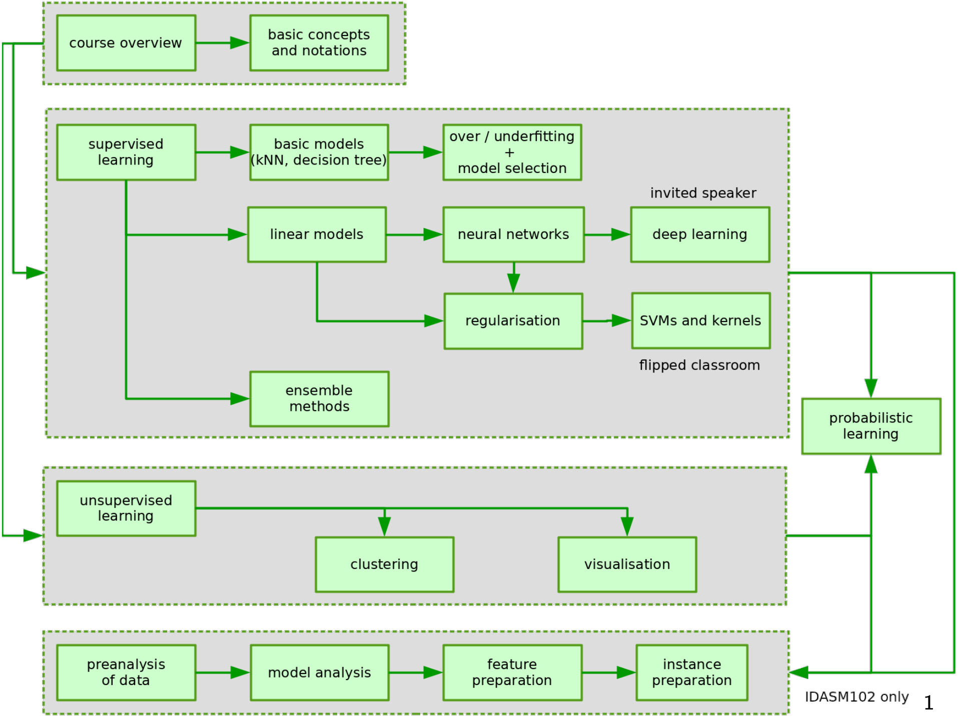
Machine Learning: Lesson 5

Overfitting, Underfitting and Model Selection

Benoît Frénay - Faculty of Computer Science

UNIVERSITÉ DE NAMUR


course overview
basic concepts and notations
supervised learning
basic models (kNN, decision tree)
over / underfitting + model selection
invited speaker
linear models
neural networks
deep learning
regularisation
SVMs and kernels
flipped classroom
ensemble methods
probabilistic learning
unsupervised learning
clustering
visualisation
preanalysis of data
model analysis
feature preparation
instance preparation
IDASM102 only

Outline of this Lesson

- overfitting/underfitting
- definition of model selection
- validation-based model selection
- practical case of model selection
- advanced techniques and model testing

Overfitting/Underfitting

Notion of Model Complexity

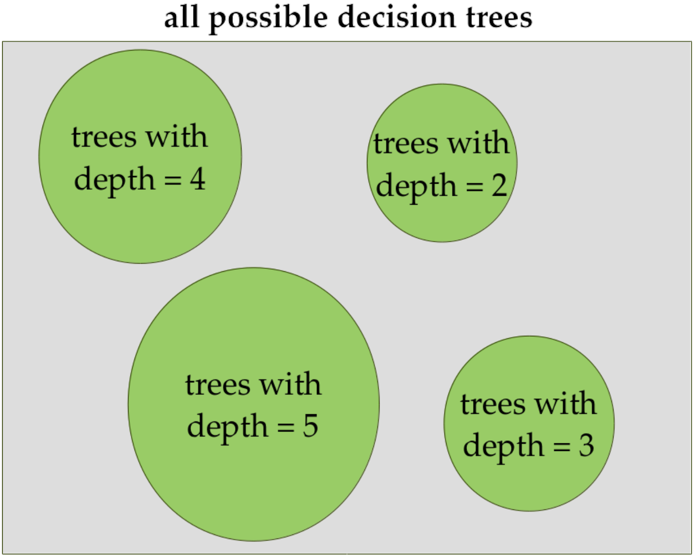
Meta-parameters vs. parameters

model	meta-parameters	parameters
kNN classifier	number k of neighbours	-
decision tree	depth / number of nodes	nodes (decisions)
polynomial	order (largest exponent)	coefficients
neural network	number of neurons	synaptic weights
clustering	number of clusters	center/size of clusters

Model complexity

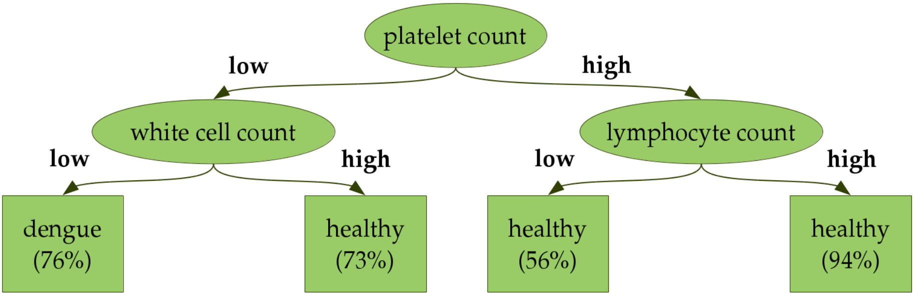
complexity = capacity of the class of function which can be approximated

- meta-parameters: determine the complexity/capacity/architecture of the model (what kind of function *can* be approximated)
- parameters: determines the particular function which is modelled


all possible decision trees
trees with depth = 4
trees with depth = 2
trees with depth = 5
trees with depth = 3

Too Complex Models: the Overfitting Phenomenon

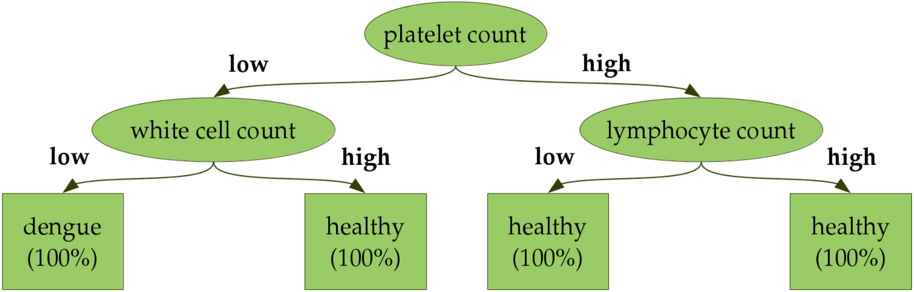
decision tree for dengue fever diagnosis inferred from 1200 cases



not too complex (leaf with **high PC** and **low LC** could probably be split)

Too Complex Models: the Overfitting Phenomenon

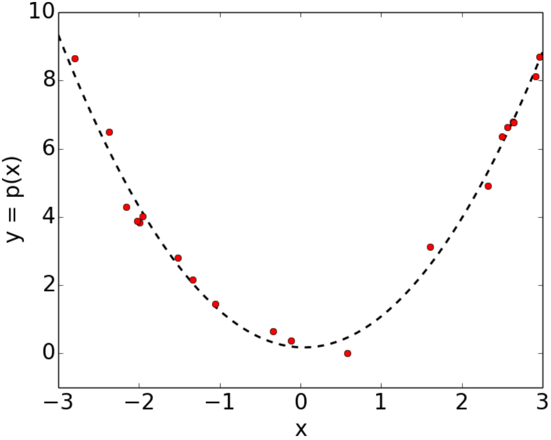
decision tree for dengue fever diagnosis inferred from 4 cases



model the training set perfectly, but poor generalisation on unseen data

Too Complex Models: the Overfitting Phenomenon

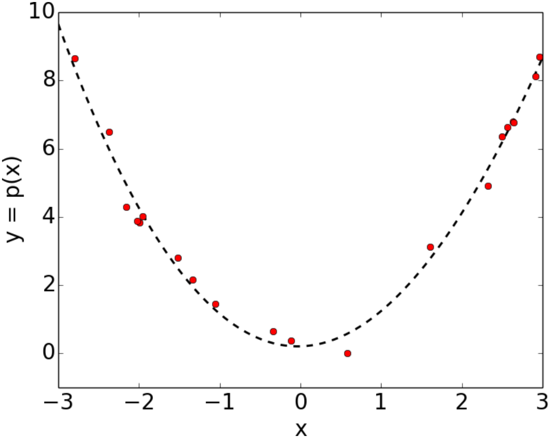
$f(x) = x^2 + \epsilon$ with $\epsilon \sim \mathcal{N}(0, 0.5) \Rightarrow$ polynomial of order $p = 2$ $(n = 9)$



$f(x) \approx 0.47 + 0.13x^1 + 0.96x^2$

Too Complex Models: the Overfitting Phenomenon

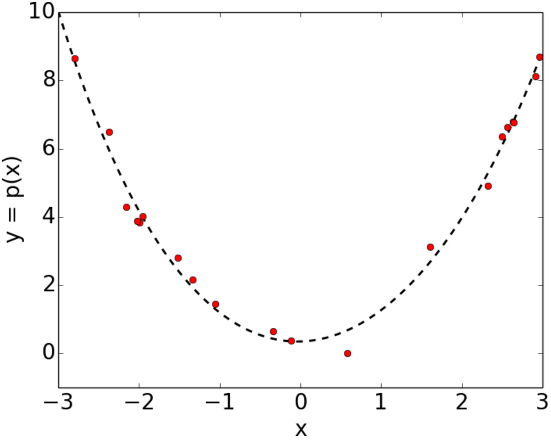
$f(x) = x^2 + \epsilon$ with $\epsilon \sim \mathcal{N}(0, 0.5) \Rightarrow$ polynomial of order $p = 3$ ($n = 9$)



$f(x) \approx 0.42 + 0.61x^1 + 0.99x^2 - 0.09x^3$

Too Complex Models: the Overfitting Phenomenon

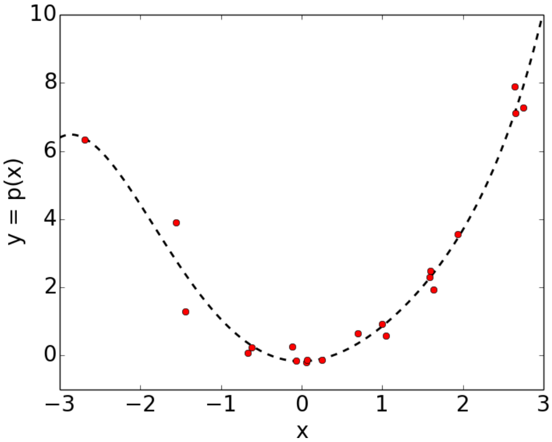
$f(x) = x^2 + \epsilon$ with $\epsilon \sim \mathcal{N}(0, 0.5) \Rightarrow$ polynomial of order $p = 4$ ($n = 9$)



$f(x) \approx 0.47 + 0.57x^1 + 0.91x^2 - 0.08x^3 + 0.01x^4$

Too Complex Models: the Overfitting Phenomenon

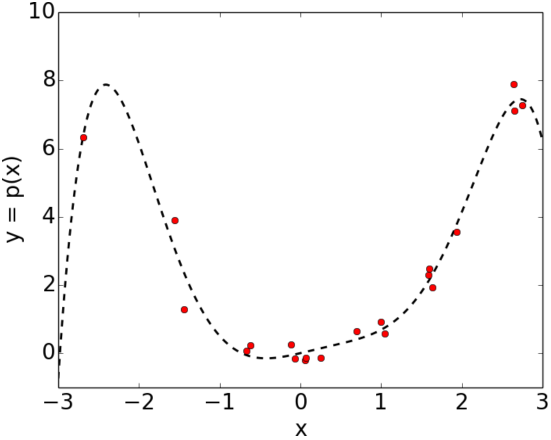
$f(x) = x^2 + \epsilon$ with $\epsilon \sim \mathcal{N}(0, 0.5) \Rightarrow$ polynomial of order $p = 5$ ($n = 9$)



$f(x) \approx 0.25 + 1.55x^1 + 1.04x^2 - 0.54x^3 - 0.00x^4 + 0.05x^5$

Too Complex Models: the Overfitting Phenomenon

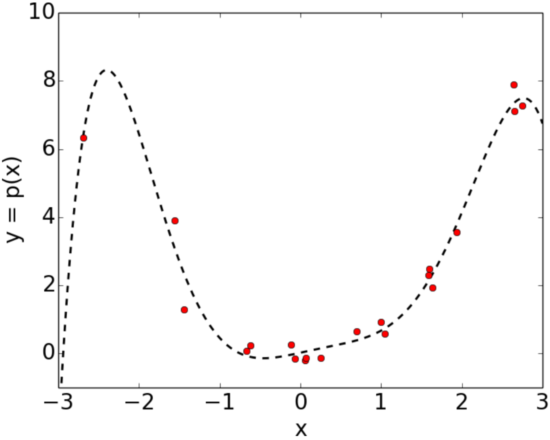
$f(x) = x^2 + \epsilon$ with $\epsilon \sim \mathcal{N}(0, 0.5) \Rightarrow$ polynomial of order $p = 6$ $(n = 9)$



$f(x) \approx 0.32 + 1.58x^1 + 0.57x^2 - 0.58x^3 + 0.18x^4 + 0.05x^5 - 0.02x^6$

Too Complex Models: the Overfitting Phenomenon

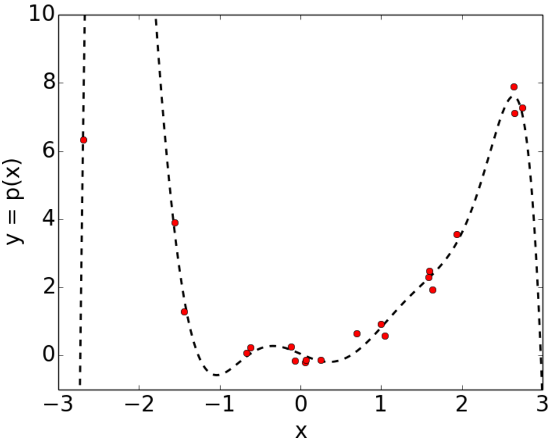
$f(x) = x^2 + \epsilon$ with $\epsilon \sim \mathcal{N}(0, 0.5) \Rightarrow$ polynomial of order $p = 7$ $(n = 9)$



$f(x) \approx 0.30 + 2.87x^1 - 0.51x^2 - 2.32x^3 + 0.57x^4 + 0.57x^5 - 0.05x^6 - 0.04x^7$

Too Complex Models: the Overfitting Phenomenon

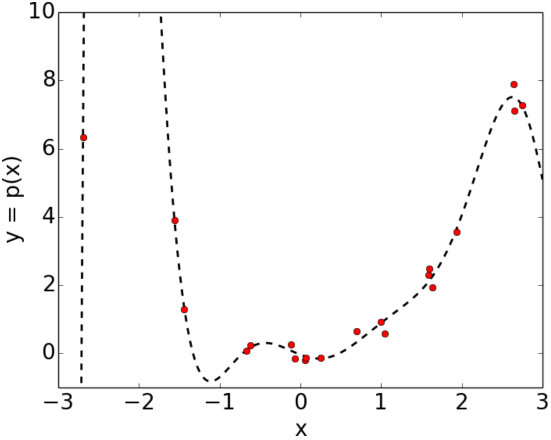
$f(x) = x^2 + \epsilon$ with $\epsilon \sim \mathcal{N}(0, 0.5) \Rightarrow$ polynomial of order $p = 8$ $(n = 9)$



$f(x) \approx -4 - 127x^1 + 202x^2 + 124x^3 - 130x^4 - 34x^5 + 28x^6 + 3x^7 - 2x^8$

Too Complex Models: the Overfitting Phenomenon

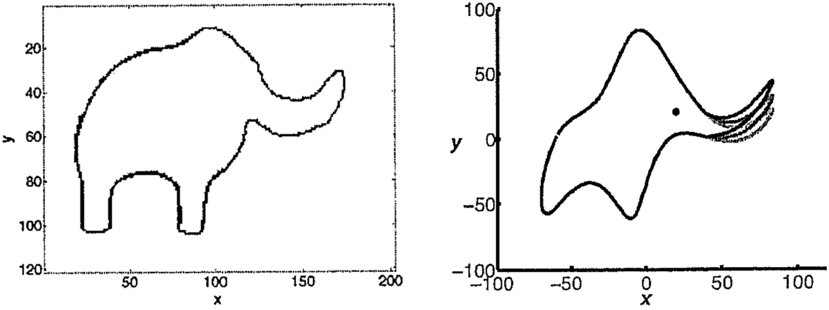
$f(x) = x^2 + \epsilon$ with $\epsilon \sim \mathcal{N}(0, 0.5) \Rightarrow$ polynomial of order $p = 9$ $(n = 9)$



$f(x) \approx -0 - 21x^1 - 8x^2 + 86x^3 + 19x^4 - 48x^5 - 6x^6 + 9x^7 + 1x^8 - 1x^9$

About Elephants and Complex Models

John von Neumann (attributed by Enrico Fermi): *"With four parameters I can fit an elephant, and with five I can make him wiggle his trunk"*.

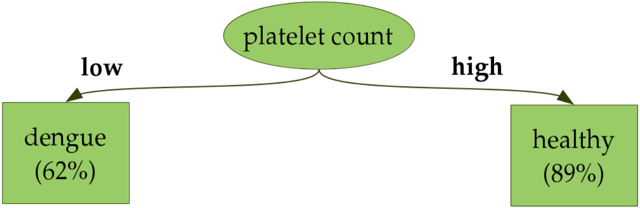


source: *"Drawing an elephant with four complex parameters" by Jurgen Mayer, Khaled Khairy, and Jonathon Howard, Am. J. Phys. 78, 648 (2010), DOI:10.1119/1.3254017.*

see also `http://www.johndcook.com/blog/2011/06/21/how-to-fit-an-elephant/`

Too Simple Models: the Underfitting Phenomenon

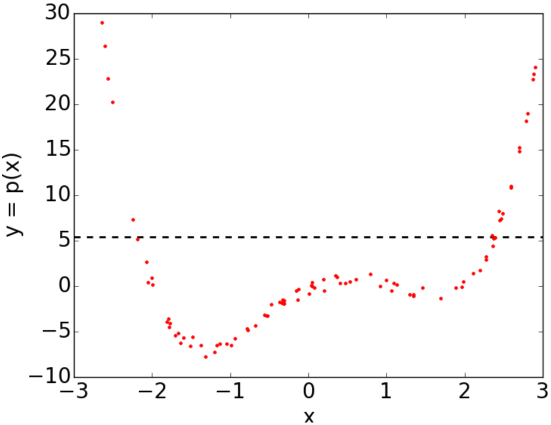
decision tree for dengue fever diagnosis inferred from 1200 cases



unable to model the training set and poor generalisation on unseen data

Too Simple Models: the Underfitting Phenomenon

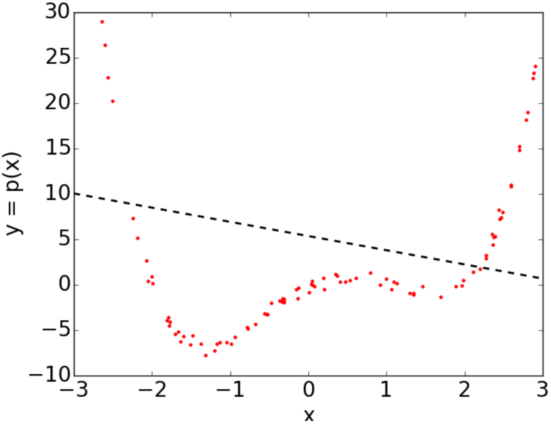
$f(x) = x(x-1)(x+2)(x-2) + \epsilon \Rightarrow$ polynomial of order $p = 0$ $(n = 100)$



$f(x) \approx 5.41$

Too Simple Models: the Underfitting Phenomenon

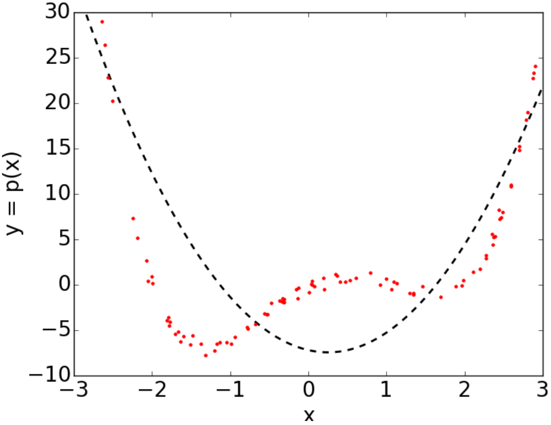
$f(x) = x(x-1)(x+2)(x-2) + \epsilon \Rightarrow$ polynomial of order $p = 1$ ($n = 100$)



$f(x) \approx 5.37 - 1.56x^1$

Too Simple Models: the Underfitting Phenomenon

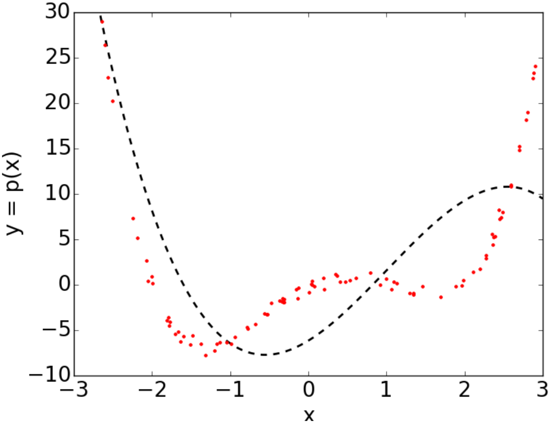
$f(x) = x(x-1)(x+2)(x-2) + \epsilon \Rightarrow$ polynomial of order $p = 2$ ($n = 100$)



$f(x) \approx -7.19 - 1.97x^1 + 3.89x^2$

Too Simple Models: the Underfitting Phenomenon

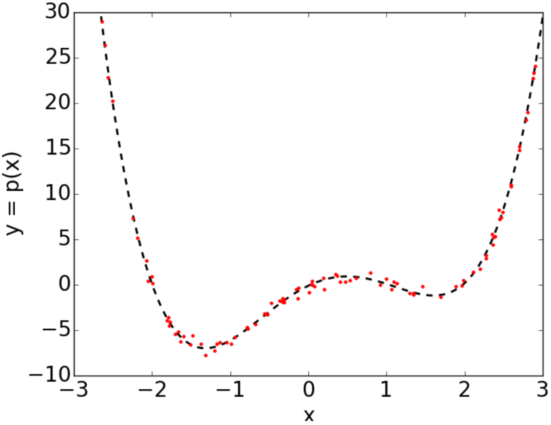
$f(x) = x(x-1)(x+2)(x-2) + \epsilon \Rightarrow$ polynomial of order $p = 3$ ($n = 100$)



$f(x) \approx -6.16 + 5.26x^1 + 3.72x^2 - 1.24x^3$

Too Simple Models: the Underfitting Phenomenon

$f(x) = x(x-1)(x+2)(x-2) + \epsilon \Rightarrow$ polynomial of order $p = 4$ ($n = 100$)



$f(x) \approx -0.11 + 4.14x^1 - 3.89x^2 - 1.03x^3 + 0.99x^4$

Motivation for Model Selection

Overfitting vs. underfitting

- overfitting: too complex $\Rightarrow$ learn data "by heart" $\Rightarrow$ "stupid" model
- underfitting: not complex enough $\Rightarrow$ unable to model dataset
- in both cases: poor generalisation performance (too simple/complex)
- cause: choice of meta-parameters (complexity/capacity/architecture)

Meta-parameters vs. parameters

- meta-parameters: chosen before learning (model selection)
- parameters: obtained after learning (influenced by meta-parameters)

Choice of the meta-parameter

- question: how to select the right complexity ?
- answer depends on the dataset (number of instances, quality)

Definition of Model Selection

Model Selection for Polynomial Fitting

Experimental settings

- process generating data: $f(x) = x^2 + \epsilon$ with $\epsilon \sim \mathcal{N}(0, 0.5)$
- only $n = 9$ training instances are available
- polynomials of order $p = 0, \ldots, 9$ are considered

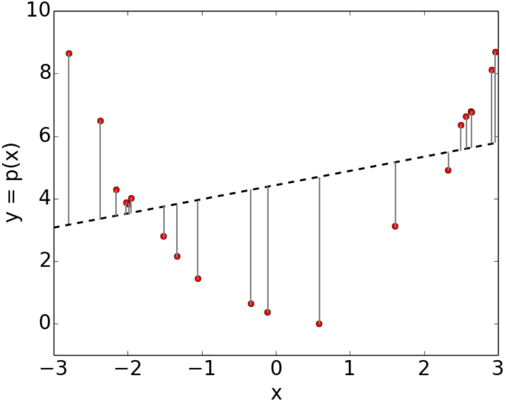
Question: what is the right model complexity ?

- meta-parameter = order p of the polynomial
- goal: choose model complexity for best generalisation
- issue: in practice, the process generating data is unknown
- common trick: generalisation error $\approx$ error on independent sample
- 10^6 instances are used here to estimate the generalisation error

Model Selection for Polynomial Fitting

Error criterion: mean square error (MSE)

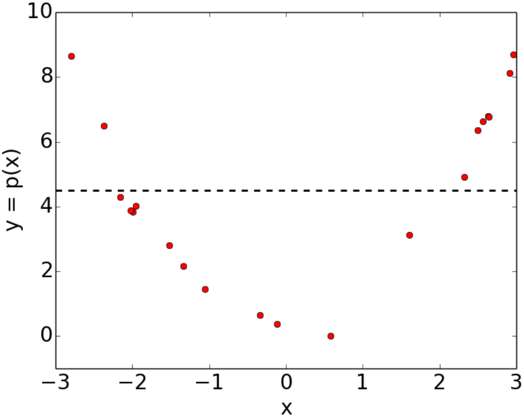
$$E = \frac{1}{n}\sum_{i=1}^{n}(f(\mathbf{x}_i) - t_i)^2$$



Model Selection for Polynomial Fitting

$f(x) = x^2 + \epsilon$ with $\epsilon \sim \mathcal{N}(0, 0.5) \Rightarrow$ polynomial of order 0 ($n = 20$)

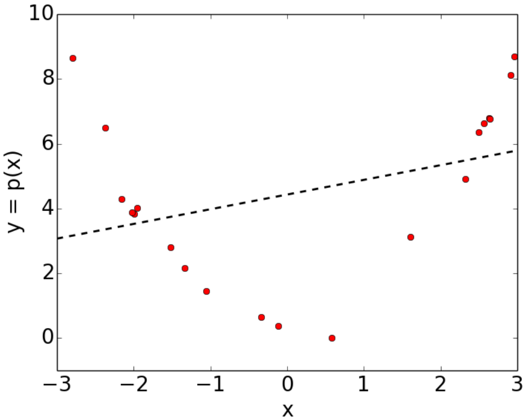
p	E_{train}	$\hat{E}_{\text{gen}}$
$\rightarrow$ **0**	7.291	7.927
1	6.288	10.013
2	0.272	0.275
3	0.223	0.351
4	0.217	0.374
5	0.205	0.484
6	0.151	2.511
7	0.151	3.243
8	0.070	211.622
9	0.046	791.109



Model Selection for Polynomial Fitting

$f(x) = x^2 + \epsilon$ with $\epsilon \sim \mathcal{N}(0, 0.5) \Rightarrow$ polynomial of order 1 ($n = 20$)

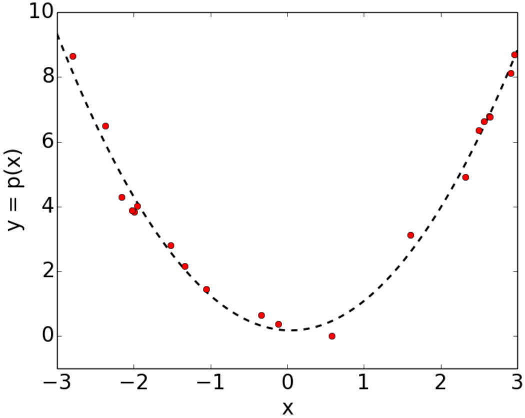
p	E_{train}	$\hat{E}_{\text{gen}}$
0	7.291	7.927
$\rightarrow$ **1**	6.288	10.013
2	0.272	0.275
3	0.223	0.351
4	0.217	0.374
5	0.205	0.484
6	0.151	2.511
7	0.151	3.243
8	0.070	211.622
9	0.046	791.109



Model Selection for Polynomial Fitting

$f(x) = x^2 + \epsilon$ with $\epsilon \sim \mathcal{N}(0, 0.5) \Rightarrow$ polynomial of order 2 ($n = 20$)

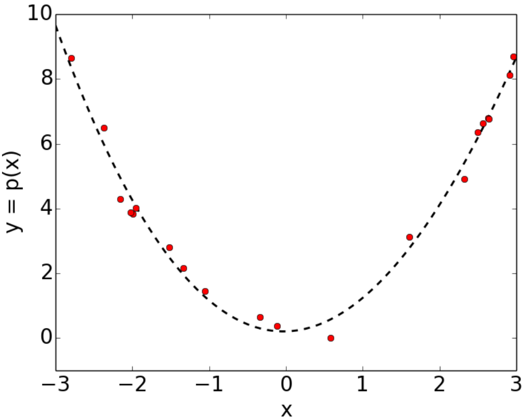
p	E_{train}	$\hat{E}_{\text{gen}}$
0	7.291	7.927
1	6.288	10.013
$\rightarrow$ **2**	0.272	0.275
3	0.223	0.351
4	0.217	0.374
5	0.205	0.484
6	0.151	2.511
7	0.151	3.243
8	0.070	211.622
9	0.046	791.109



Model Selection for Polynomial Fitting

$f(x) = x^2 + \epsilon$ with $\epsilon \sim \mathcal{N}(0, 0.5) \Rightarrow$ polynomial of order 3 ($n = 20$)

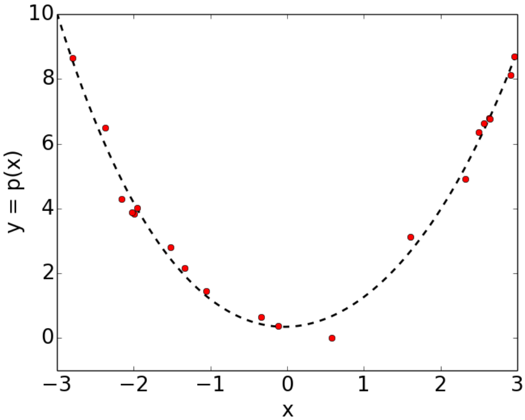
p	E_{train}	$\hat{E}_{\text{gen}}$
0	7.291	7.927
1	6.288	10.013
2	0.272	0.275
$\rightarrow$ **3**	0.223	0.351
4	0.217	0.374
5	0.205	0.484
6	0.151	2.511
7	0.151	3.243
8	0.070	211.622
9	0.046	791.109



Model Selection for Polynomial Fitting

$f(x) = x^2 + \epsilon$ with $\epsilon \sim \mathcal{N}(0, 0.5) \Rightarrow$ polynomial of order 4 ($n = 20$)

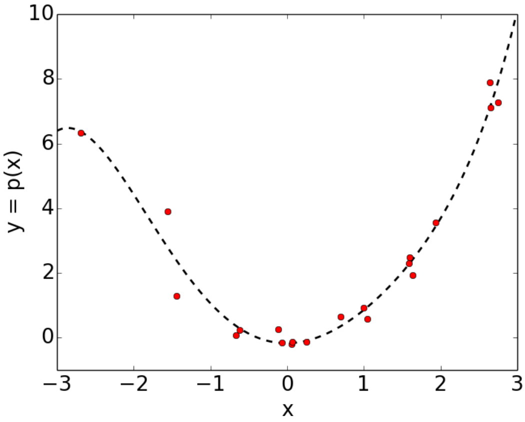
p	E_{train}	$\hat{E}_{\text{gen}}$
0	7.291	7.927
1	6.288	10.013
2	0.272	0.275
3	0.223	0.351
$\rightarrow$ **4**	0.217	0.374
5	0.205	0.484
6	0.151	2.511
7	0.151	3.243
8	0.070	211.622
9	0.046	791.109



Model Selection for Polynomial Fitting

$f(x) = x^2 + \epsilon$ with $\epsilon \sim \mathcal{N}(0, 0.5) \Rightarrow$ polynomial of order 5 ($n = 20$)

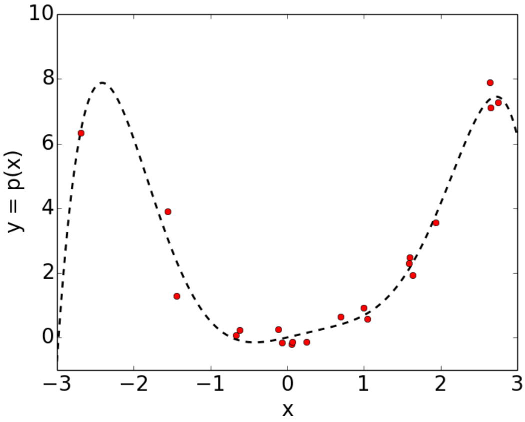
p	E_{train}	$\hat{E}_{\text{gen}}$
0	7.291	7.927
1	6.288	10.013
2	0.272	0.275
3	0.223	0.351
4	0.217	0.374
$\rightarrow$ **5**	0.205	0.484
6	0.151	2.511
7	0.151	3.243
8	0.070	211.622
9	0.046	791.109



Model Selection for Polynomial Fitting

$f(x) = x^2 + \epsilon$ with $\epsilon \sim \mathcal{N}(0, 0.5) \Rightarrow$ polynomial of order 6 ($n = 20$)

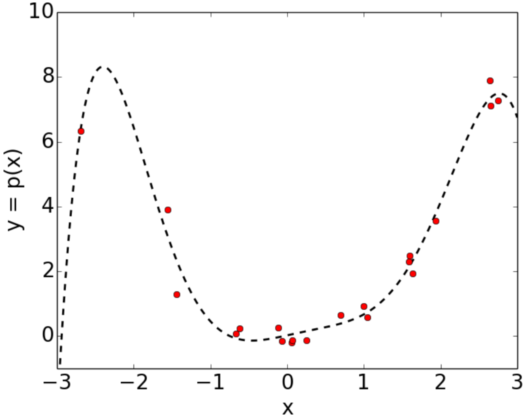
p	E_{train}	$\hat{E}_{\text{gen}}$
0	7.291	7.927
1	6.288	10.013
2	0.272	0.275
3	0.223	0.351
4	0.217	0.374
5	0.205	0.484
$\rightarrow$ **6**	0.151	2.511
7	0.151	3.243
8	0.070	211.622
9	0.046	791.109



Model Selection for Polynomial Fitting

$f(x) = x^2 + \epsilon$ with $\epsilon \sim \mathcal{N}(0, 0.5) \Rightarrow$ polynomial of order 7 ($n = 20$)

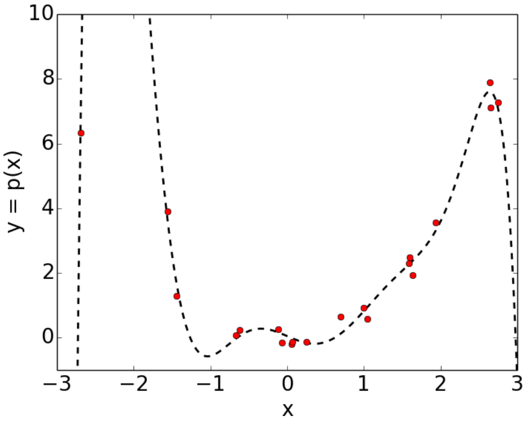
p	E_{train}	$\hat{E}_{\text{gen}}$
0	7.291	7.927
1	6.288	10.013
2	0.272	0.275
3	0.223	0.351
4	0.217	0.374
5	0.205	0.484
6	0.151	2.511
$\rightarrow$ **7**	0.151	3.243
8	0.070	211.622
9	0.046	791.109



Model Selection for Polynomial Fitting

$f(x) = x^2 + \epsilon$ with $\epsilon \sim \mathcal{N}(0, 0.5) \Rightarrow$ polynomial of order 8 ($n = 20$)

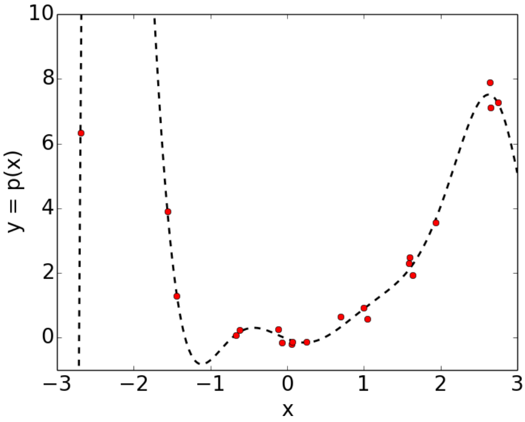
p	E_{train}	$\hat{E}_{\text{gen}}$
0	7.291	7.927
1	6.288	10.013
2	0.272	0.275
3	0.223	0.351
4	0.217	0.374
5	0.205	0.484
6	0.151	2.511
7	0.151	3.243
$\rightarrow$ **8**	0.070	211.622
9	0.046	791.109



Model Selection for Polynomial Fitting

$f(x) = x^2 + \epsilon$ with $\epsilon \sim \mathcal{N}(0, 0.5) \Rightarrow$ polynomial of order 9 ($n = 20$)

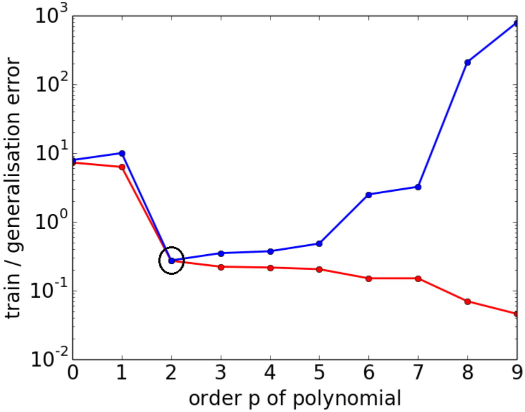
p	E_{train}	$\hat{E}_{\text{gen}}$
0	7.291	7.927
1	6.288	10.013
2	0.272	0.275
3	0.223	0.351
4	0.217	0.374
5	0.205	0.484
6	0.151	2.511
7	0.151	3.243
8	0.070	211.622
$\rightarrow$ **9**	0.046	791.109



Model Selection for Polynomial Fitting

$f(x) = x^2 + \epsilon$ with $\epsilon \sim \mathcal{N}(0, 0.5) \Rightarrow$ best order $p = 2$ ($E_{\text{gen}} \approx 0.5^2 = .25$)

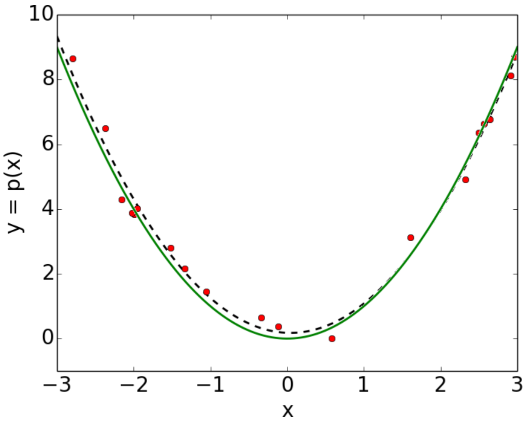
p	E_{train}	$\hat{E}_{\text{gen}}$
0	7.291	7.927
1	6.288	10.013
* 2	**0.272**	**0.275**
3	0.223	0.351
4	0.217	0.374
5	0.205	0.484
6	0.151	2.511
7	0.151	3.243
8	0.070	211.622
9	0.046	791.109



Model Selection for Polynomial Fitting

$f(x) = x^2 + \epsilon$ with $\epsilon \sim \mathcal{N}(0, 0.5) \Rightarrow$ best order $p = 2$ ($E_{\text{gen}} \approx 0.5^2 = .25$)

p	E_{train}	$\hat{E}_{\text{gen}}$
0	7.291	7.927
1	6.288	10.013
* 2	**0.272**	**0.275**
3	0.223	0.351
4	0.217	0.374
5	0.205	0.484
6	0.151	2.511
7	0.151	3.243
8	0.070	211.622
9	0.046	791.109



Model Selection: from Theory to Practice

Definition

Model selection consists in choosing the best meta-parameters for a model.

In practice

- model selection is performed **before** the parameter optimisation
- meta-parameters **should not/cannot** be chosen by hand (intractable)
- meta-parameters depend on dataset characteristics (size, quality, etc.)

Generalisation error minimisation

- the generalisation error is unknown and has to be estimated
- the training error **cannot** be used (biased, overoptimistic estimator)
- efficient use of the limited amount of data (we cheated in the example)

Model Selection: from Theory to Practice

Definition

Model selection consists in choosing the best meta-parameters for a model.

In practice

- model selection is performed **before** the parameter optimisation
- meta-parameters **should not**/**cannot** be chosen by hand (intractable)
- meta-parameters depend on dataset characteristics (size, quality, etc.)

Generalisation error minimisation

- the generalisation error is unknown and has to be estimated
- the training error **cannot** be used (biased, overoptimistic estimator)
- efficient use of the limited amount of data (we cheated in the example)

Model Selection: Common Error Criteria and Estimators

Generalisation in regression

in regression, the average square error for model f is used since

$$\mathop{\mathbb{E}}_{\mathbf{x}}\left[\mathop{\mathbb{E}}_{t|\mathbf{x}}\left[(f(\mathbf{x})-t)^2\right]\right]=\mathop{\mathbb{E}}_{\mathbf{x},t}\left[(f(\mathbf{x})-t)^2\right]\approx\frac{1}{n}\sum_{i=1}^{n}(f(\mathbf{x}_i)-t_i)^2$$

Generalisation in classification

in classification, the misclassification rate for model f is used since

$$\mathop{\mathbb{E}}_{\mathbf{x}}\left[\mathop{\mathbb{E}}_{t|\mathbf{x}}\left[\mathbb{I}\left[f(\mathbf{x})\neq t\right]\right]\right]=\mathop{\mathbb{E}}_{\mathbf{x},t}\left[\mathbb{I}\left[f(\mathbf{x})\neq t\right]\right]\approx\frac{1}{n}\sum_{i=1}^{n}\mathbb{I}\left[f(\mathbf{x}_i)\neq t_i\right]$$

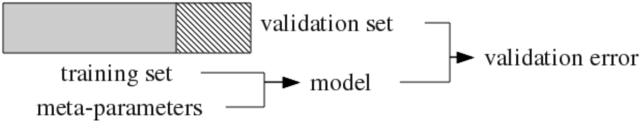
both estimators converge to the true generalisation error when $n\to\infty$, but in practice the size of the validation set n is finite $\Rightarrow$ we must be careful!

Validation-Based Model Selection

Simple Validation

Procedure

- split data into a training set and a validation set
- training set is used to train the model with meta-parameters
- validation set is used to estimate the generalisation error



Pros and cons

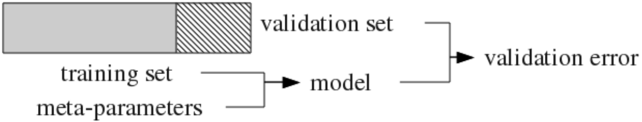
✓ easy and fast, intuitive, converge when $n \to \infty$ (unbiased estimator)

× unreliable: only one repetition, what if we are (un)lucky ?

Cross-Validation

Procedure

- same than simple validation, except that the procedure is repeated
- dataset is shuffled before each repetition
- generalisation error estimates for each repetition are averaged



Pros and cons

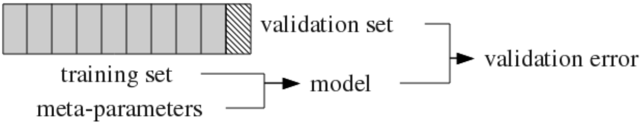
✓ more reliable: unlikely to be (un)lucky if enough repetitions

× potentially large overlapping between training and validation sets

k-Fold Cross-Validation

Procedure

- dataset is split in k folds (fixed over repetitions), usually $k = 10$
- at each repetition, training $= k - 1$ folds and validation $= 1$ fold
- each fold is only once used as the validation set



Pros and cons

✓ same advantages than cross-validation, small number k of repetitions

✓ no overlapping (generalisation error estimates are $\pm$ independent)

Grid Search for Meta-Parameter Optimisation

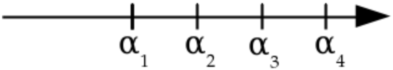
grid_search($\mathcal{D}$, k)

Input: dataset $\mathcal{D} = \{(\mathbf{x}_i, t_i)\}$ and number of folds k
Output: model with optimal meta-parameters
for each possible meta-parameter values $\boldsymbol{\alpha}$ **do**
  $\hat{E}_{\text{gen}}(\boldsymbol{\alpha}) = \text{compute\_kfcv\_error}(\mathcal{D}, k, \boldsymbol{\alpha})$
end for
return model learnt with $\mathcal{D}$ and meta-parameters $\boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha}} \hat{E}_{\text{gen}}(\boldsymbol{\alpha})$

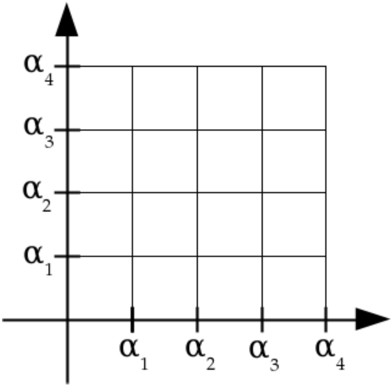
compute_kfcv_error($\mathcal{D}$, k, $\boldsymbol{\alpha}$)

Input: dataset $\mathcal{D} = \{(\mathbf{x}_i, t_i)\}$, number of folds k and meta-parameters $\boldsymbol{\alpha}$
Output: estimated generalisation error of the best model with meta-parameters $\boldsymbol{\alpha}$
$\hat{E}_{\text{gen}}(\boldsymbol{\alpha}) = 0$
for each fold **do**
  divide the k folds of $\mathcal{D}$ in $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{val}}$
  learn model with $\mathcal{D}_{\text{train}}$ and meta-parameters $\boldsymbol{\alpha}$
  $\hat{E}_{\text{gen}}(\boldsymbol{\alpha}) \mathrel{+}= \frac{1}{k}$ (prediction error of model on $\mathcal{D}_{\text{val}}$)
end for
return $\hat{E}_{\text{gen}}(\boldsymbol{\alpha})$

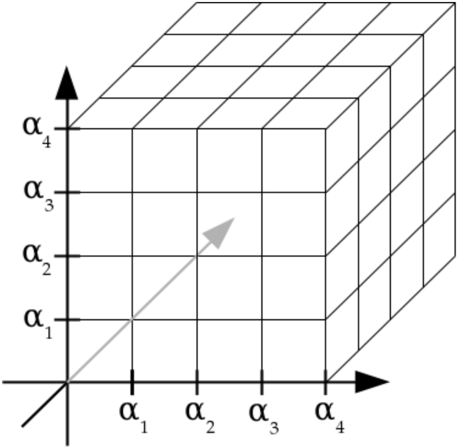
Hypergrids for Grid Search



Hypergrids for Grid Search



Hypergrids for Grid Search

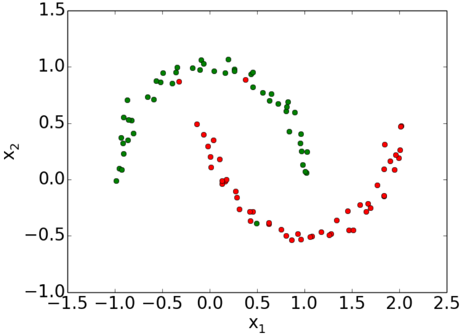


Practical Case of Model Selection

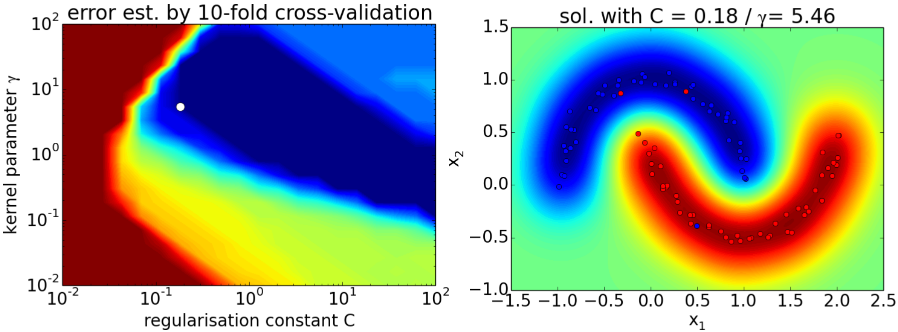
Experimental Settings

Dataset

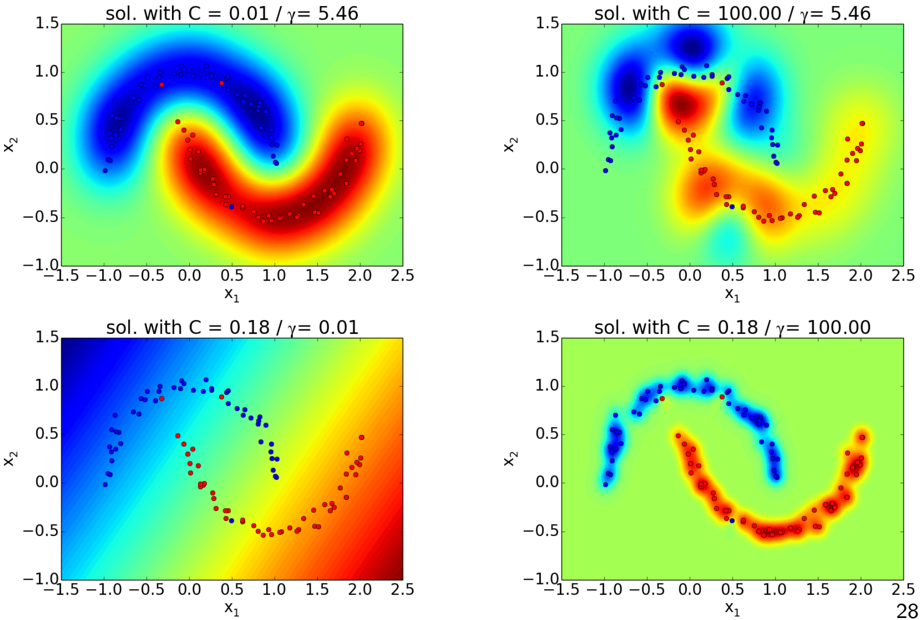
- artificial problem "Two Moons" (`sklearn.datasets.make_moons`)
- $n = 30$ (inc. 3 mislabelled) with non-linear support vector machine
  - meta-parameter C : regularisation constant (simple $\leftrightarrow$ complex)
  - meta-parameter γ: scale at which we "look" at data (small $\leftrightarrow$ large)



Results with 10-fold Cross-Validation



Results with Suboptimal Meta-parameter Choices



Advanced Techniques and Model Testing

Comparison with other Techniques

Akaike/Bayesian information criterion for linear models

- AIC: $\hat{E}_{\text{gen}} = E_{\text{train}} + \frac{2}{n} dim(\theta)$ BIC: $\hat{E}_{\text{gen}} = E_{\text{train}} + \frac{\log n}{n} dim(\theta)$
- based on (strong) simplifying assumptions: lead to overfitting

Leave-one-out (LOO)

- k-fold CV with $k = n$ (analytical expression for linear methods)
- only used in specific cases: otherwise, very costly and high variance

Bootstrap

- estimates the bias of the training error $E_{\text{gen}} - E_{\text{train}}$ with resampling
- theoretically better than validation-based schemes (smaller variance)
- not used in practice because thousands of resampling are necessary

Comparison with other Techniques

Akaike/Bayesian information criterion for linear models

- AIC: $\hat{E}_{\text{gen}} = E_{\text{train}} + \frac{2}{n} dim(\theta)$ BIC: $\hat{E}_{\text{gen}} = E_{\text{train}} + \frac{\log n}{n} dim(\theta)$
- based on (strong) simplifying assumptions: lead to overfitting

Leave-one-out (LOO)

- k-fold CV with $k = n$ (analytical expression for linear methods)
- only used in specific cases: otherwise, very costly and high variance

Bootstrap

- estimates the bias of the training error $E_{\text{gen}} - E_{\text{train}}$ with resampling
- theoretically better than validation-based schemes (smaller variance)
- not used in practice because thousands of resampling are necessary

Comparison with other Techniques

Akaike/Bayesian information criterion for linear models

- AIC: $\hat{E}_{\text{gen}} = E_{\text{train}} + \frac{2}{n} dim(\theta)$ BIC: $\hat{E}_{\text{gen}} = E_{\text{train}} + \frac{\log n}{n} dim(\theta)$
- based on (strong) simplifying assumptions: lead to overfitting

Leave-one-out (LOO)

- k-fold CV with $k = n$ (analytical expression for linear methods)
- only used in specific cases: otherwise, very costly and high variance

Bootstrap

- estimates the bias of the training error $E_{\text{gen}} - E_{\text{train}}$ with resampling
- theoretically better than validation-based schemes (smaller variance)
- not used in practice because thousands of resampling are necessary

Why we Need Model Testing

Training and validation error cannot be used

- training error is biased since it was used to select parameters
- validation error is biased since it was used to select meta-parameters

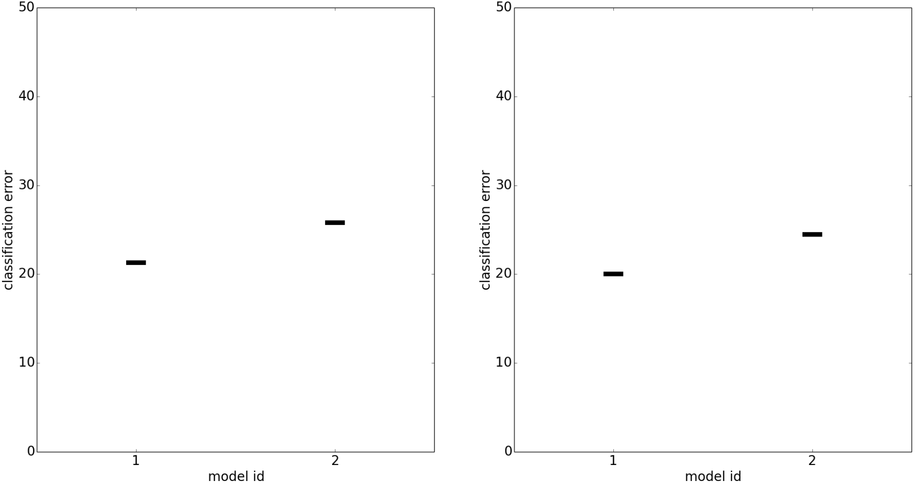
In practice

- use another set of instances which has not been used yet
- assess the method in a real setting (where it is supposed to be used)

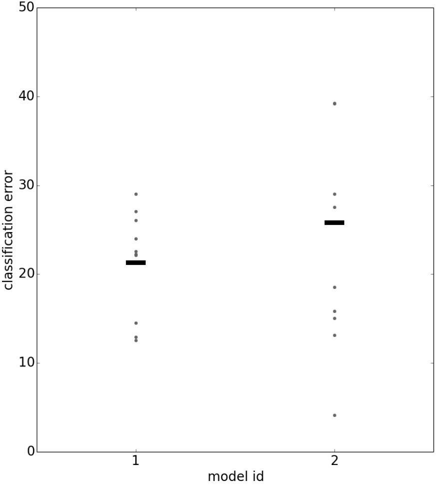
About model selection and testing

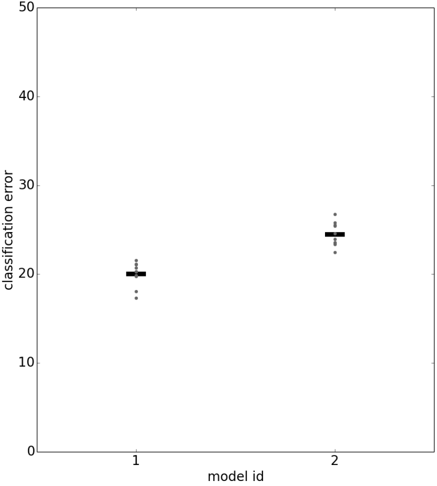
validation techniques are **very** important: training error cannot be trusted

Model Comparison in Terms of Generalisation Error

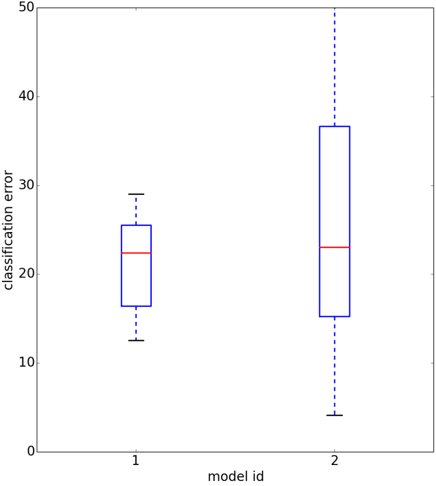


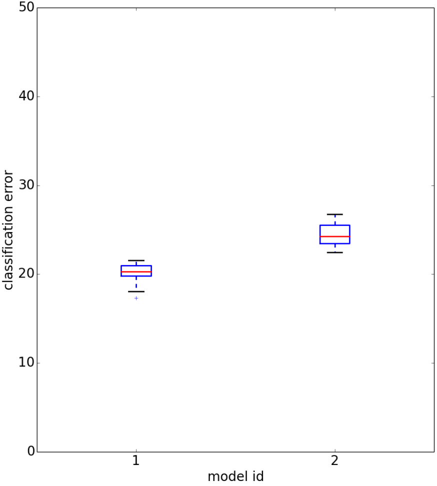
Model Comparison in Terms of Generalisation Error





Model Comparison in Terms of Generalisation Error

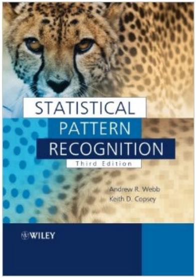


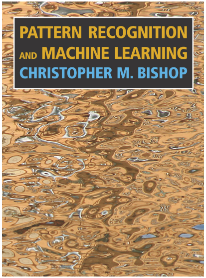


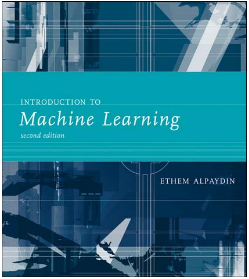
Outline of this Lesson

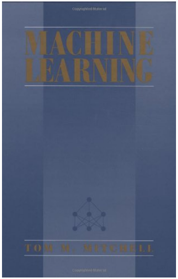
- overfitting/underfitting
- definition of model selection
- validation-based model selection
- practical case of model selection
- advanced techniques and model testing

References


STATISTICAL
PATTERN
RECOGNITION
Third Edition
Andrew R. Webb
Keith D. Copsey
WILEY


PATTERN RECOGNITION
AND MACHINE LEARNING
CHRISTOPHER M. BISHOP


INTRODUCTION TO
Machine Learning
second edition
ETHEM ALPAYDIN


MACHINE
LEARNING
TOM M. MITCHELL