

# ARTIFICIAL INTELLIGENCE & CYBER SECURITY

## INTRO TO MACHINE LEARNING REGULARIZATION

Estefanía Talavera Martínez  
e.talaveramartinez@utwente.nl

# supervised learning - generalisation

key problems in supervised learning

**model selection:** Neural Networks, Support Vector Machine, Decision trees... ?

how many prototypes/seeds, neurons, which kernel,... ?

**data representation:** coding, normalization, transformation, ... ?

**algorithm, (hyper-) parameters:**

which training prescription ?

how many training epochs, which learning rate, ... ?

---

training: based on performance with respect to training data

aim : low error with respect to new data **generalization**

how can we test the generalization performance ?

## Validation procedures

basic idea: split available data  $D = \left\{ \left\{ \xi^\mu, S^\mu \right\} \right\}_{\mu=1}^P$

(randomly) into disjoint sets

$$D_{training} = \left\{ \left\{ \xi^\mu, S^\mu \right\} \right\}_{\mu=1}^Q \quad D_{test} = \left\{ \left\{ \xi^\mu, S^\mu \right\} \right\}_{\mu=Q+1}^P$$

→ estimate of test error  $E_{test}$  (e.g. number of misclassifications)

→ comparison/choice of different models, algorithms, parameter settings

→ prediction of performance with respect to novel data ( ? )

# supervised learning - generalisation

## problems:

### - lack of data

can we afford to *waste* example data *only* for validation ?

### - representative results ?

*lucky / unlucky* set composition can give misleading outcome !

### - variation of results ?

how safe is the prediction ? error bars of the estimates ?

**example strategy: " n-fold cross-validation "**

split data  $D = \left\{ \left\{ \xi^\mu, S^\mu \right\} \right\}_{\mu=1}^P$  (randomly) into **n** disjoint sets

$$D = \bigcup_{i=1}^n D^{(i)} \quad D_{train}^{(i)} = D \setminus D^{(i)} \quad D_{test}^{(i)} = D^{(i)}$$

all data

training data (i)

test data (i)

- repeat training n times
- calculate average training / test errors ( and variances )

- repeat cross-validation for different models, parameter settings, etc.
- select the best system with respect to test errors  
(model, number of units, learning rate, ... )

## remarks:

### - **which n** in n-fold cross-validation ?

- larger n → larger fraction of D used in each training run
- more estimates of  $E_{\text{test}}$  / smaller test sets
- higher computational effort

extreme case:  $n = P$

use all but one examples for training, test on single example,  
repeat P times " **leave-one-out estimate** "

### - **statistics ?**

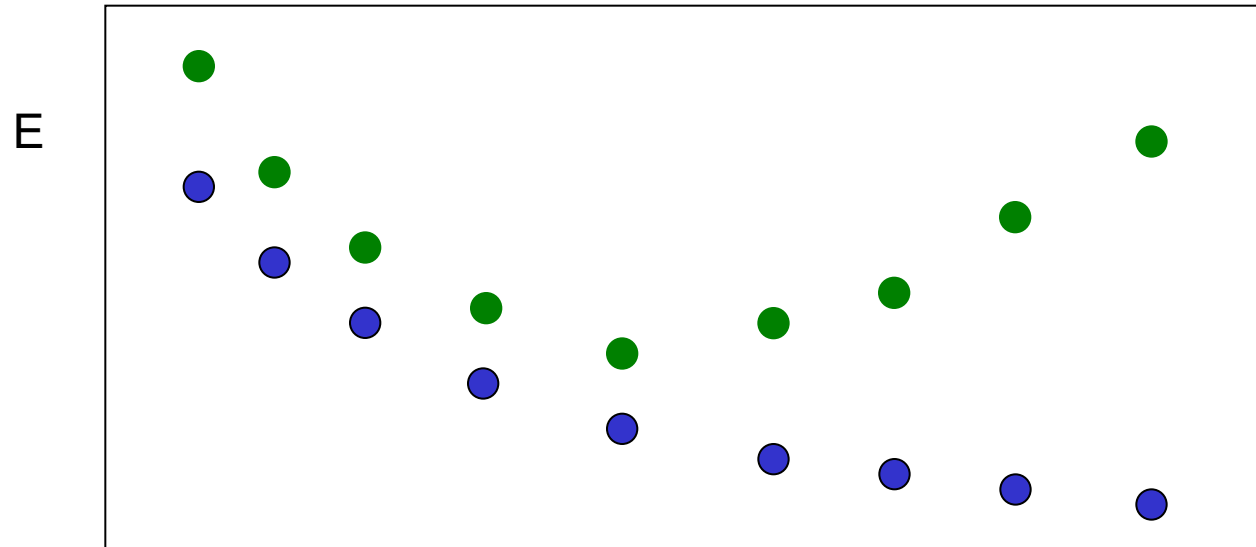
n results are not statistically independent

highly overlapping training sets! → difficult to estimate variances

with respect to training set dependence

# Over-fitting

**test / training errors** (e.g. observed in cross-validation)  
**vs. complexity of the model** (e.g. # of prototypes, neurons, ... )



in general:

$$E_{train} < E_{test}$$

"complexity" (e.g.: number of prototypes, hidden units... )

- expect: better classification ( of  $D_{train}$  ) with increasing complexity
- classifier / regression can become over-specific to training set !  
**over-fitting** ( low training, high test error )

## the bias / variance dilemma (qualitative discussion)

competing aims in training:

low **bias** = small **systematic deviation** from the "true solution"  
on average over all possible data sets of the same size

low **variance** = weak dependence on the actual training set,  
**robustness** of the hypothesis

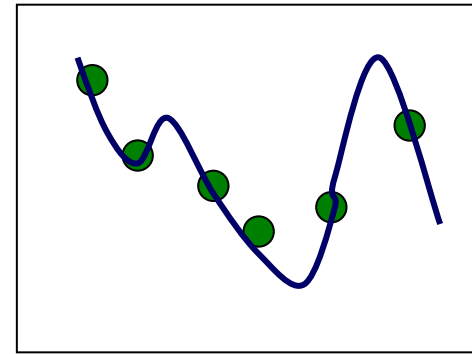
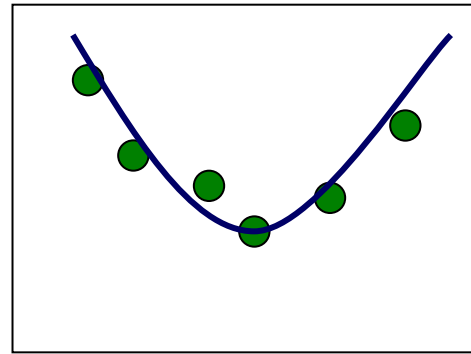
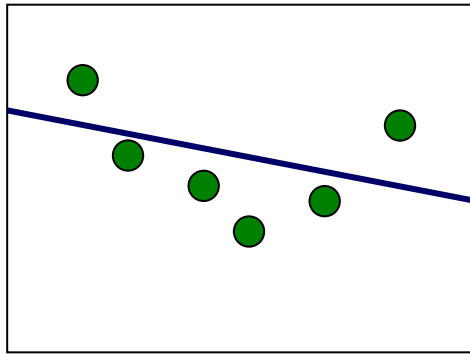
dilemma:

small variance: simple model, *under-fitting* → large bias

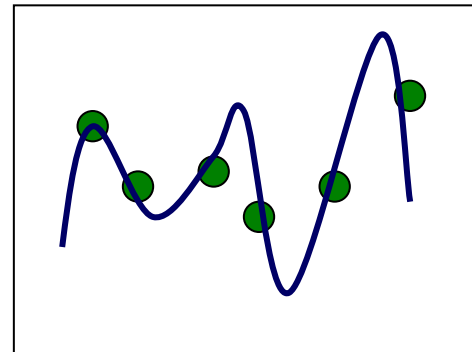
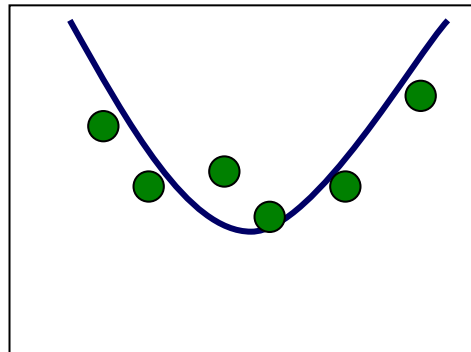
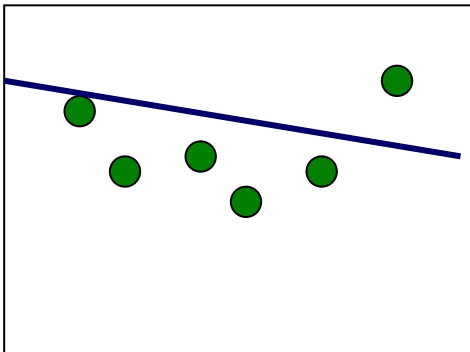
small bias: complex model, *over-fitting* → large variance

illustrative example: curve fitting to noisy data points (regression)

data set 1



data set 2



low variance  
high bias

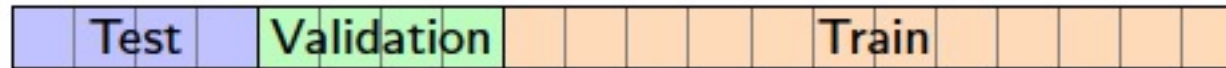
best generalization,  
matching complexity

low bias  
high variance

## Remarks

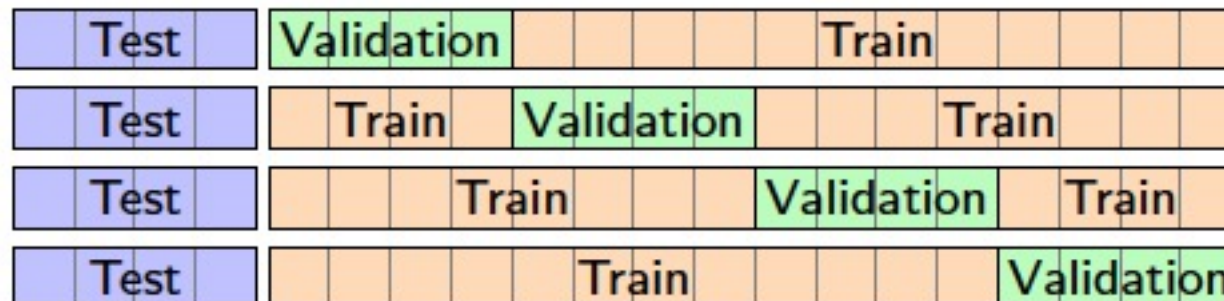
- in a potentially overfitting learning system, we can use **algorithm** parameters to control **effective complexity**  
i.e. the degree to which the training error can be minimized  
e.g. number of training epochs  
learning rates
- **validation procedures can overfit !!!**  
example: selection of parameters based on  $E_{test}$  by cross-validation  
does depend on the entire data set D  
unclear performance with respect to entirely new data
- strategies:
  - second level of validation (extra data?)
  - base selection only on training error, if possible

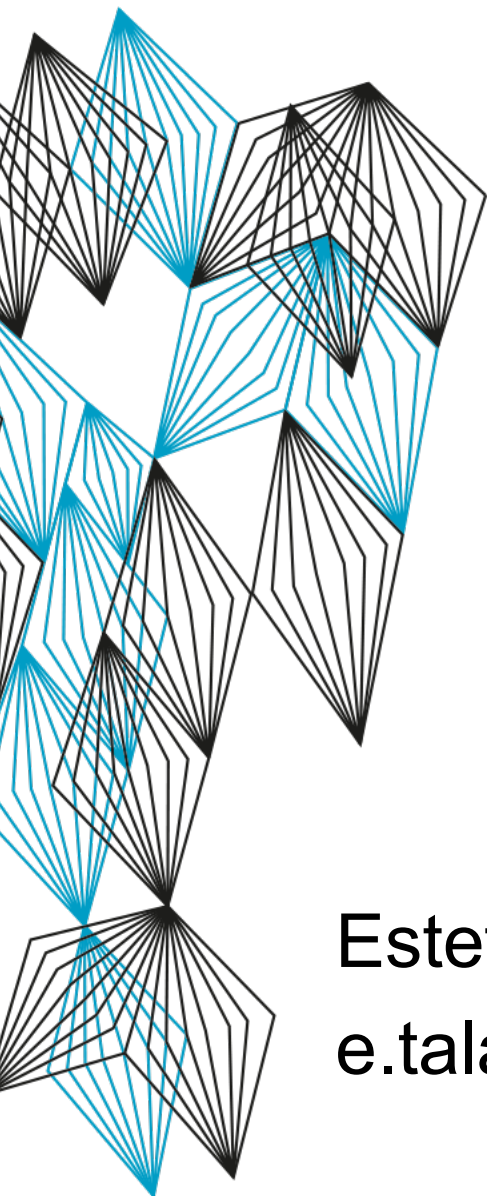
In practice, available training data is often limited  
Splitting the data in sets further reduces this



Solution: n-fold cross-validation

Repeatedly split the data and average the results (here,  $n = 4$ )





# ARTIFICIAL INTELLIGENCE & CYBER SECURITY

## INTRO TO MACHINE LEARNING REGULARIZATION

Estefanía Talavera Martínez  
e.talaveramartinez@utwente.nl