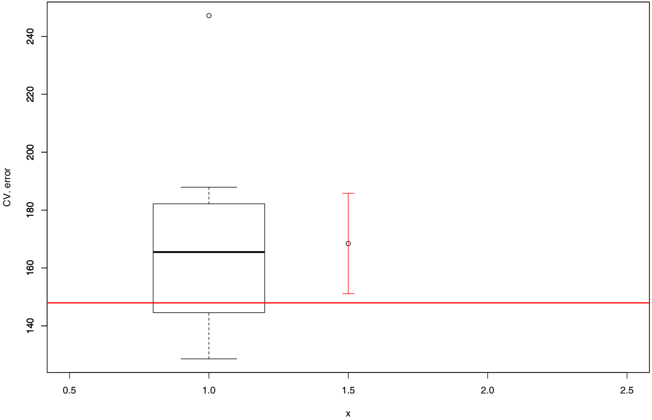
☞ 200-300 parameters are almost equal to zero $\Longrightarrow$ $\exp(\boldsymbol{x_i}) * \ln(\boldsymbol{x_i})\beta_3$ does not contribute in the model.

Summary all models



☞ My model held is Polynomial + Logarithm regression model.

Testing Error MSE



☞ Error testing: 147.952.

Boxplot is not CONFIDENCE INTERVAL

Thanks