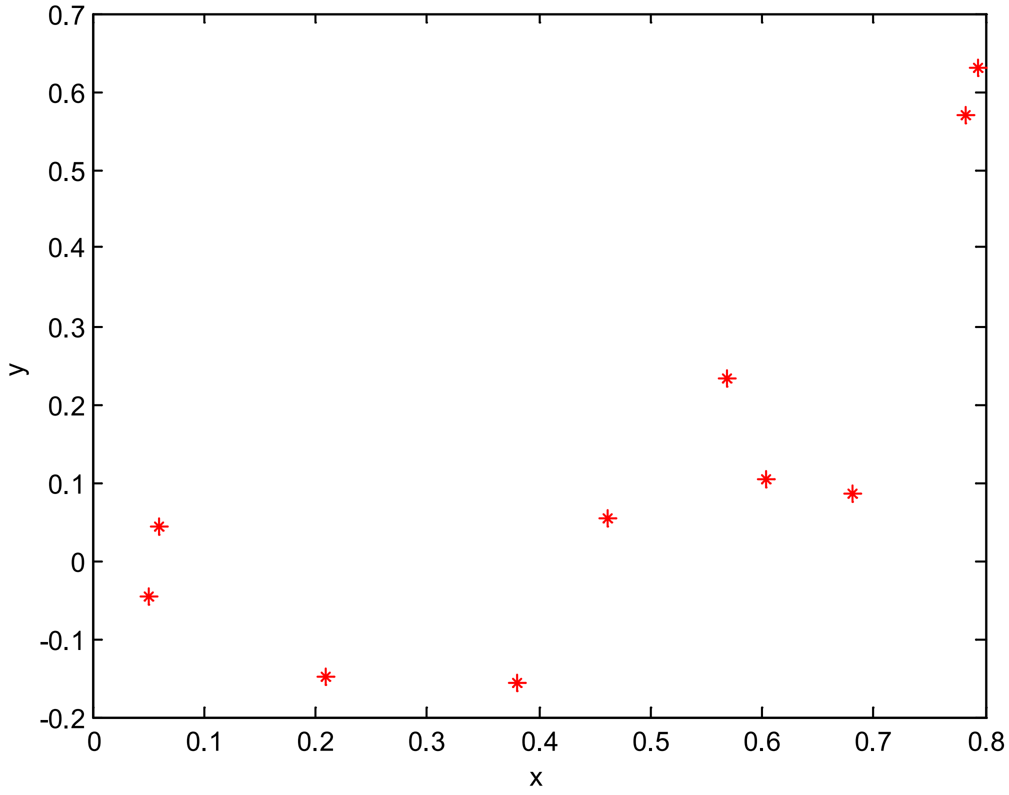
ELEC2870 - Machine learning: regression and dimensionality reduction

Model selection

Michel Verleysen
Machine Learning Group
Université catholique de Louvain
Louvain-la-Neuve, Belgium
michel.verleysen@uclouvain.be

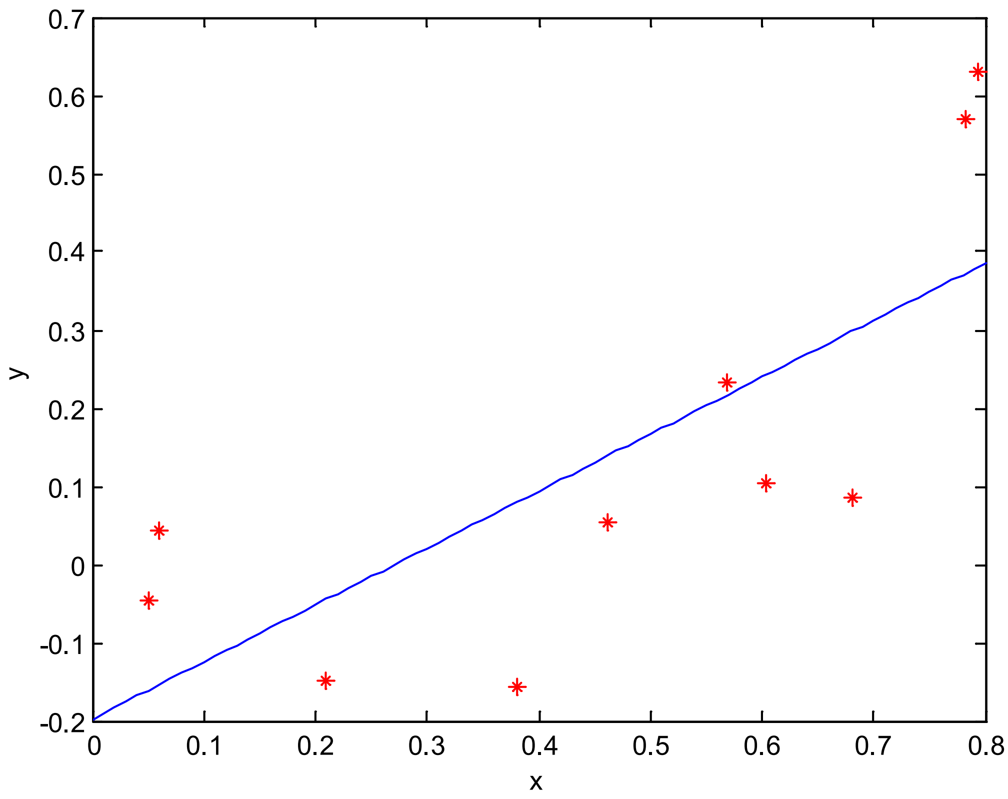
The question

- How to select a model between several possible ones ?



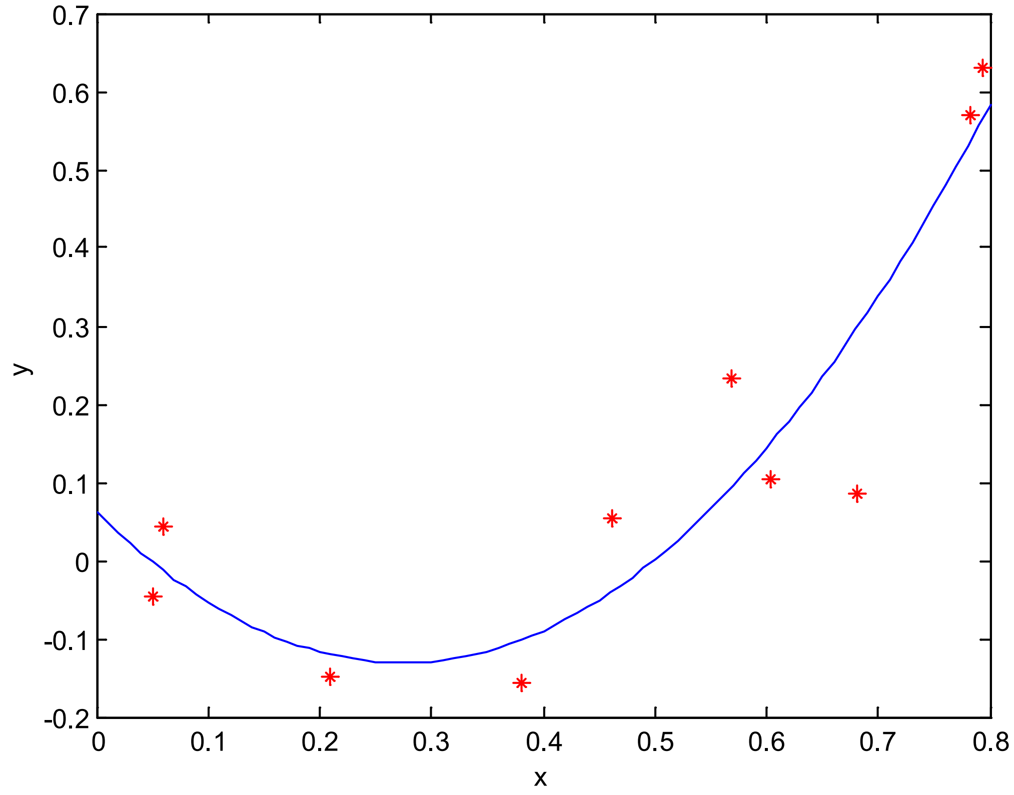
The question

- How to select a model between several possible ones ?
  - Linear model



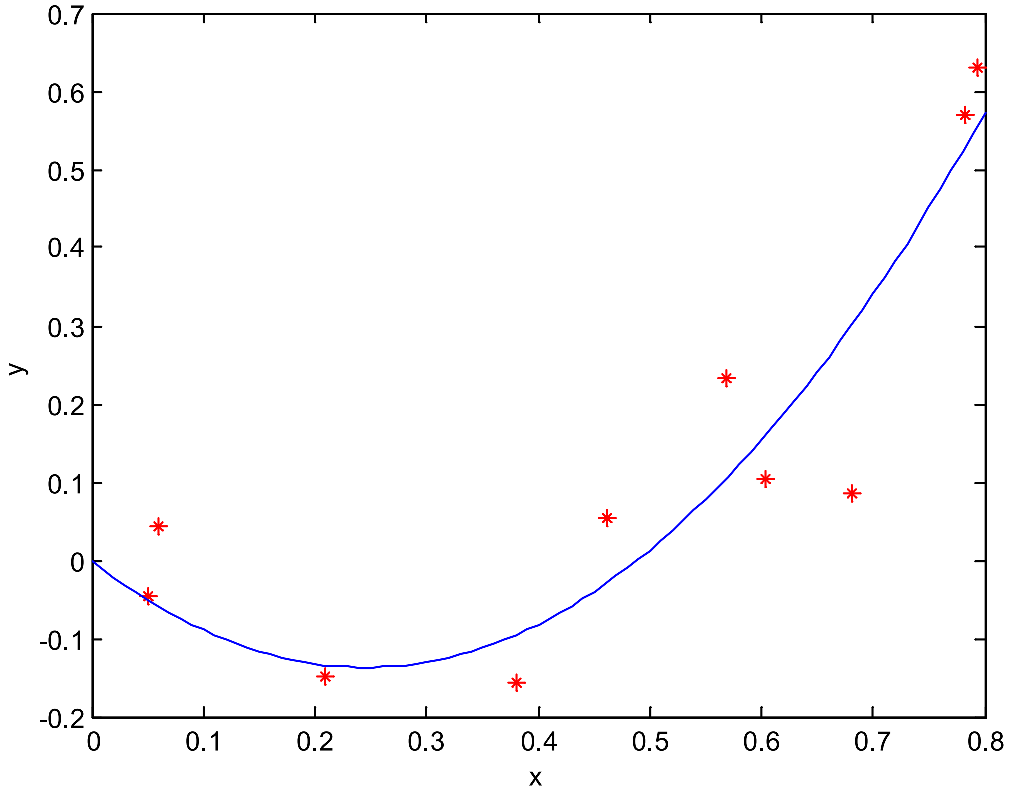
The question

- How to select a model between several possible ones ?
  - Quadratic model



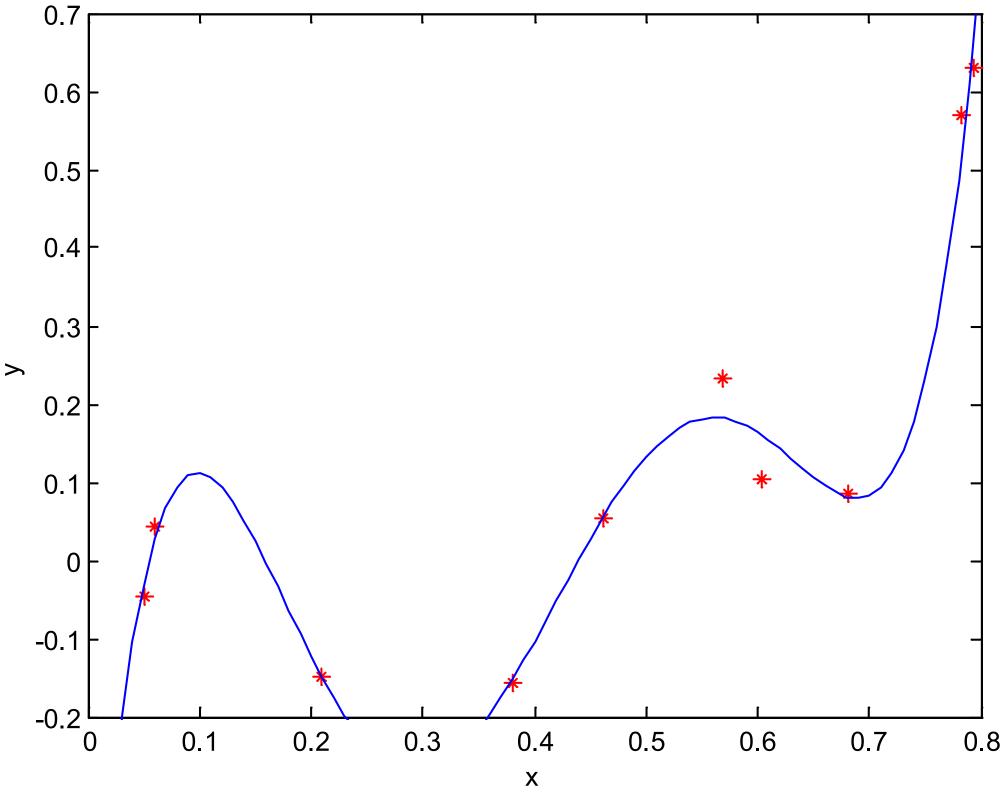
The question

- How to select a model between several possible ones ?
  - Quadratic model without independent term
    (the one used to generate the data here…)



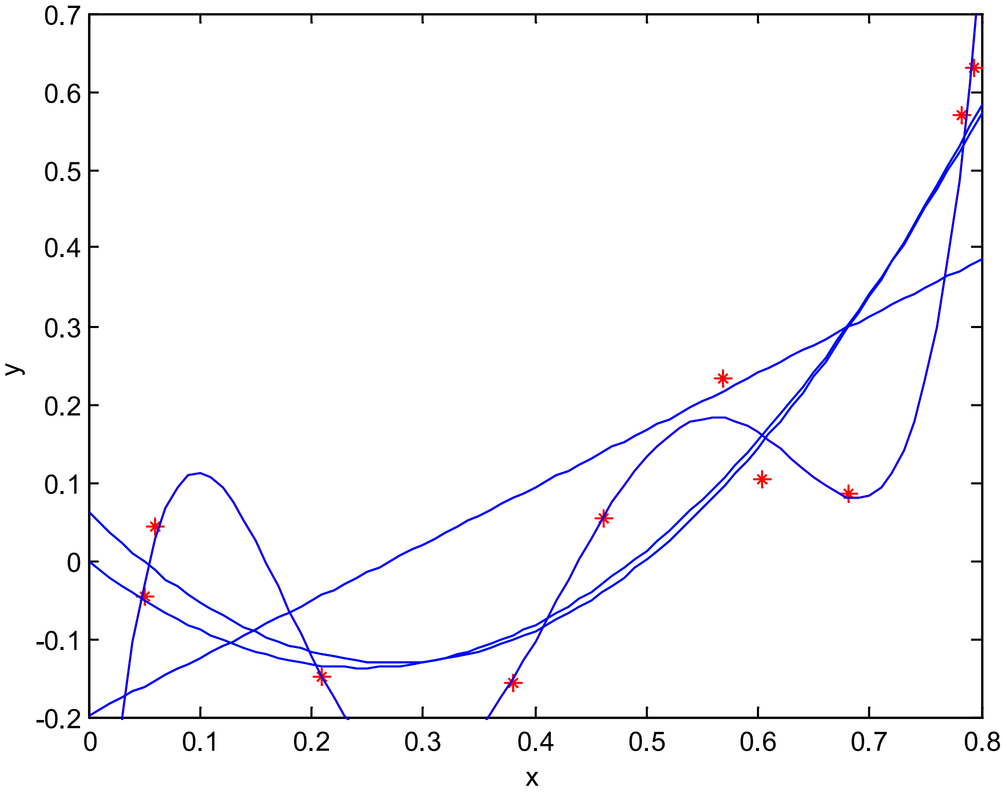
The question

- How to select a model between several possible ones ?
  - A 5^{th}-order polynomial



The question

- How to select a model between several possible ones ?



Parameters and hyperparameters

- Notations $x \in \mathcal{R}^d, y \in \mathcal{R}$
  $y = g(x, \theta)$

- Parameters: $\theta = \{a, b, c\} \longrightarrow$ learning

- Hyperparameters:
  $$g_{\alpha_1}(x, \theta) = g_{\alpha_1}(x, \{a, b\}) = a + bx$$
  $$g_{\alpha_2}(x, \theta) = g_{\alpha_2}(x, \{a, b, c\}) = a + bx + cx^2$$
  α is the hyperparameter

- Question: how to set α (= how to select the model) ?

Error criterion

- Model structure selection is performed according to an error criterion

- Ideally: error criterion on all possible new data that could be used when using the model (generalization):

$$E_{gen}(\theta) = \int_x \left(g(x,\theta) - y(x)\right)^2 dx = \lim_{N\to\infty} \sum_{i=1}^{N} \frac{(g(x_i,\theta) - y_i)^2}{N}$$

- In practice: we don't have $N \to \infty$. We estimate E_{gen} with

$$\hat{E}_{gen}(\theta) = \sum_{i=1}^{N} \frac{(g(x_i,\theta) - y_i)^2}{N}$$

  for a finite (sometimes small) value of N

Model structure selection

- Test methods
  - AIC + BIC
  - validation, cross-validation, k-fold and leave-one-out
  - bootstrap

- Model Selection
  - some theoretical insights
  - examples

- Test of classifiers

- Pruning
  - pruning during learning (regularization)
  - pruning after learning (direct pruning, local pruning, OBD, OBS)

AIC and BIC: the linear criteria

- Used in linear statistics, system identification and control, ...
- Principle:
  - the same set of data is used for learning and performance estimation (→ overfitting not detected)
  - instead the (estimation of) the generalization error is penalized with a term that depends on the complexity of the model (number of parameters)

- Akaike Information Criterion (AIC)

$$\hat{E}_{gen,AIC}(\theta) = \sum_{i=1}^{N} \frac{(g(x_i,\theta) - y_i)^2}{N} + \frac{2}{N}\dim(\theta)$$

- Bayesian Information Criterion / Minimum Description Length (BIC / MDL)

$$\hat{E}_{gen,BIC}(\theta) = \sum_{i=1}^{N} \frac{(g(x_i,\theta) - y_i)^2}{N} + \frac{\ln N}{N}\dim(\theta)$$

AIC and BIC: the linear criteria

$$\hat{E}_{gen,AIC}(\theta) = \sum_{i=1}^{N} \frac{(g(x_i,\theta) - y_i)^2}{N} + \frac{2}{N}\dim(\theta)$$

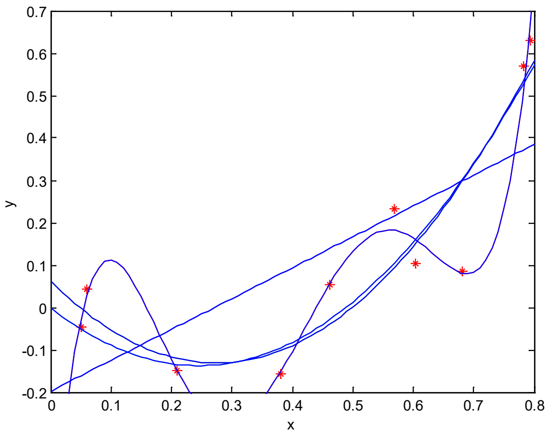
$$\hat{E}_{gen,BIC}(\theta) = \sum_{i=1}^{N} \frac{(g(x_i,\theta) - y_i)^2}{N} + \frac{\ln N}{N}\dim(\theta)$$

- Both are very good if:
  - the model is linear
  - N is very large
- In practice:
  - non-linear models
  - N is small

Both AIC and BIC lead here to overfitting

$$\hat{E}_{gen,AIC}(\theta) = \sum_{i=1}^{N} \frac{(g(x_i,\theta) - y_i)^2}{N} + \frac{2}{N}\dim(\theta)$$

$$\hat{E}_{gen,BIC}(\theta) = \sum_{i=1}^{N} \frac{(g(x_i,\theta) - y_i)^2}{N} + \frac{\ln N}{N}\dim(\theta)$$

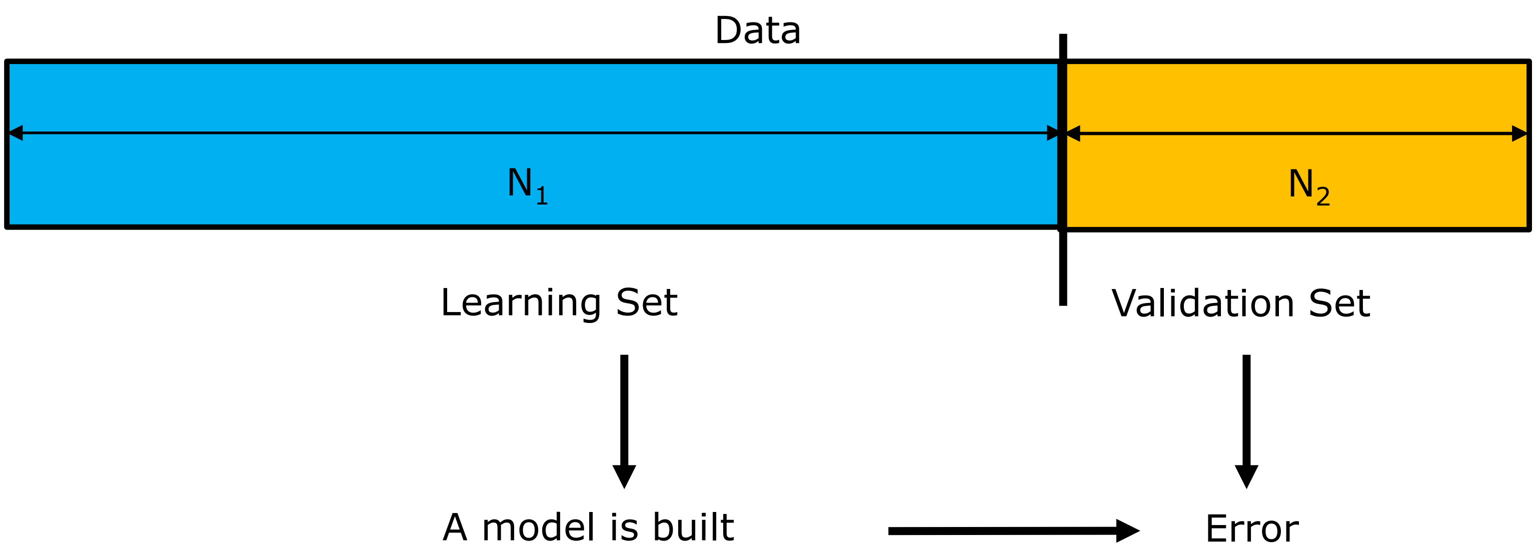


	linear model	quadratic model	quadratic model without indep. term	5th-order model
AIC	0.1752	0.0858	0.0570	0.0319
BIC	0.1973	0.0974	0.0641	0.0366

This model would be chosen in both cases

Validation

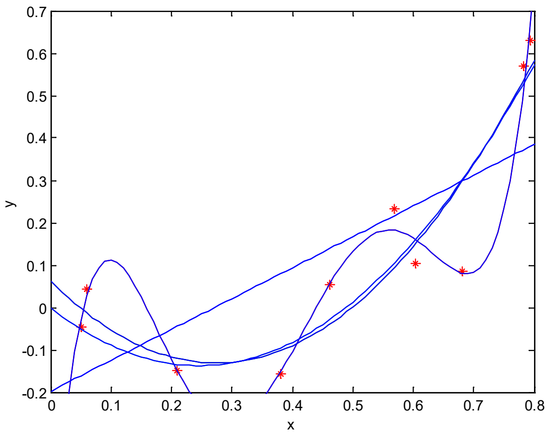
- Validation is the building block for further methods



$$\hat{E}_{gen}(\theta) = \sum_{x_i \in VS} \frac{(g(x_i,\theta) - y_i)^2}{N_2}$$

- Validation overfits here too

$$\hat{E}_{gen}(\theta) = \sum_{x_i \in VS} \frac{(g(x_i, \theta) - y_i)^2}{N_2}$$

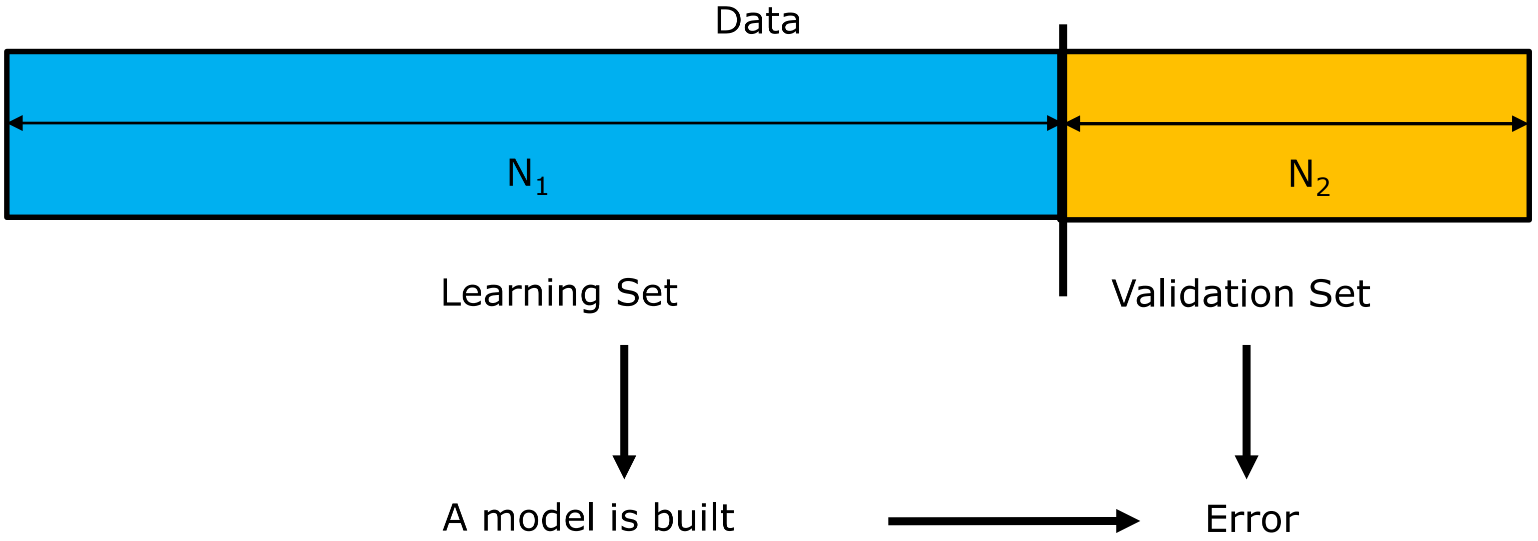


	linear model	quadratic model	quadratic model without indep. term	5^{th}-order model
validation	0.0370	0.0181	0.0118	$\boxed{0.0038}$

This model would be chosen

Cross-validation

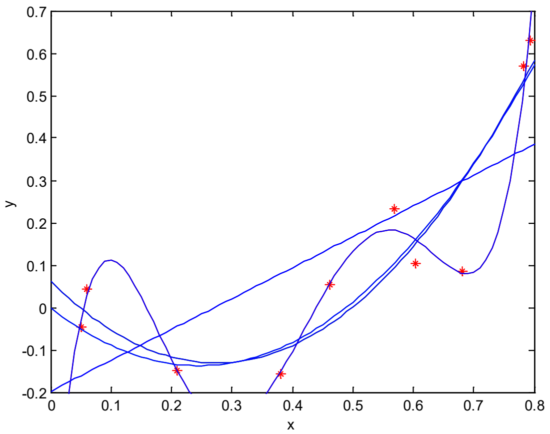
- K random repetitions of the validation, with different learning and validation sets



$$\hat{E}_{gen}(\theta) = \frac{1}{K}\sum_{k=1}^{K}\sum_{x_i \in VS}\frac{(g(x_i,\theta)-y_i)^2}{N_2}$$

Cross-validation

- Cross-validation selects one of the quadratic models



$$\hat{E}_{gen}(\theta) = \frac{1}{K}\sum_{k=1}^{K}\sum_{x_i \in VS}\frac{(g(x_i,\theta)-y_i)^2}{N_2}$$

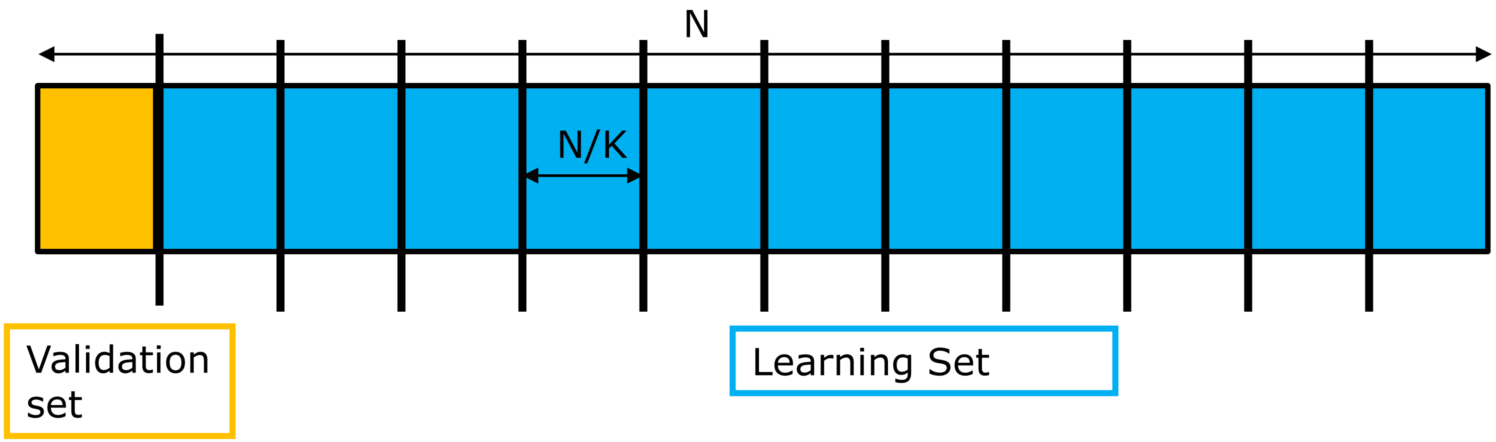
K	linear model	quadratic model	quadratic model without indep. term	5th-order model
10	0.0184	0.0229	0.0201	0.0097
100	0.0652	0.0275	0.0251	13.8062
1000	0.0743	0.0276	0.0257	154.8485
10000	0.0798	0.0267	0.0250	208.5566

This model would be chosen

The right model is selected

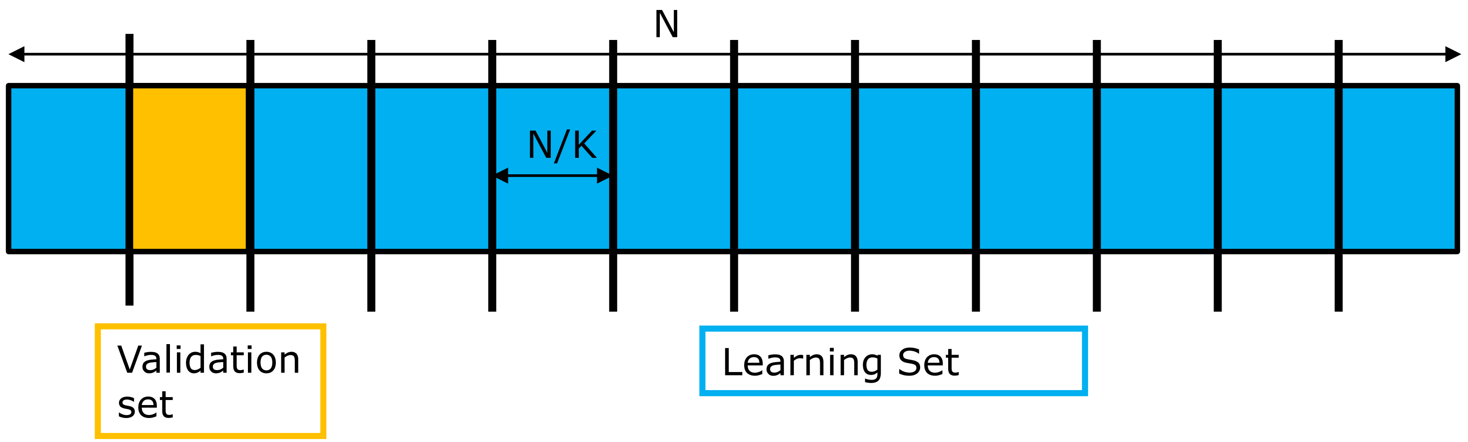
K-fold cross-validation

- Makes sure that each data is used once and only once for validation
- First step:



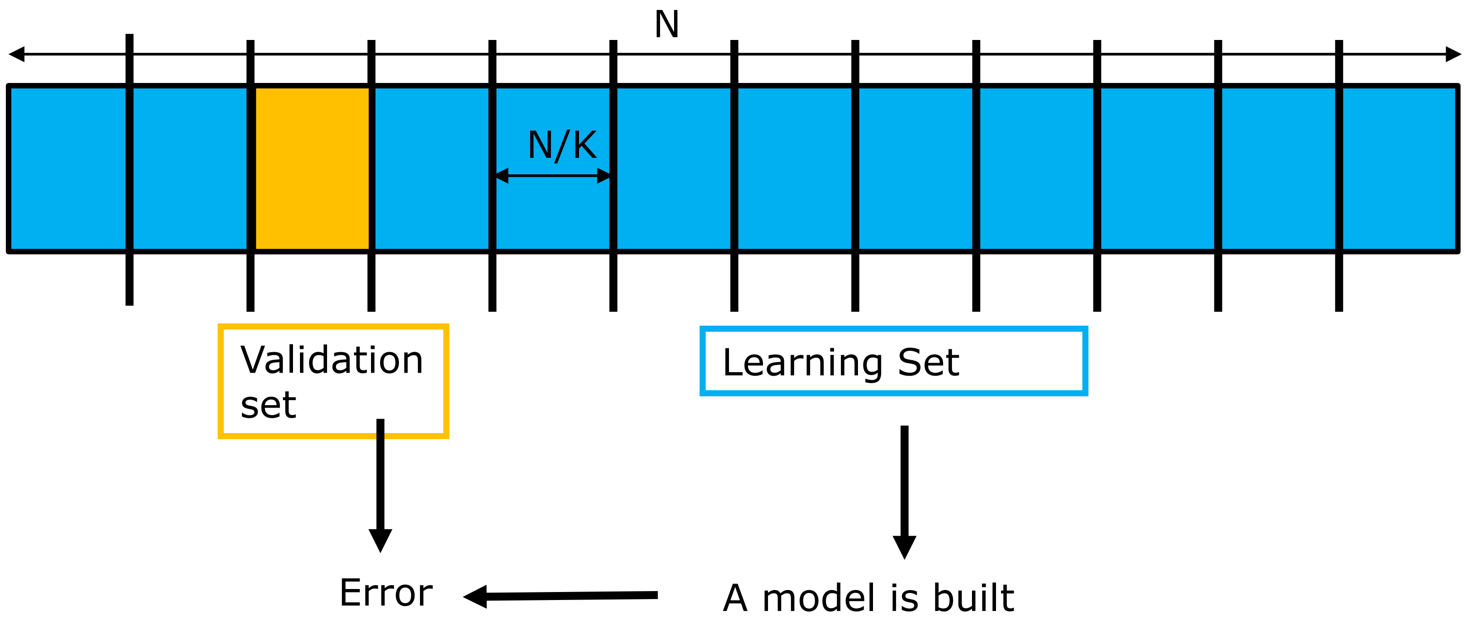
K-fold cross-validation

- Makes sure that each data is used once and only once for validation
- Second step:



K-fold cross-validation

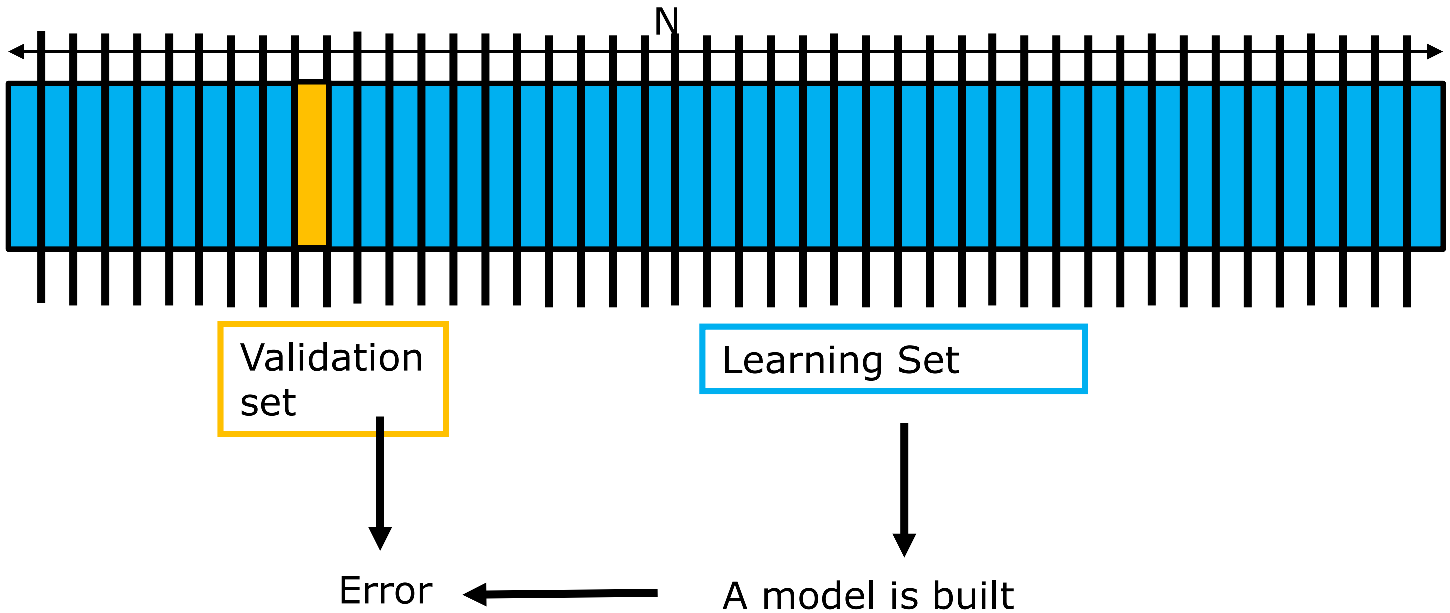
- Makes sure that each data is used once and only once for validation
- Third step, and so on…:



$$\hat{E}_{gen}(\theta) = \frac{1}{K}\sum_{k=1}^{K}\sum_{x_i \in VS}\frac{(g(x_i,\theta)-y_i)^2}{N/K}$$

Leave-one-out

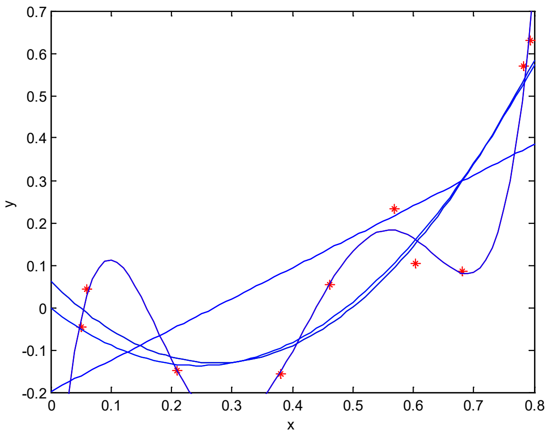
- K-fold cross-validation with K=N



$$\hat{E}_{gen}(\theta) = \frac{1}{N}\sum_{k=1}^{N}\sum_{x_i \in VS}\frac{(g(x_i,\theta)-y_i)^2}{N}$$

Leave-one-out

- Leave-one-out here leads to overfitting too



$$\hat{E}_{gen}(\theta) = \frac{1}{N}\sum_{k=1}^{N}\sum_{x_i \in VS}\frac{(g(x_i,\theta)-y_i)^2}{N}$$

	linear model	quadratic model	quadratic model without indep. term	5^{th}-order model
LOO	0.0488	0.0153	0.0146	$\boxed{0.0045}$

This model would be chosen

- Principle :
  - We would like to test our model on all possible data (the *world*)
  - We can't, so we have to use what is available (the *sample*)
  - But using the sample for both learning and evaluation introduces an optimism in the evaluation: the overfitting is not detected

world - sample = optimism

Bootstrap

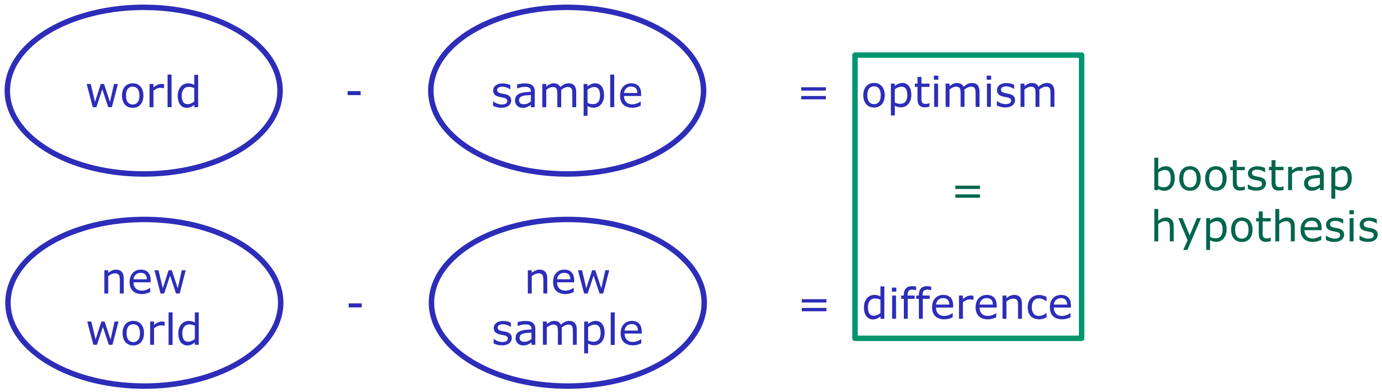
- Principle :
  - We would like to test our model on all possible data (the *world*)
  - We can't, so we have to use what is available (the *sample*)
  - But using the sample for both learning and evaluation introduces an optimism in the evaluation: the overfitting is not detected

world - sample = optimism

- Idea :
  - We build a new world, and a new sample

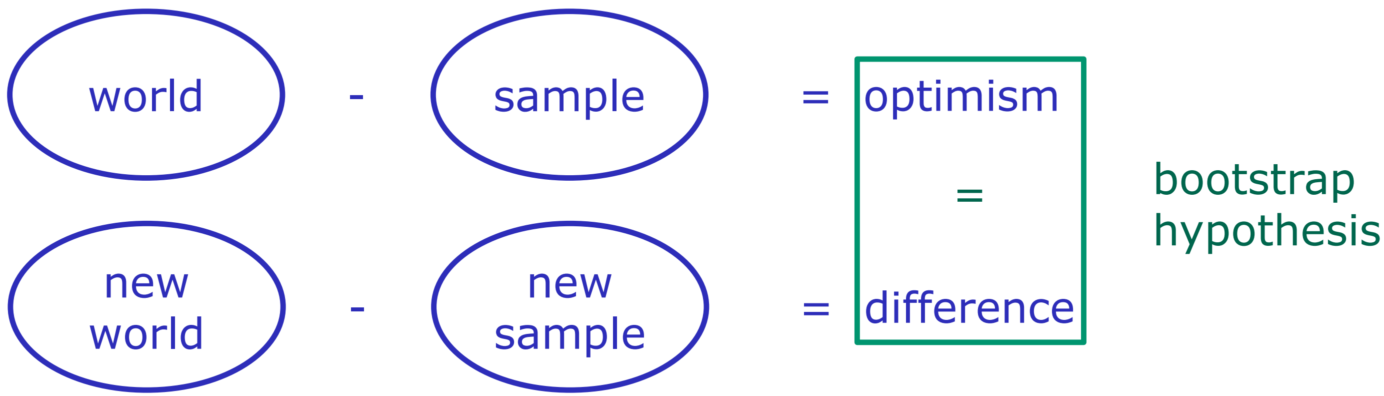
new world - new sample = difference

Bootstrap: plug-in principle



- Bootstrap hypothesis: if the new world and the new sample are well chosen, the optimism and the difference will be approximately identical

Bootstrap: plug-in principle



- Bootstrap hypothesis: if the new world and the new sample are well chosen, the optimism and the difference will be approximately identical

- New world: the sample (all available data)
- New sample: drawn with replacement from sample (same size)

Bootstrap: drawing with replacement

- Drawing with replacement

New world = sample		New sample
1		4
2		6
3		3
4		7
5	Same size	3
6		10
7		5
8		3
9		9
10		10

Boostrap: Estimation of $\hat{E}_{gen}$

- Notations:
  - $E_{A,B}$ is the error of a model learned on A and tested on B
  - Therefore $E_{gen} = E_{sample,world}$
  - W=world, S=sample, W^*=new world, S^* =new sample

- We want $E_{S,W}$ - $E_{S,S}$ = optimism

  =

  We have $E_{S^*,W^*} = E_{S^*,S}$ - E_{S^*,S^*} = difference

$$
\begin{aligned}
\hat{E}_{gen} &= E_{S,W} + E_{S,S} - E_{S,S} \\
&= E_{S,S} + \left(E_{S,W} - E_{S,S}\right) \\
&= E_{S,S} + \left(E_{S^*,S} - E_{S^*,S^*}\right) \\
&= E_{S,S} + optimism
\end{aligned}
$$

bootstrap : plug-in principle

Bootstrap: estimation of the optimism

- In practice the optimism must be estimated
- Random sample → repeat the estimation with different S^*

$$optimism = \frac{1}{K}\sum_{k=1}^{K}\left(E_{S^{*k},S} - E_{S^{*k},S^{*k}}\right)$$

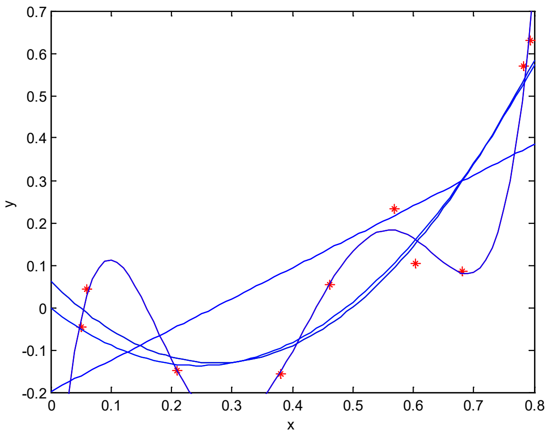
- The generalization error is thus
  the apparent error plus a bias correction

$$\hat{E}_{gen} = E_{S,S} + \frac{1}{K}\sum_{k=1}^{K}\left(E_{S^{*k},S} - E_{S^{*k},S^{*k}}\right)$$

- Necessitates to build $K+1$ models

Bootstrap

- Bootstrap does find the true model (observe the number of replications)



K	linear model	quadratic model	quadratic model without indep. term	5^{th}-order model
10	0.0384	0.0209	0.0155	8.3898
100	0.0345	0.0301	0.0128	703.74
1000	0.0325	0.0807	0.0112	7846.5
10000	0.0333	0.1267	0.0118	6422.5

The right model is selected

- Bootstrap 632

  – For each bootstrap replication compute the complementary set of the bootstrap sample and compute the replication optimism on these data (and <u>only</u> on them !)

$$\hat{E}_{gen} = E_{S,S} + optimism = E_{S,S} + \left(0.368\, E_{S,S} + 0.632 \frac{1}{K} \sum_{k=1}^{K} \left(E_{S^{*k},S} - E_{S^{*k},S^{*k}} \right) \right)$$

- Bootstrap 632+ : more technical

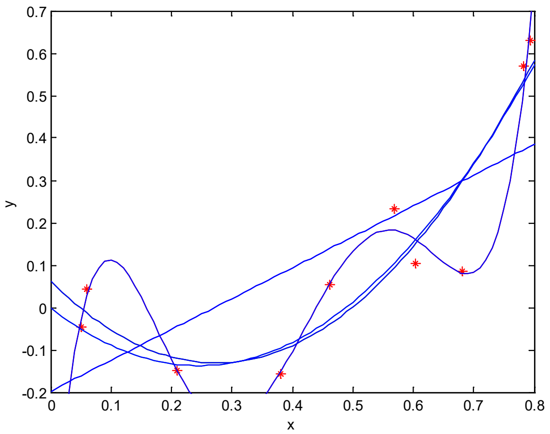
  – Better for Nearest Neighbour problems (lazy learning, vector quantization, *k*-NN classification, ...)

Bootstrap 632

- Bootstrap 632 gives a bias-corrected result (compared to the bootstrap)

$$\hat{E}_{gen} == E_{S,S} + \left(0.368\, E_{S,S} + 0.632\, optimism\right)$$

Here *optimism* is computed only on those samples that were *not* kept in the bootstrap sample



K	linear model	quadratic model	quadratic model without indep. term	5th-order model
10	0.0384	0.0118	0.0115	0.0090
100	0.0342	0.0152	0.0128	9.3315
1000	0.0331	0.0243	0.0134	4.1025
10000	0.0339	0.0244	0.0137	3.3666

The right model is selected

Model structure selection

- Test methods
  - AIC + BIC
  - validation, cross-validation, k-fold and leave-one-out
  - bootstrap

- Model Selection
  - some theoretical insights
  - examples

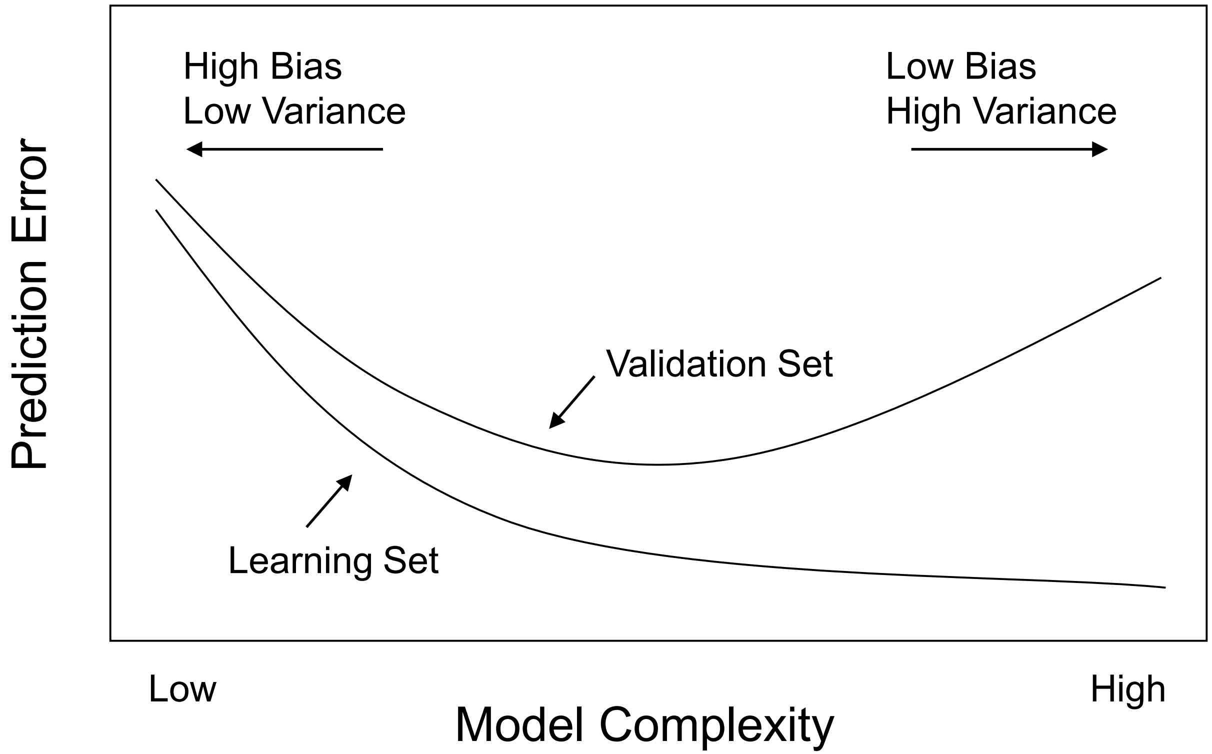
- Test of classifiers

- Pruning
  - pruning during learning (regularization)
  - pruning after learning (direct pruning, local pruning, OBD, OBS)

Model selection is more than finding hyperparameters

- Problem of choosing between different paradigms
  - ex: Linear or nonlinear ?
  - ex: MLP vs. RBFN ?
- Problem of choosing between different structures
  - ex: RBFN with 10 or 15 radial functions ?
- Problem of choosing between different models
  - ex: two structurally identical MLP with different weights (after learning)

Compromise between bias and variance



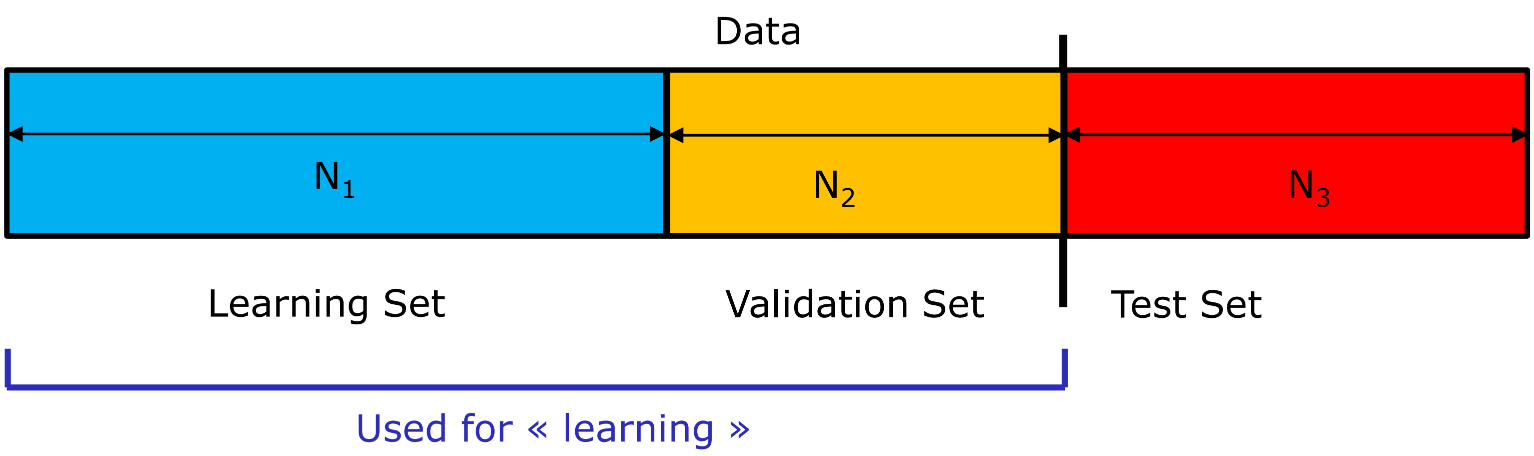
Bias and variance

	Bias	Variance
Leave-one-out	No	High
Cross-validation	No	High
K-fold	No	Moderate
Bootstrap	Yes	Low
Bootstrap 632	No	Low

- Is bias a problem?
  - Not necessarily worse than variance
  - Might be a low-impact problem when models are *compared* (if biases are similar)

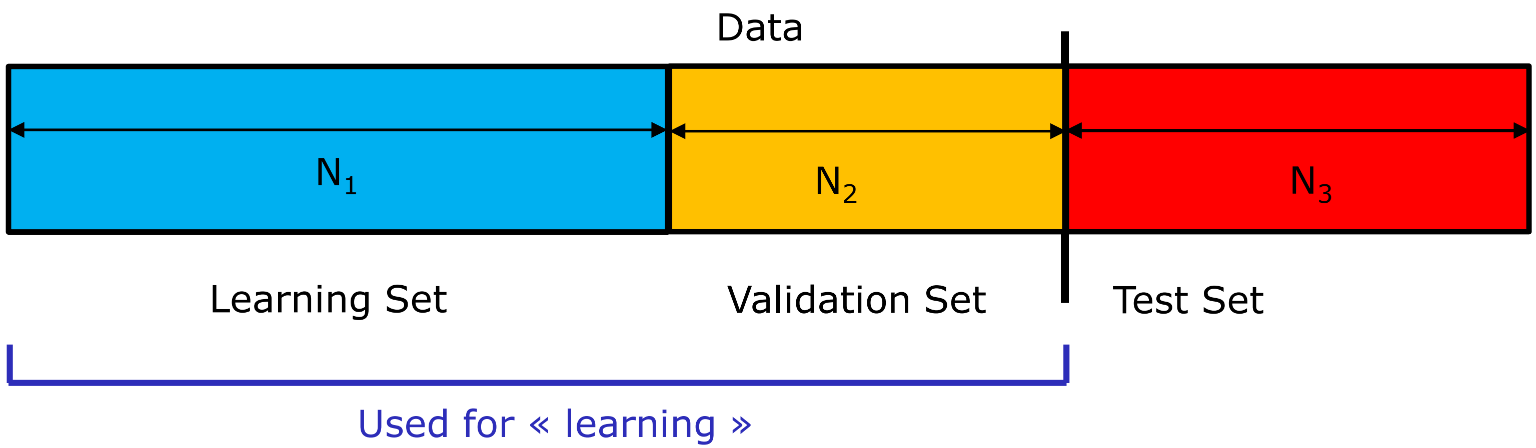
Learning, validation and test

- Warning: validation set ≠ test set
- Validation set is used for learning (the hyperparameters)
  → it cannot be used for (independent, objective) testing
- Need for 3rd, independent test set



- Cross-test can be used (should be, but rarely the case in practice…)

Learning, validation and test



- In practice: this is only to
  - choose a model structure
  - evaluate the performances

- What is the model that has to be delivered to the user ?
  - another one learned on $N_1 + N_2$
  - or another one learned on $N_1 + N_2 + N_3$

Model structure selection

- Test methods
  - AIC + BIC
  - validation, cross-validation, k-fold and leave-one-out
  - bootstrap

- Model Selection
  - some theoretical insights
  - examples

- Test of classifiers

- Pruning
  - pruning during learning (regularization)
  - pruning after learning (direct pruning, local pruning, OBD, OBS)

Test of classifiers

- Measuring the accuracy (% of correct classifications) is nice, but…
  - It does not give any insight on the consequences of possible errors (think to a nuclear plant, that you have to stop or not…)
  - It does not give much information when classes are unbalanced (a wonderful classifier if 99% of data are in class 1 and 1% in class 2 is… classify everything in class 1)

- Better use the confusion matrix:

$$\begin{array}{cc} \left(\begin{array}{cc} C_{11} & C_{12} \\ C_{21} & C_{22} \end{array}\right) & \begin{array}{l} \leftarrow \text{classified in class 1} \\ \leftarrow \text{classified in class 2} \end{array} \\ \begin{array}{cc} \text{true class 1} & \text{true class 2} \end{array} & \end{array}$$

Test of classifiers

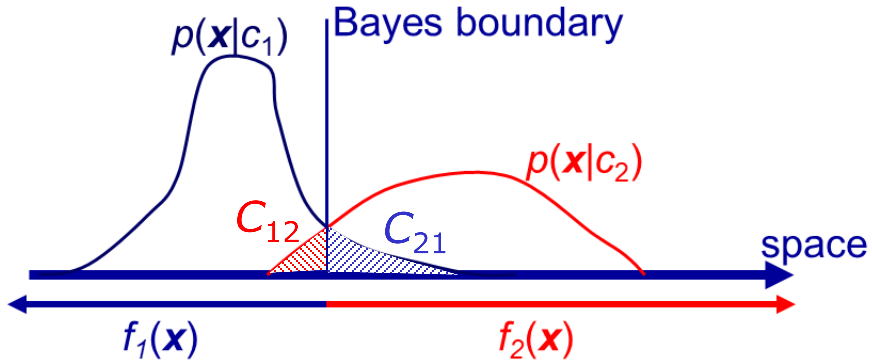
- Let's say that class 1 is 'positive', class 2 'negative'
- We want to retrieve elements from class 1, not from class 2

$$
\begin{array}{cl}
\begin{pmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{pmatrix} & \begin{array}{l} \leftarrow \text{classified in class 1} \\ \leftarrow \text{classified in class 2} \end{array} \\
\begin{array}{cc} \uparrow & \uparrow \\ \text{true class 1} & \text{true class 2} \end{array} &
\end{array}
$$

- Definitions:
  - True positives: c_{11} - true negatives: c_{22}
  - False positives: c_{12} - false negatives: c_{21}
  - Precision: $c_{11} / (c_{11} + c_{12})$
  - Recall: $c_{11} / (c_{11} + c_{21})$
  - Accuracy: $(c_{11} + c_{22}) / (c_{11} + c_{12} + c_{21} + c_{22})$

Bayes classifier

- Even the ideal classifier (Bayes classifier) does not have a unit confusion matrix!
- Here is a 1-dimensional Bayesian classifier:



  - $f_j(x)$ are *indicator* functions
  - Even the best "decision boundary" (here a threshold) makes errors → the confusion matrix is not a unit matrix

Bayes classifier

- Confusion matrix of Bayes (ideal) classifier

$$C_{ij}(f): \int_{\mathcal{R}^D} p(x|c_i) f_j(x) dx = \int_{D_j} p(x|c_i) dx$$

- To compute $C_{ij}(f)$ we should know
  - the ideal (Bayes) classifier (= the exact boundaries D_j of classes)
  - the exact pdf functions $p(x|c_i)$

- In practice
  - we have a classifier, not the Bayes one
  - we estimate the pdf with a finite number of data

- This is the apparent error (because estimated through a finite sample) of an actual classifier (not the ideal classifier)

Model structure selection

- Test methods
  - AIC + BIC
  - validation, cross-validation, k-fold and leave-one-out
  - bootstrap
- Model Selection
  - some theoretical insights
  - examples
- Test of classifiers
- Pruning
  - pruning during learning (regularization)
  - pruning after learning (direct pruning, local pruning, OBD, OBS)

Pruning

- Simplifying a model limits overfitting

- How can we simplify a model ?

  – During learning: regularization

  – After learning: removing parameters without affecting too much the output of the model

Pruning during learning

- "regularization"
- Function $E(w)$ to optimize is replaced with

$$\Theta(w) = E(w) + \lambda C(w)$$

Large number of parameters w
$\rightarrow$ overfitting

compromise !
(bias – variance dilemma…)

penalizes complexity
(or value of weights)
ex:

$$C(w) \sum_i w_i^2$$

$$C(w) \sum_i |w_i|$$

…

- Closer to optimizing the true generalization error (remember AIC and BIC)

Pruning after learning: “direct pruning”

- a unit (neuron) is removed if
  - its output remains fixed
  - its output remains identical (or opposite – or strongly correlated) with another
  - its output is random

- a connection (parameter, or weight) is removed if
  - its contribution to activation is fixed
  - …

- Very heuristic, dangerous except in obvious cases!

Pruning after learning: Local least squares

- A parameter w_{ik} is removed and locally compensated by other parameters in the same sum operation:

$$\sum_{j \neq k} \left(w_{ij} + \delta_{ij} \right) z_j \approx \sum_j w_{ij} z_j$$

or

$$w_{ik} z_k \approx \sum_{j \neq k} \delta_{ij} z_j$$

Pruning after learning: « Optimal Brain Damage » (OBD)

- A parameter w_i is removed, and is not compensated
- The effect of this removal on the error is approximated as follows:

$$\delta E = \sum_i \frac{\partial E}{\partial w_i} \delta w_i + \sum_i \frac{\partial^2 E}{\partial w_i^2} \delta w_i^2 + \sum_{i,j} \frac{\partial^2 E}{\partial w_i \partial w_j} \delta w_i \delta w_j + \Theta(\|w_i\|^3)$$

- We have $\delta w_i = (0 - w_i)$
- Hypotheses
  - Learning leads to a minimum of E
  - E is almost quadratic
- Therefore (for a single δw_i):

$$\delta E = \frac{\partial^2 E}{\partial w_i^2} w_i^2$$

- Remove the parameter with the lowest δE !

Pruning after learning: « Optimal Brain Surgeon » (OBS)

- Still only one parameter w_i is removed, but other ones may vary (to partly compensate)

- Minimize

$$\delta E = \delta \boldsymbol{w}^T H \delta \boldsymbol{w} = \sum_i \frac{\partial^2 E}{\partial w_i^2} \delta w_i^2 + \sum_{i,j} \frac{\partial^2 E}{\partial w_i \partial w_j} \delta w_i \delta w_j$$

subject to constraint $\mathrm{e}_i^T \delta \boldsymbol{w} = -w_i$

- Use of Lagrangian leads to

$$\delta \boldsymbol{w} = -\frac{w_i}{[H^{-1}]_{ji}} H^{-1} e_i$$

Sources and references

- Simulations and some slides come from the work realized by Amaury Lendasse
- Two good books on bootstrap:
  - B. Efron and R.J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, first edition, 1993.
  - A.C. Davison, D.V. Hinkley. *Bootstrap Methods and their Applications.* Cambridge University Press, 3rd edition, 1999.
- A good statistical point of vue for model selection:
  - T. Hastie, R. Tibshirani, J. Friedman. *The Elements of Statistical Learning*, *Data Mining, Inference and Prediction*. Springer Series in Statistics, 4th edition, 2003.
- To my knowledge: no good review on test methods (with practical aspects), although many papers…