Validation procedures, over-fitting

Introduction to Intelligent Systems

Michael Biehl

validation

key problems in supervised learning

model selection: LVQ, Neural Networks, Support Vector Machine,… ?
how many prototypes, 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 $D = \left\{ \left\{ \boldsymbol{\xi}^{\mu}, S^{\mu} \right\} \right\}_{\mu=1}^{P}$

(randomly) into disjoint sets

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

→ 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 (?)

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 ?

cross-validation

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

$$D = \bigcup_{i=1}^{n} D^{(i)} \qquad D_{train}^{(i)} = D \,/\, D^{(i)} \qquad D_{test}^{(i)} = D^{(i)}$$

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_{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 ?**

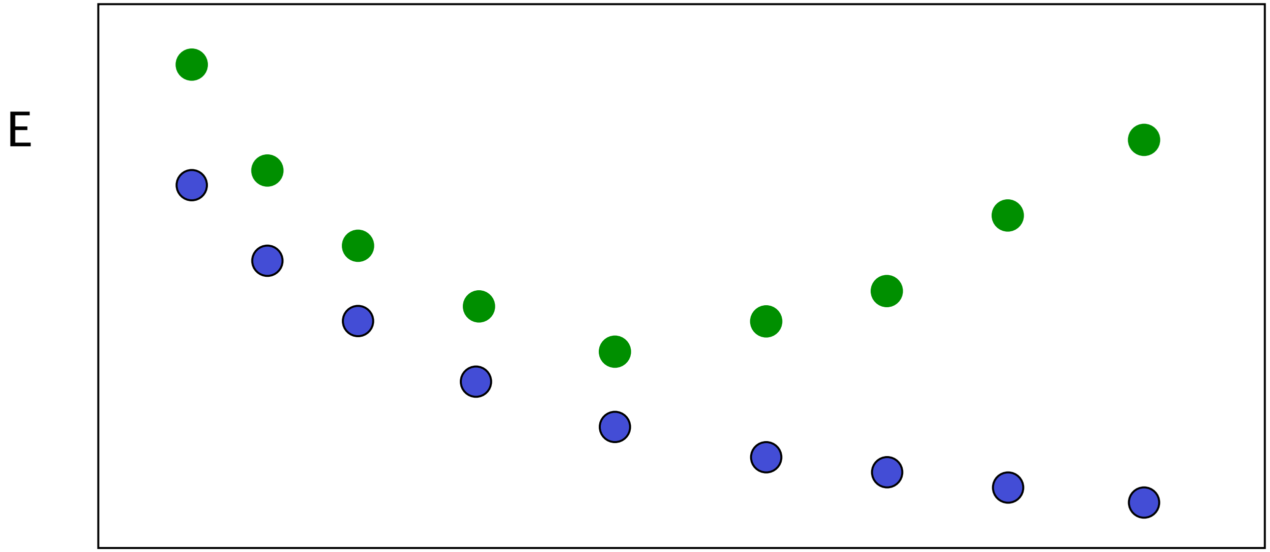
  n results are not statistically independent

  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, …)



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

in general:

$E_{train} < E_{test}$

– 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

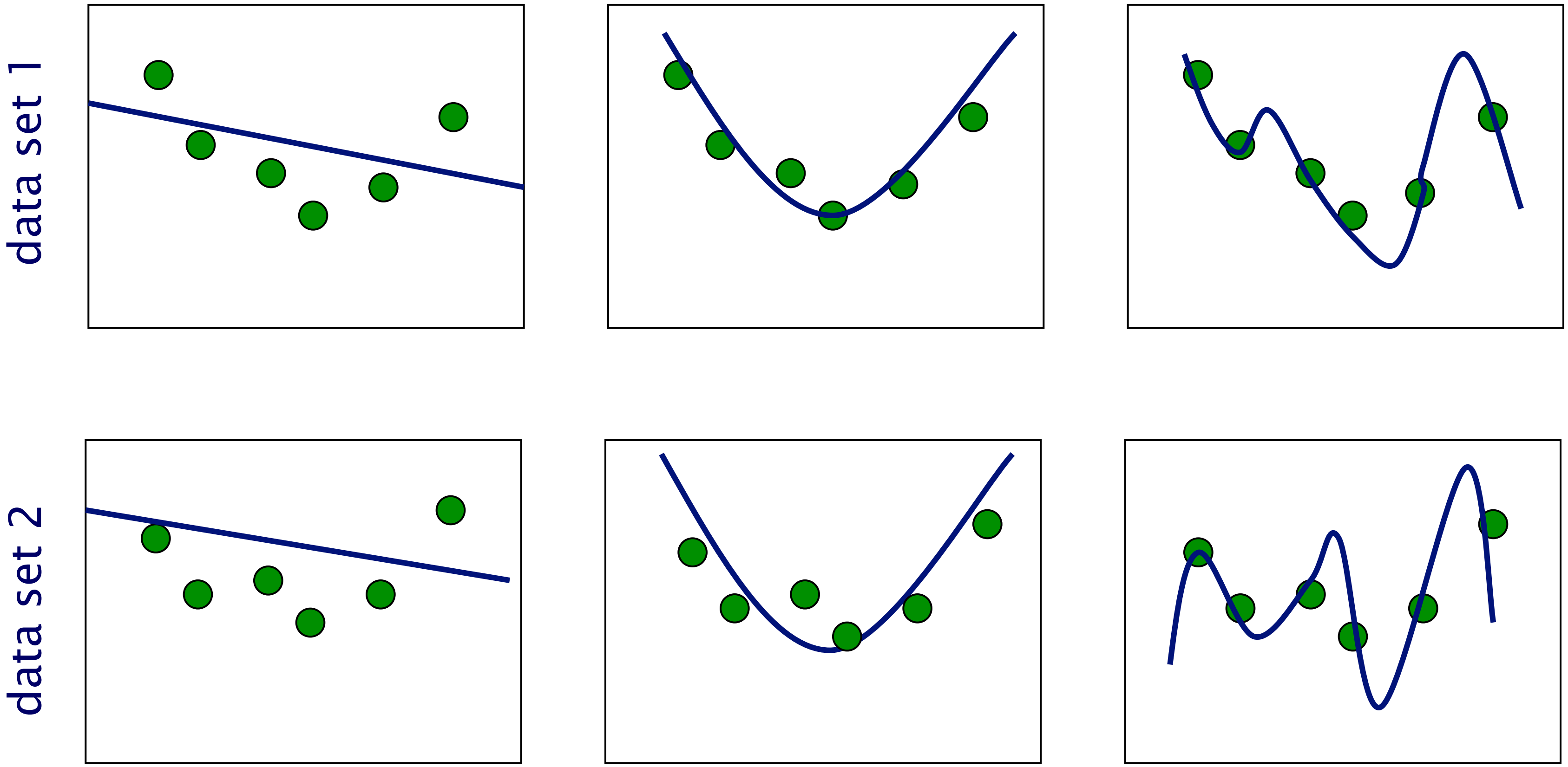
dilemma:

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

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

bias / variance

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



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

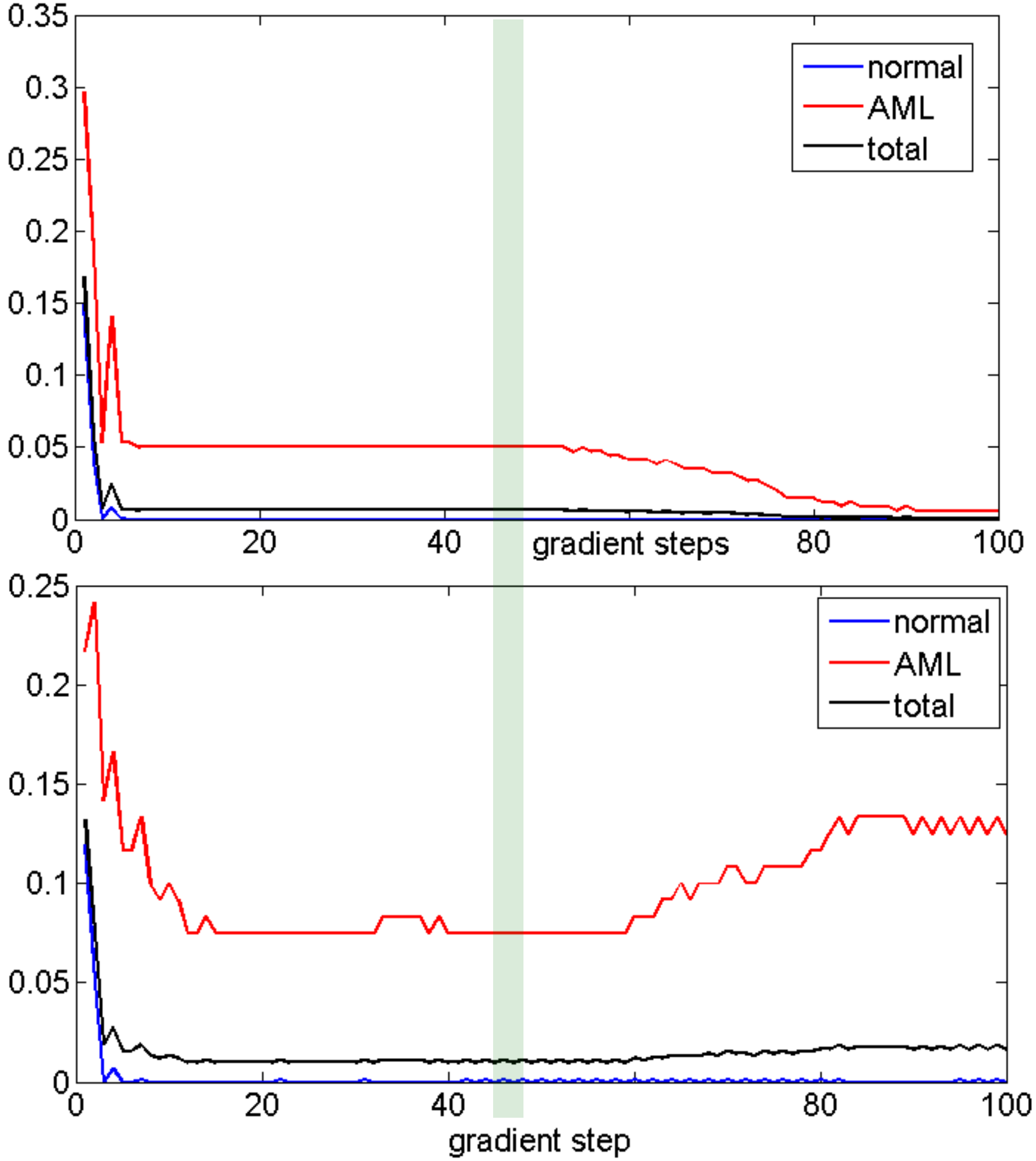
- **validation procedures can overfit !!!**
  example: selection of parameters based on E_{test} by cross-validation
  does depend on the entire data set D
  unclear performace with respect to entirely new data

  strategies: - second level of validation (extra data?)
  - base selection only on training error, if possible

real world example

training set errors

validation set errors



Leukaemia diag-nosis from flow cytometry data"

over-fitting (here: due to mislabelled example data)

"early stopping"