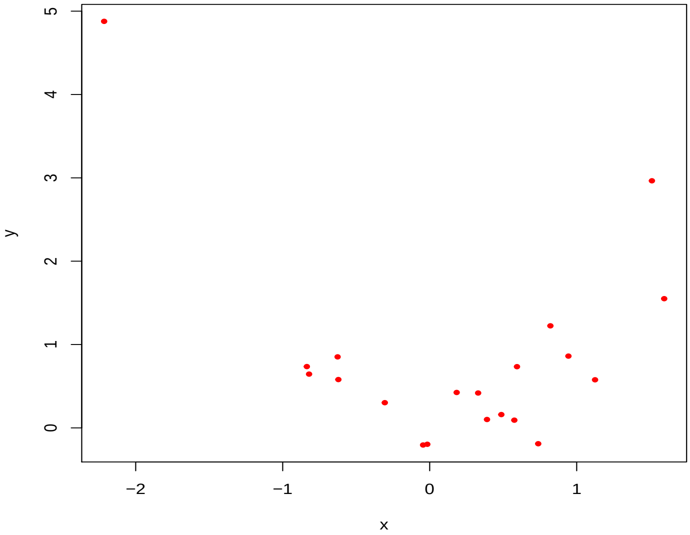
Ch1. An instructive example

ST4240, 2016/2017

Version 0.1

Alexandre Thiéry

Department of Statistics and Applied Probability


5
4
3
2
1
0
y
−2
−1
0
1
x

- Data $y = (y_1, \ldots, y_n)$ are noisy observations of a function f,

$$y_i = f(x_i) + \varepsilon.$$

- For simplicity, we suppose that $f(x)$ is a polynomial

$$f(x) = \sum_{k=0}^{d} \beta_k^\star \, x^k.$$

- In other words, we observe $y = (y_1, \ldots, y_n)$ with

$$y_i = \sum_{k=0}^{d} \beta_k^\star \, x_i^k + \varepsilon_i$$

- Our task is to reconstruct $\beta^\star = (\beta_0^\star, \ldots, \beta_d^\star)$.
- Note that d is not known.

- **[Exercise]** the linear model can be written as

$$y = X\,\beta^\star + \varepsilon$$

with $y \in \mathbb{R}^n$ and $X \in \mathbb{R}^{n\times(d+1)}$ and $\beta^\star \in \mathbb{R}^{d+1}$.

- The least square estimate $\widehat{\beta}$ is

$$\widehat{\beta} = \mathbf{argmin} \left\{ \beta \mapsto \|Y - X\,\beta\|^2 \right\}$$

- We will see later in the course that $\widehat{\beta}$ is given by

$$\widehat{\beta} = \left(X^\top X \right)^{-1} X^\top y.$$

- For a new value $x \in \mathbb{R}$, prediction is made through

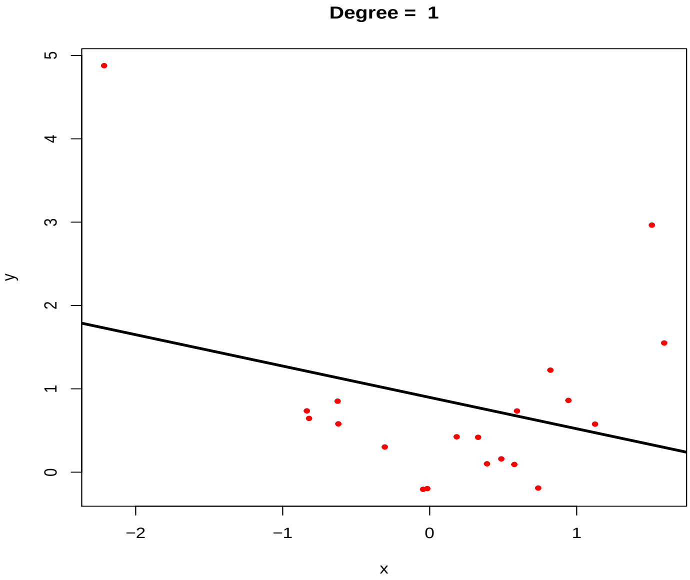
$$\sum_{k=0}^{d} \widehat{\beta}_k \, x^k$$

- Let us look at the predicted (or fitted) value on the data that have been used to construct $\widehat{\beta}$; we have $\widehat{y} = X\widehat{\beta}$, which also reads

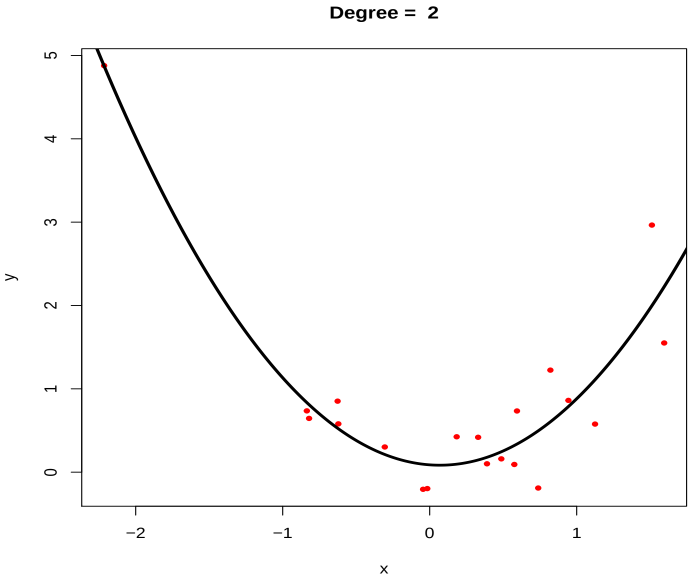
$$\widehat{y} = H \, y \qquad \text{with} \qquad H \equiv X \left(X^\top X \right)^{-1} X^\top,$$

- the matrix H is usually called the hat matrix.
- [**Exercise**] the matrix H is a projection: $H^2 = H$.
- [**Exercise**] the matrix H is such that $H^\top = H$.

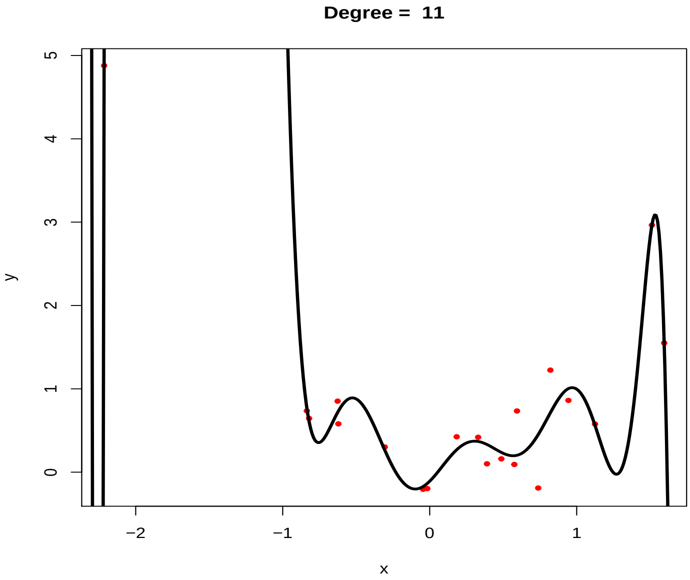
d too low



d just right



d too high



Measuring performances

- A common way of measuring performance is

$$(\text{performance}) = \sum_{i=1}^{n} \textsf{\textbf{Loss}}(y_i, \widehat{y}_i)$$

where the Loss function $\textsf{\textbf{Loss}}(\cdot)$ measures how well the prediction $\widehat{y}_i$ approximate the true value y_i.

- A common choice, because this leads to tractable computations, is the squared error loss function

$$\textsf{\textbf{Loss}}(y, \widehat{y}) \equiv (y - \widehat{y})^2.$$

- The resulting measure of performance is called the Residual Sum of Square,

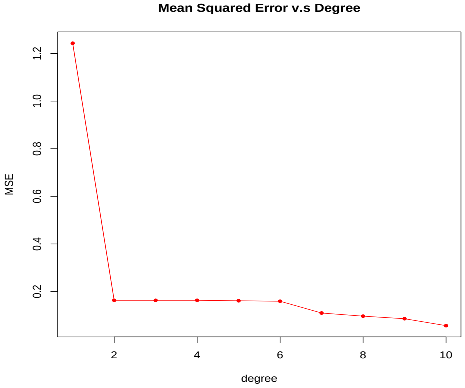
$$\textsf{\textbf{RSS}} = \sum_{i=1}^{N} (y_i - \widehat{y}_i)^2.$$

MSE as a function of d

- The Mean Squared Error

$$\textbf{MSE} = (1/n)\textbf{RSS}$$

is equivalent to the Residual Sum of Square.

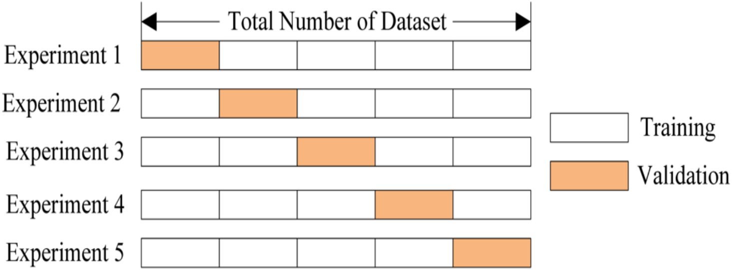


- [**Exercise**] the **MSE** decreases as d increases.
- In most situations of interest, we are trying to do some predictions on data that have not indeed been used to train the model. In the above situations, the coefficient $\widehat{\beta}$ has been determined by using the whole dataset $\{y_i\}_{i=1}^{n}$ and the **MSE** has been estimated on the same dataset!

Training and Validation sets

- One needs to test the procedure on data that have not been used to train the algorithm
- Split the whole dataset into a training set and a validation set.
- Train (i.e. find $\widehat{\beta}$) on the training set
- Estimate performances (i.e. evaluate the **MSE**) on the validation set.

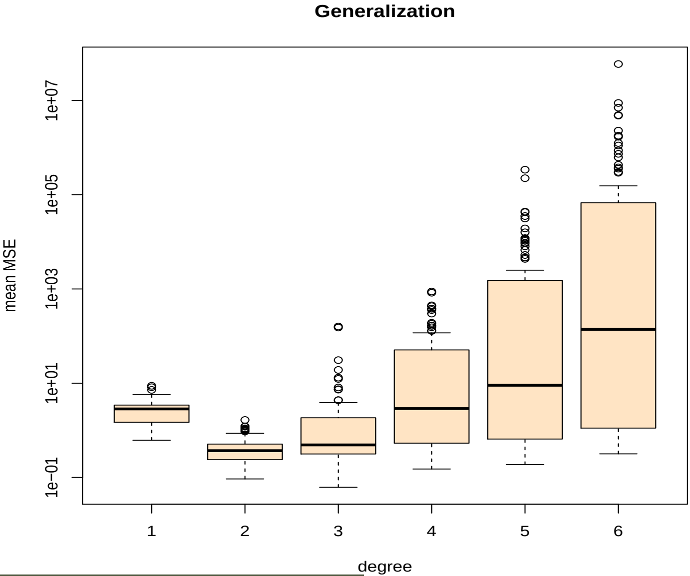
k-fold Cross Validation



Monte-Carlo Cross Validation

- Dataset of size N
- Randomly choose $p\%$ of the dataset as training set
- Use the remaining $(100 - p)\%$ as training set
- Iterate as many times as necessary

So how do we choose d?



Least Square v.s. Maximum Likelihood

- Consider the linear model $y = X\,\beta + \varepsilon$
- Assume that ε is Gaussianly distributed
- **[Exercise]** show that the least square estimate $\widehat{\beta}$ is also the maximum likelihood estimate.