A Short Introduction to Machine Learning

Introduction to Machine Learning
Lect. 4

Alessio Micheli

micheli@di.unipi.it

Dipartimento di Informatica
Università di Pisa - Italy

Computational Intelligence & Machine Learning Group

Lect 4

Introduction to Machine Learning (continuation)

Introduction to Generalization in ML

Alessio Micheli

micheli@di.unipi.it

ML in a Nutshell

- **DATA**: available experience represented as vectors, structures,…

- **TASK**: supervised (classification, regression), unsupervised, …
  - E.g. Given data as labeled examples, find good approximation of the unknown *f.*

- **MODEL:**
  - describes the relationships among the data / the knowledge
  - defines the class of functions that the learning machine can implement (*hypothesis space*)

- **LEARNING ALGORITHM:**
  - (given data, task and model) the learning algorithm performs a (heuristic) *search* through a space of hypotheses that are valid in the given data
  - E.g. it adapts the free parameters of the model to the task at hand

- VALIDATION: evaluate generalization capabilities (of your hp)

ML issues

Easy use of ML tools

versus

correct/good use of ML

ML issues (I)

- Inferring general functions from know data: an *ill posed problem (e.g. in principle the solution is not unique)*
  - *With finite data we cannot expect to find the exact solution*

- Work with a restricted hypothesis space
  - see also the inductive bias concept

- What can we represent ?
- (Secondary) What can we learn ?

(as if you cannot represent a function you cannot also learn it)

ML issues (II) Generalization

- **Learning phase**: to build the model (including training)
- **Prediction phase**: evaluate the learned function over novel samples of data (generalization capability)

- Inductive learning hypothesis
  - Any *h* that approximates *f* well on training examples will also approximate *f* well on new (unseen) instances x (?)

Def

- **Overfitting**: A learner overfits the data if
  - it outputs a hypothesis *h* $(\cdot) \in$ H having true/generalization error (risk) R and empirical (training) error E, but there is another *h′*$(\cdot) \in$ H having E′>E and R′ < R (so that *h′*$(\cdot)$ is the better one, despite a worst fitting).

- Critical aspect: accuracy / performance estimation
  - Theoretical
  - Empirical (training, test) and cross-validation techniques

Complexity on a case study

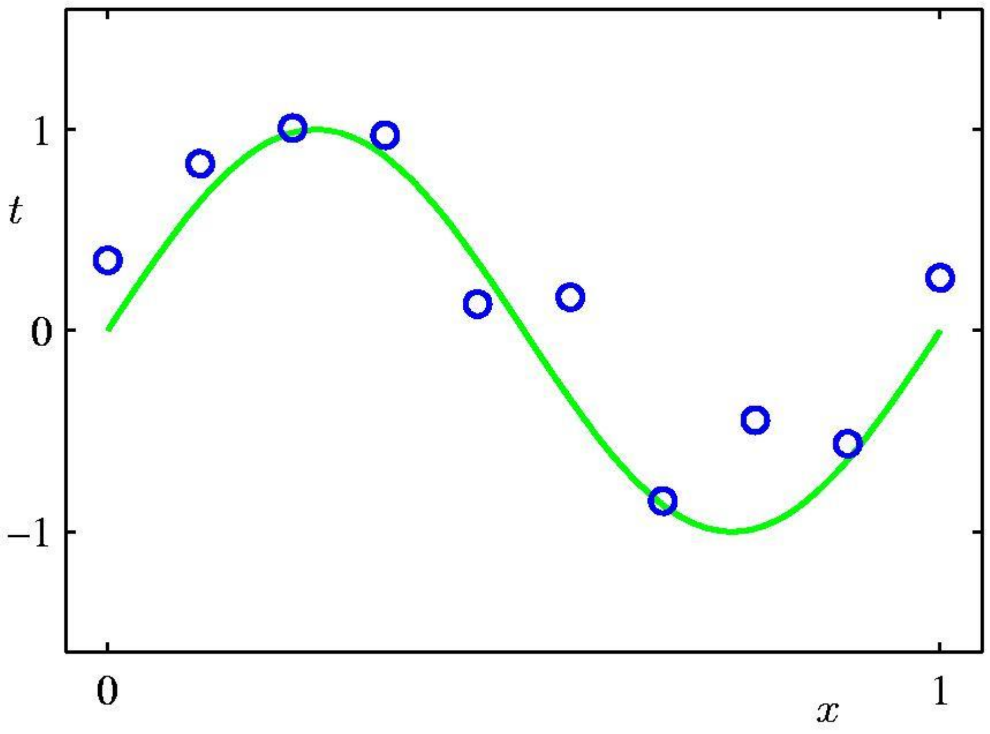
- An example on a parametric model for *regression*:

- The set of functions is assumed as polynomials with degree M
- The **complexity** of the hypothesis increases with the degree M
- l = number of examples

- Warning: This is an artificial simplified task (unrealistic due to the use of just 1 input variable, the fact that we know the target function in advance, …)

Polynomial Curve Fitting

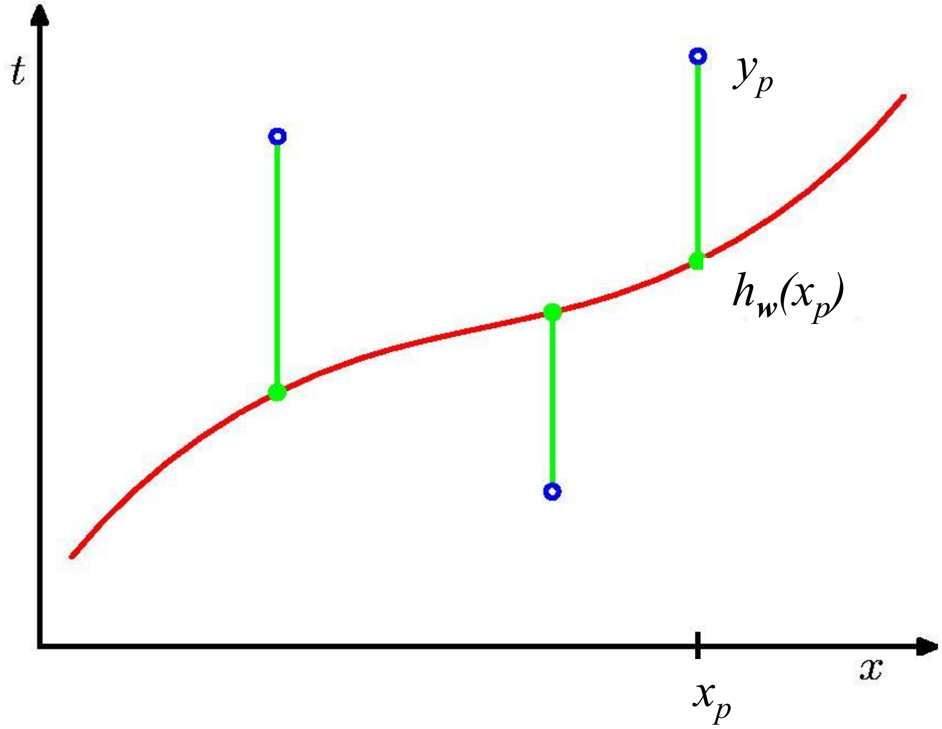
Target = sin(2*pi*x) + random noise (gaussian)



$$h_{\mathbf{w}}(x) = w_0 + w_1 x + w_2 x^2 + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j$$

Samples affected by noise (not always on the green “true” line)

Sum-of-Squares Error Function

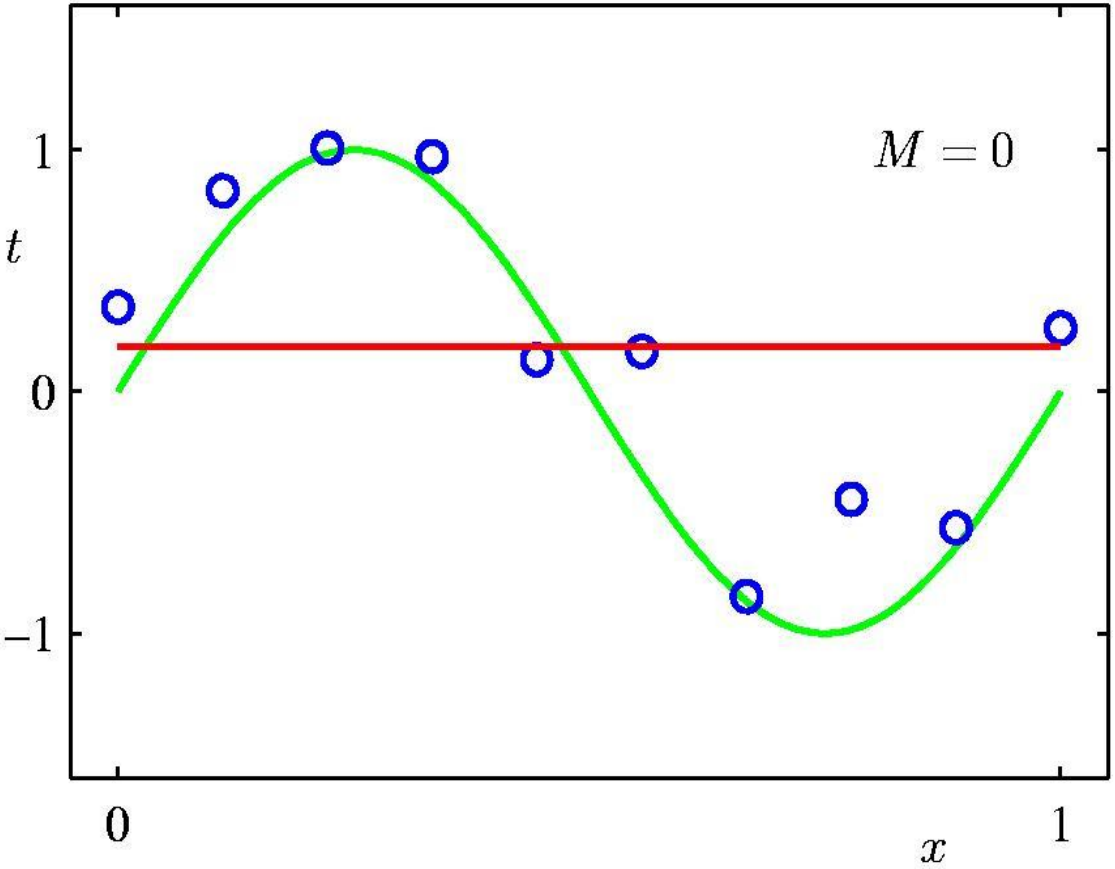


$$E(\boldsymbol{w}) = \sum_{p=1}^{l} (y_p - h_{\mathbf{w}}(x_p))^2$$

Note: p is the example, y_p the target for p
l the total number of examples
$h_{\mathbf{w}}(x_p)$ is the model output at the point x_p
(x is a single variable, n=1)

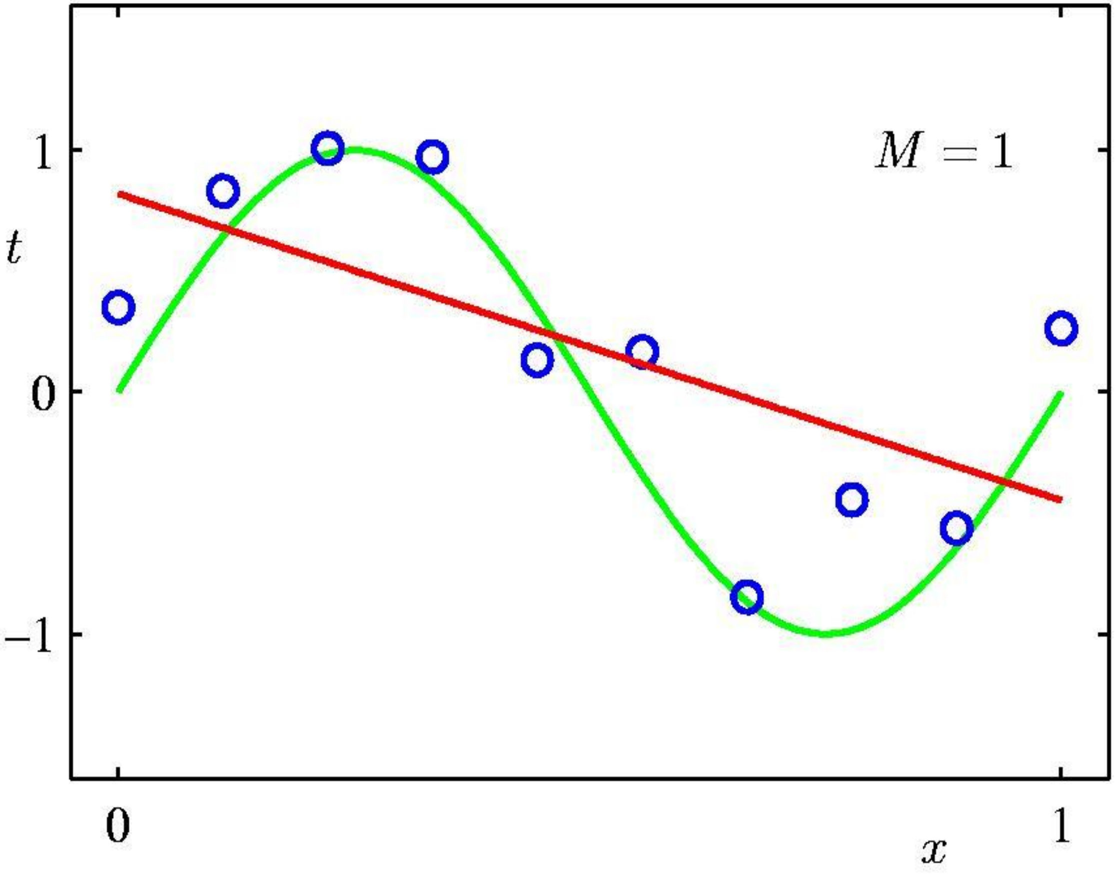
Minimize $E(\boldsymbol{w})$ (Square Error) to find the best $\boldsymbol{w}$ (fitting)

0^{th} Order Polynomial



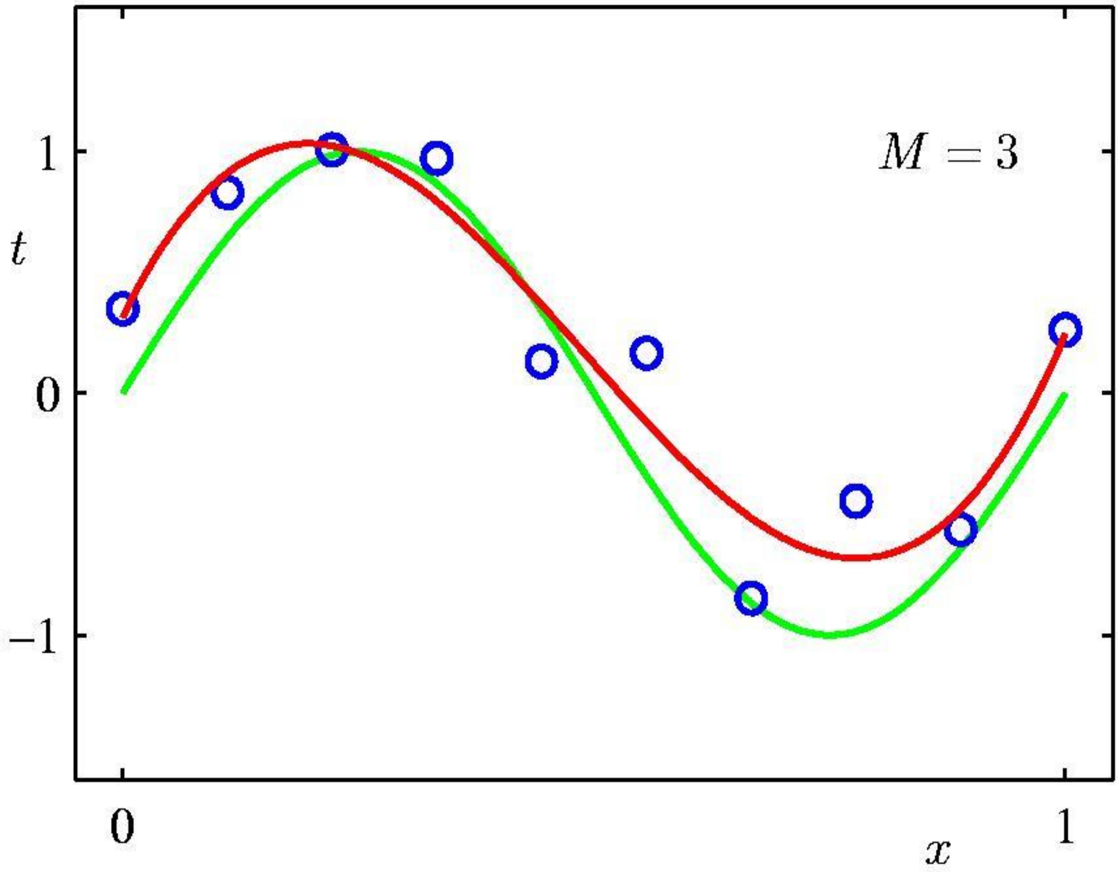
Underfitting: too simple model (red line)
w.r.t. to the target function

1st Order Polynomial



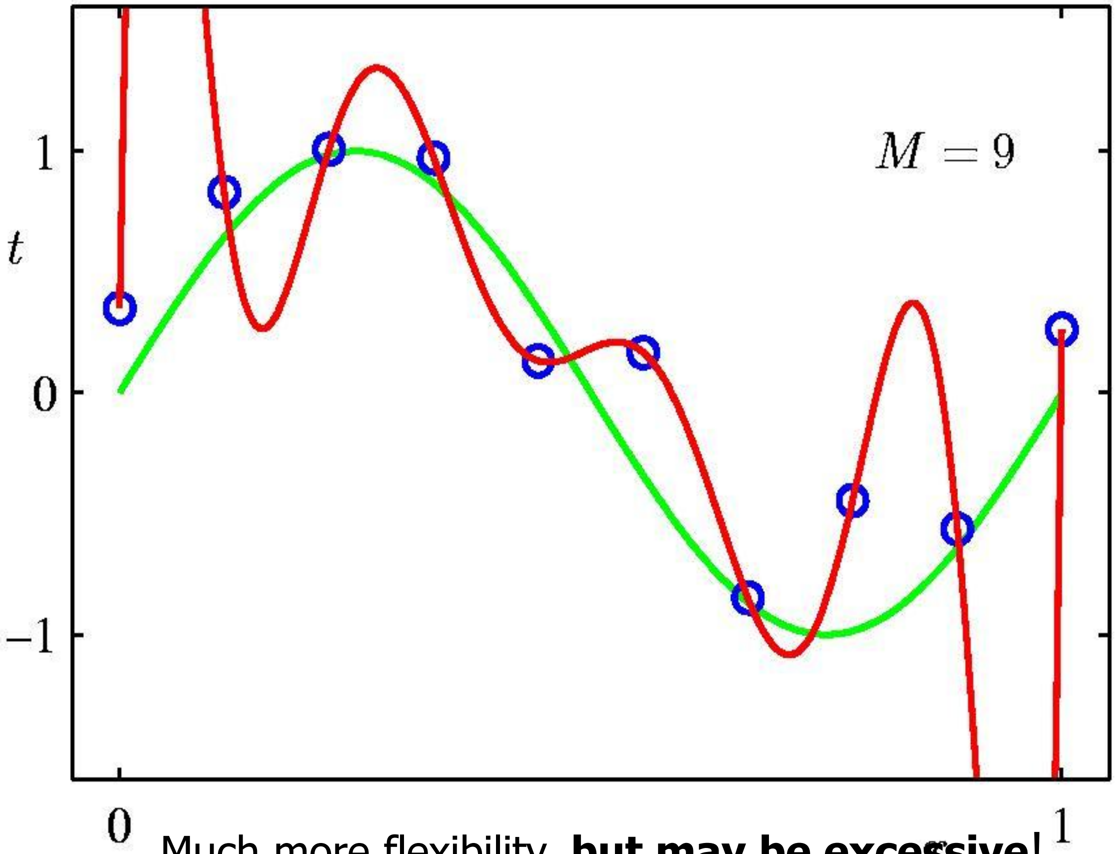
Still poor solution (due to **underfitting**)

3^{rd} Order Polynomial



More **flexibility** is useful !!!

9th Order Polynomial



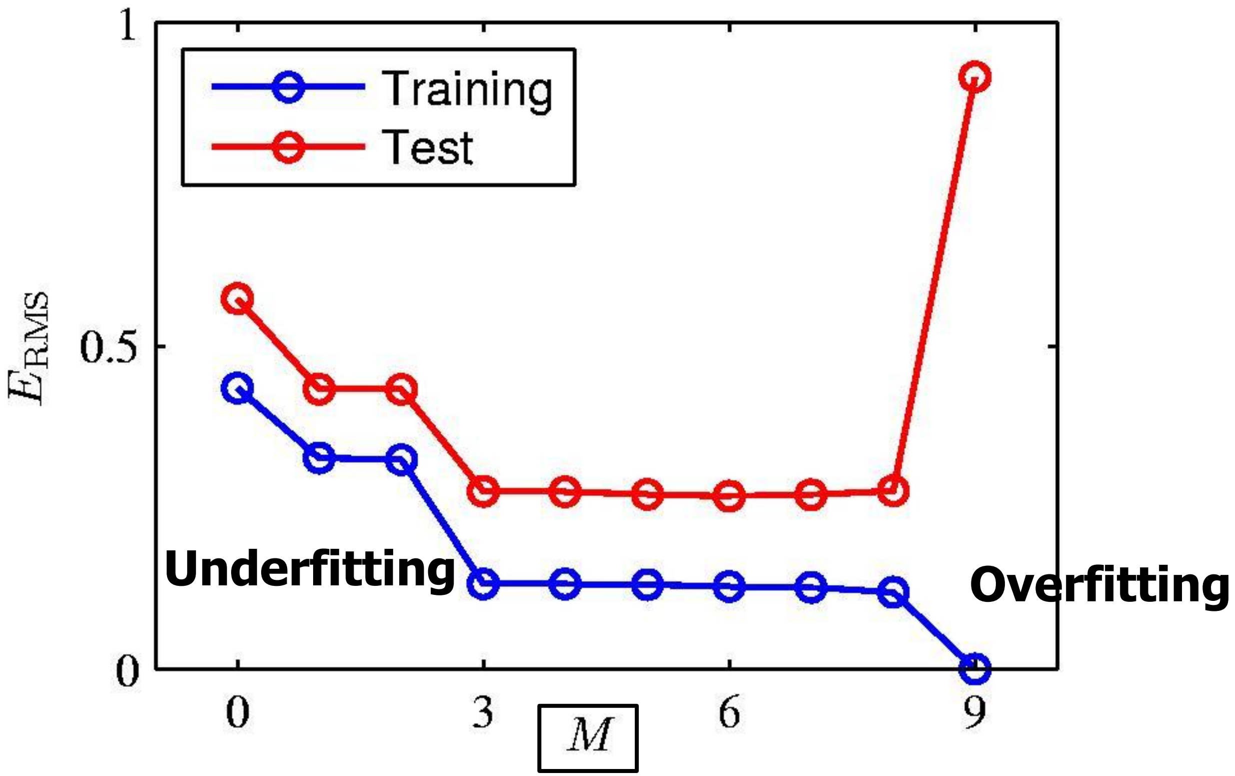
Much more flexibility, **but may be excessive**!

E(w)= 0 on training data!!! But error on test set ?

Too complex model (in this case it fits even the noise)!

Poor representation of the (green) true function (due to **overfitting**)

Underfitting and Overfitting with the complexity (*M*)



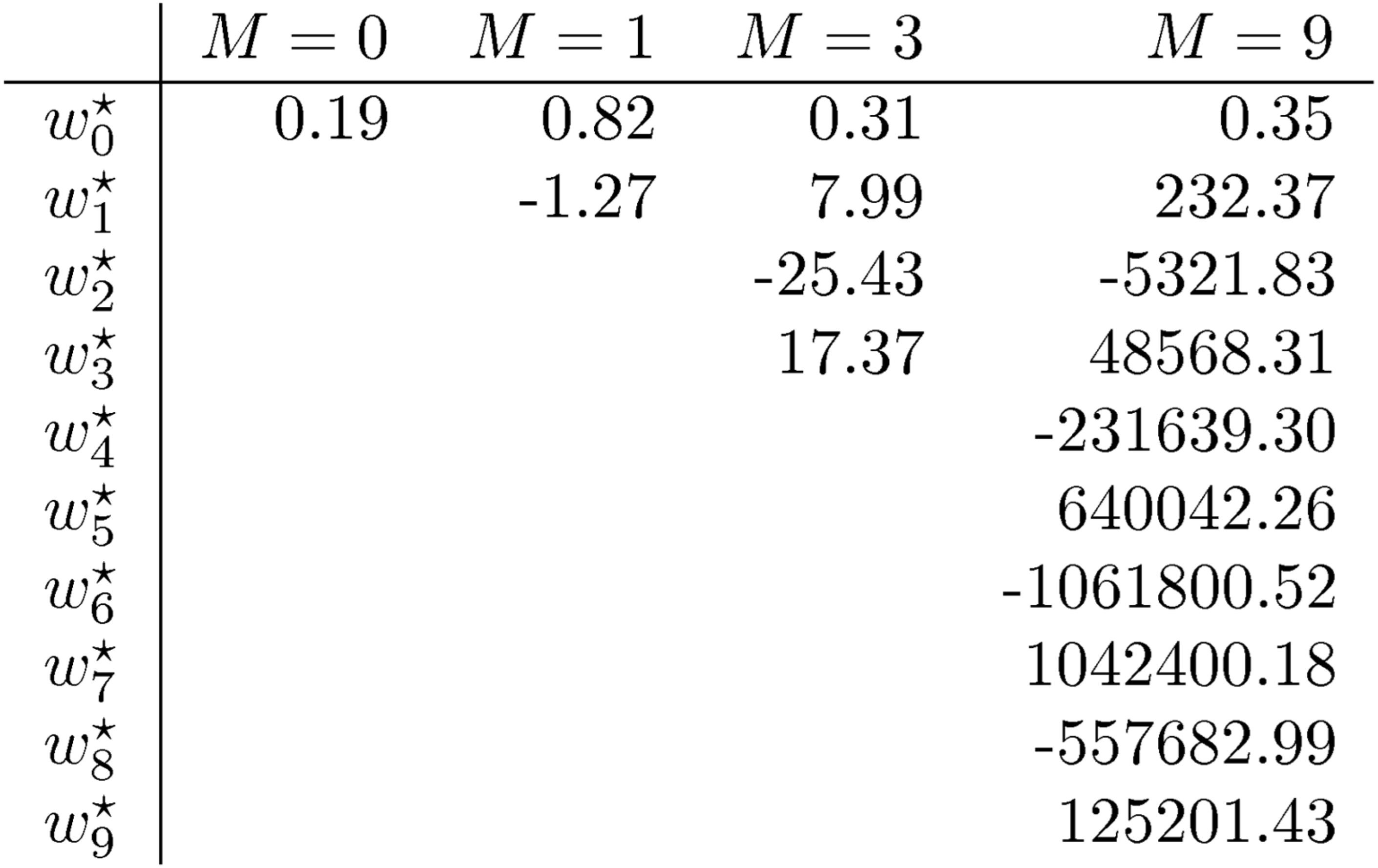
Underfitting

Overfitting

Root-Mean-Square (RMS) Error: $E_{\mathrm{RMS}} = \sqrt{2E(\mathbf{w}^\star)/l}$

Where $E(\mathbf{w}^*)$ is the error for the trained model

Polynomial Coefficients

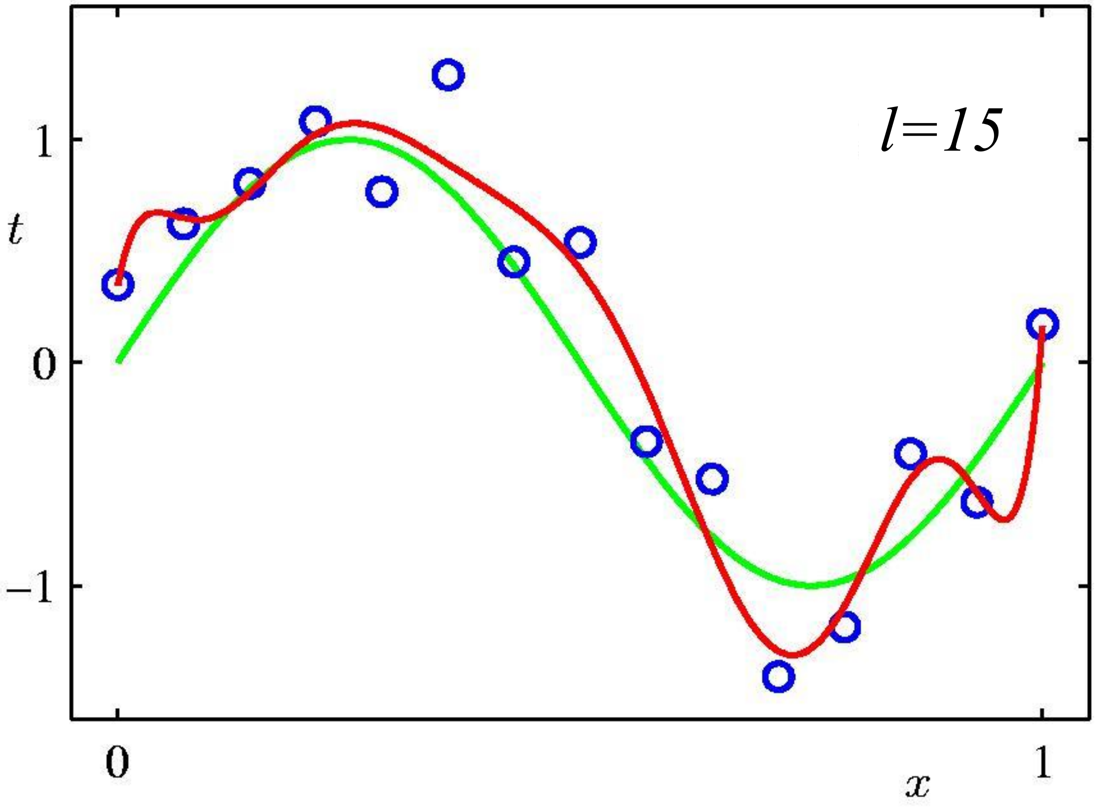


	$M = 0$	$M = 1$	$M = 3$	$M = 9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

Data Set Size: *l=15*

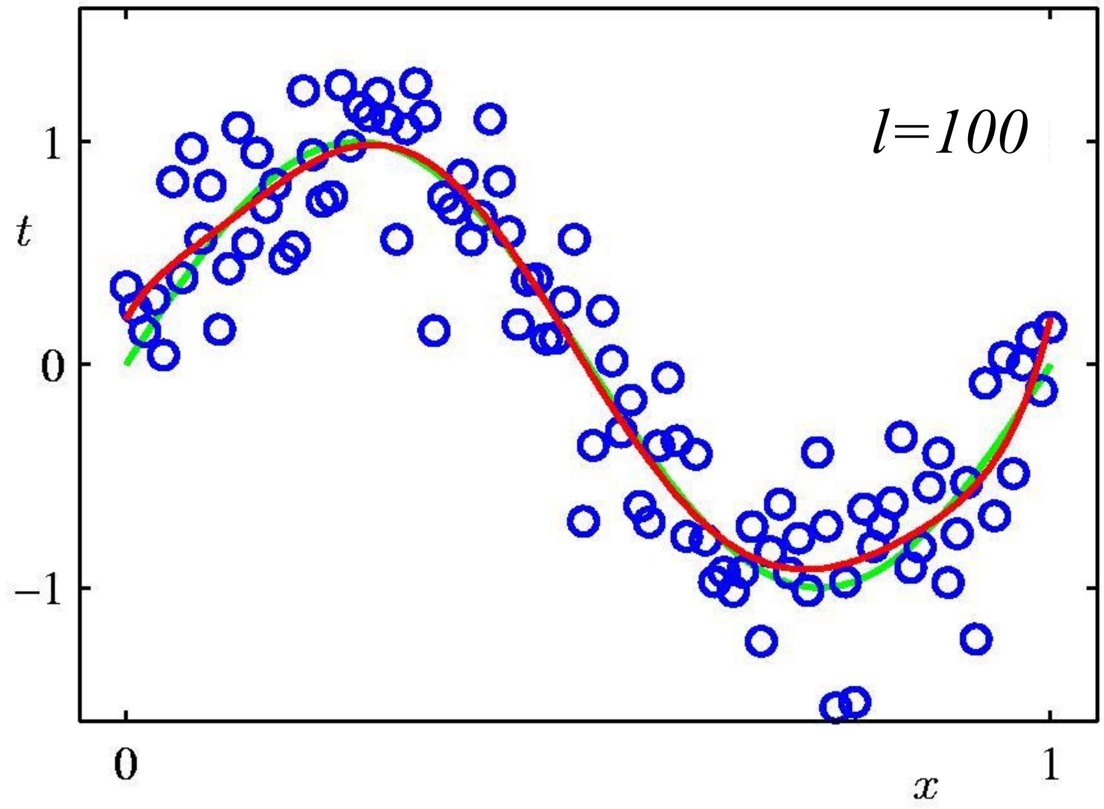
previous was 10

9th Order Polynomial



Data Set Size: *l=100*

9th Order Polynomial (even more data)



We can use higher *M* with a higher number of data

Toward SLT

Putting all together:

- We want to investigate on the *generalization* capability of a model (measured as a risk or test error)
  - with respect to the training error
  - overfitting and underfitting zones

- The role of model complexity
- The role of the number of data

- *Statistical Learning Theory (SLT):* a general theory relating such topics

(Simplified) Formal Setting

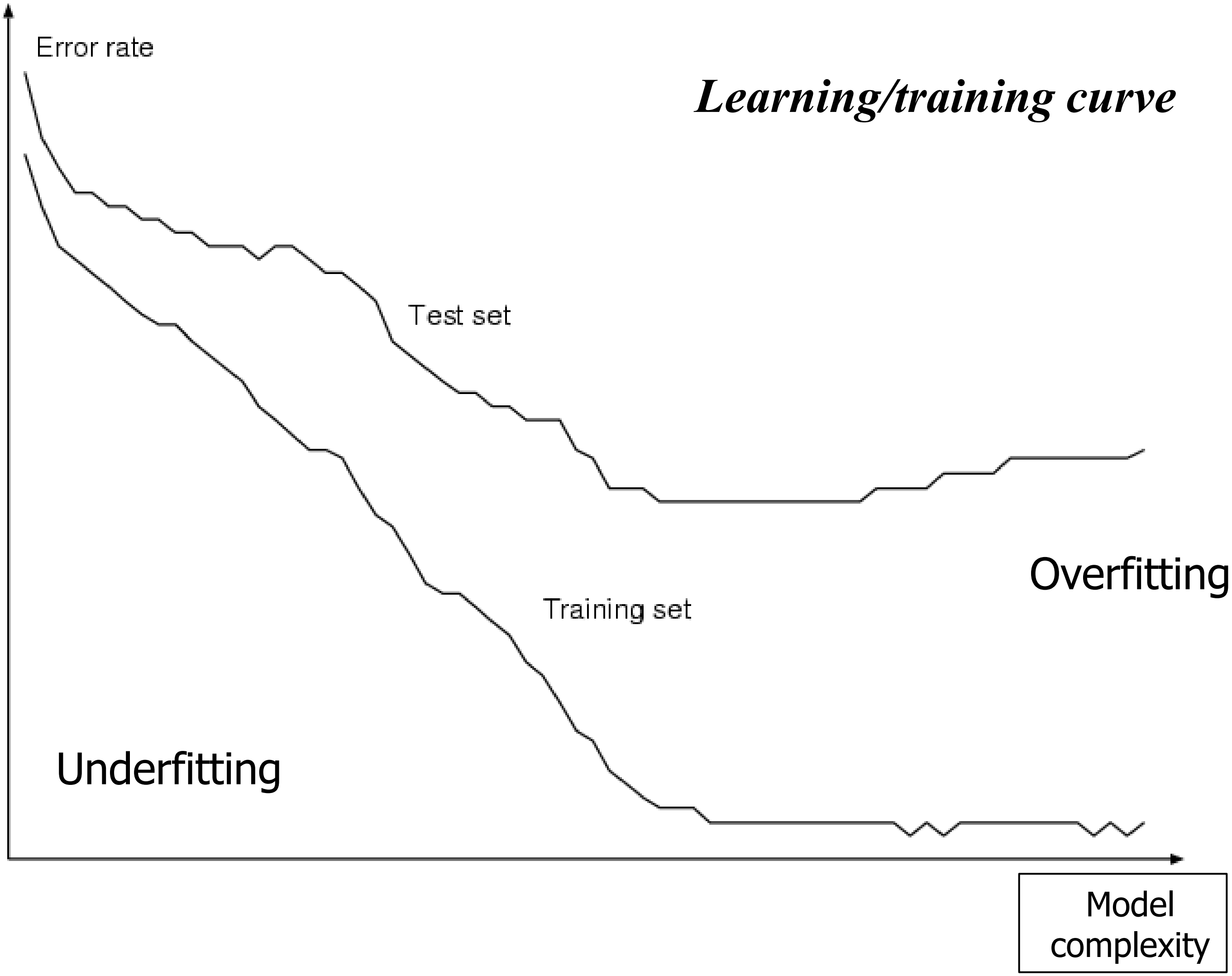
- Approximate unknown *f*($\boldsymbol{x}$), *d is the target (d=true f +noise)*
- Minimize *risk function* $R = \int L(d, h(\boldsymbol{x}))dP(\boldsymbol{x}, d)$ True Error Over *all* the data
- Given
  - value from teacher (d) and the probability distribution $P(\boldsymbol{x}, d)$
  - a loss (or cost) function, e.g. $L(h(\boldsymbol{x}), d) = (d - h(\boldsymbol{x}))^2$

- Search h in H : Min R
- But we have only the finite data set $TR = (\boldsymbol{x}_{p,} d_p), \quad p = 1 \ldots l$
- To search h: minimize empirical risk (training error E), finding the best values for the model free parameters

$$R_{emp} = R_{emp}(h, TR) = \frac{1}{l}\sum_{p=1}^{l}(d_p - h(\boldsymbol{x}_p))^2$$

- Empirical Risk Minimization (ERM) Inductive Principle
- *Can we use R_{emp} to approximate R?*

Typical behavior of learning



Vapnik-Chervonenkis-dim and SLT: a general theory (I)

- Given the *VC-dim* (VC), a measure *complexity* of *H (flexibility to fit data)* (e.g. Num. of parameters for linear models/polynomials)

Repetita: Can we use R_{emp} to approximate R?

Very important!

Def

- *VC-bounds in the form:* it holds with probability 1-δ that

guaranteed risk

$$R \boxed{\leq} \underline{R_{emp}} + \varepsilon(1/l, VC, 1/\delta)$$

VC-confidence

- First (basic) explanation, ingredients:
  - ε is a function that grows with VC *(VC-dim)*, and that decreases with (higher) l and δ (i.e. l and δ are in the denominator)
  - We know that R_{emp} decreases using complex models (with high *VC-dim*) (e.g. the polynomial degree in the example)
  - δ is the confidence, it rules the probability that the bound holds (e.g. low delta 0.01 → the bound holds with probability 0.99)

- Now we can see how it can "explain" the *underfitting* and *overfitting* and the aspects that control them.

Vapnik-Chervonenkis-dim and SLT: a general theory (II)

Very important!

Comments:

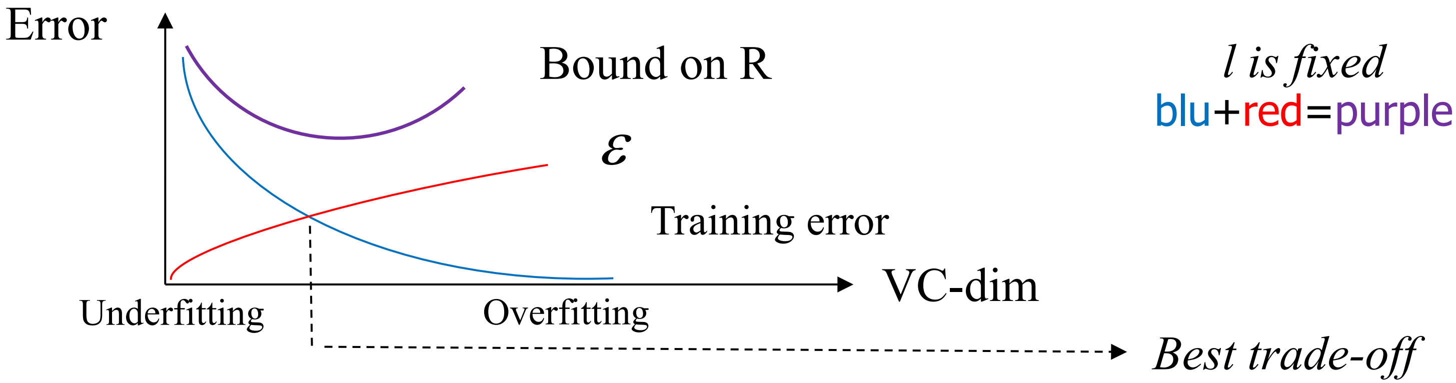
- *VC-bounds in the form:* it holds with probability 1-δ that

guaranteed risk $$R \boxed{\leq} \underline{R_{emp}} + \underbrace{\varepsilon}_{VC\text{-}confidence}(1/l, VC, 1/\delta)$$

Intuition:

- Higher l (data) → lower *VC-confidence* and a bound close to R
- Too simple models (low VC-dim) can be not suff. due to high R_{emp} (underfitting)
- Higher VC-dim (fix l) → lower R_{emp} but *VC-conf.*, and hence R, may increase (overfitting)

- *Structural risk minimization*: minimize the bound !



l is fixed
blu+red=purple

- Concept of control of the model complexity (flexibility): trade-off between model complexity (VC-dim) and TR accuracy (fitting)

An example

It is possible to derive an upper bound of the ideal error which is valid with probability (1-delta), delta being arbitrarily small, of the form:

- General: $R \leq R_{emp} + \varepsilon(1/l, VC, 1/\delta)$
- Example: $R \leq R_{emp} + \varepsilon(VC/l, -\ln(\delta/l))$
- There are different bounds formulations according to different classes of f, of tasks, etc.
- More in general, in other words (simplifying): we can make a good approximation of f from examples, provided we have a good number of data, and the complexity of the model is suitable for the task at hand.
  - Fit data as much as possible to avoid underfitting (high R_{emp}), but not too much so to avoid overfitting (due to the increase of *VC-confidence* term)
  - (Double descent/over-parametrization/benign overfit phenomena will be discussed later)

Discussion
Complexity control

- SLT - Statistical Learning Theory:
  - It allows formal framing of the problem of generalization and overfitting, providing analytic upper-bound to the risk *R* for the prediction over all the data, regardless to the type of learning algorithm or details of the model
  - *The ML is well founded*: the Learning risk can be analytically limited and only few concepts are fundamentals !
  - It leads to new models (SVM) (and other methods that directly consider the control of the complexity in the construction of the model)
  - It bases one of the inductive principles on the control of the complexity
  - It explains the main difference with respect to supporting methods from CM (providing the techniques to perform fitting), apart from modelling aspects

Open questions:

- What (other) principles are to found the control of the complexity? How to work in practice?
  - How to measure the complexity (or fitting flexibility)?
  - How find the best trade-off between fitting and model complexity?

Exercises

- Reinterpretation of some parts in the first lectures: Why a zero error for the training does not necessarily imply a good solution?
- Connect again by your-self the underfitting and overfitting to the SLT inequality interpretation
- Connect again by your-self to the example with polynomials
- Looking at the def. of overfitting, denote the *h* and *h'* on the plot of the SLT bound.
- Is the presence of noise the cause of overfitting (or the complexity trade-off)? Can you image an example, like the overfitting with a polynomial with high degree, were even with completely cleaned data (no noise) you have overfitting?

Validation

- Evaluation of performances for ML systems =
  Generalization/Predictive accuracy evaluation

- "*The performance on training data provide an overoptimistic evaluation*"

- Validation !
- Validation !!
- Validation !!!

- ➢ In the following: an introduction, → 🏃
- ➢ Validation will be the topic of specific lectures later, in the "***Validation & SLT***" part of the course
- ➢ And it has a *central role* for the applications and the project

Validation: Two aims

Very important!

- After models training on the training set
- ***Model selection:*** estimating the performance (*generalization error*) of different learning models in order to choose the best one (to generalize).
  - this includes search the best *hyper-parameters* of your model (e.g. polynomial order, …).

  It returns a model

- ***Model assessment:*** having chosen a final model, estimating/evaluating its prediction error/ risk (*generalization error*) on new *test* data (measure of the quality/performance of the ultimately chosen model).

  It returns an estimation

Gold rule: Keep separation between goals and use separate data sets

Validation: ideal world

- A large *training set* (to find the best hypothesis, see the theory)
- A large validation set for model selection
- A very large *external* unseen data *test set*

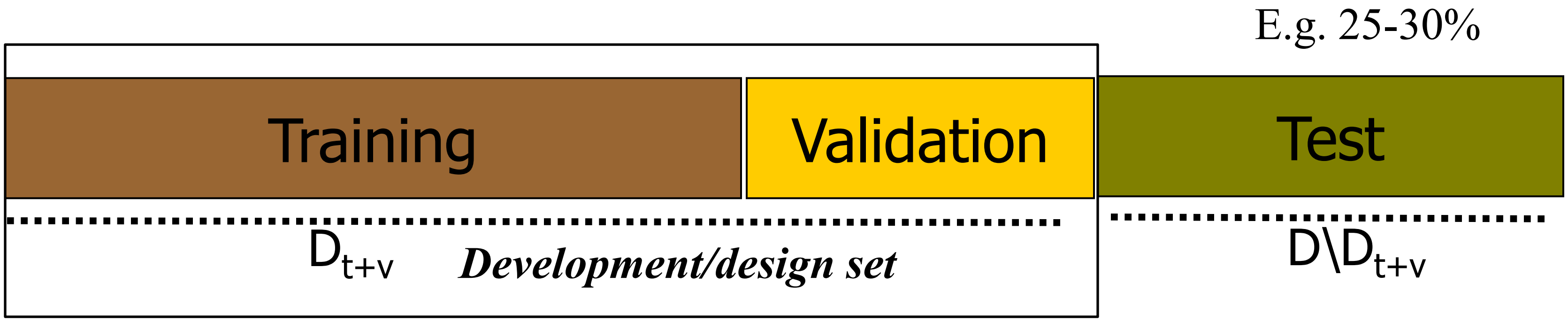
- *With finite and often small data sets?*
- *….Just estimation of the generalization performance*

- *We anticipate to basic techniques:*
  - *Simple hold-out (basic setting)*
  - *K-fold Cross Validation (just an hint in this lecture)*

Hold out cross validation

Hold out: basic setting

- Partition data set D into *training set* (TR), *validation or selection set* (VL) *and test* set (TS)
  - All the three sets are disjoint sets !!!
  - **TR** is used to run the **training** algorithm
  - **VL** can be used to **select the best model** (e.g hyper-parameters tuning)
  - **Test** set (result) is *not* to be used for tuning/selecting the best h: it is only for **model assessment**



E.g. 25-30%

Training

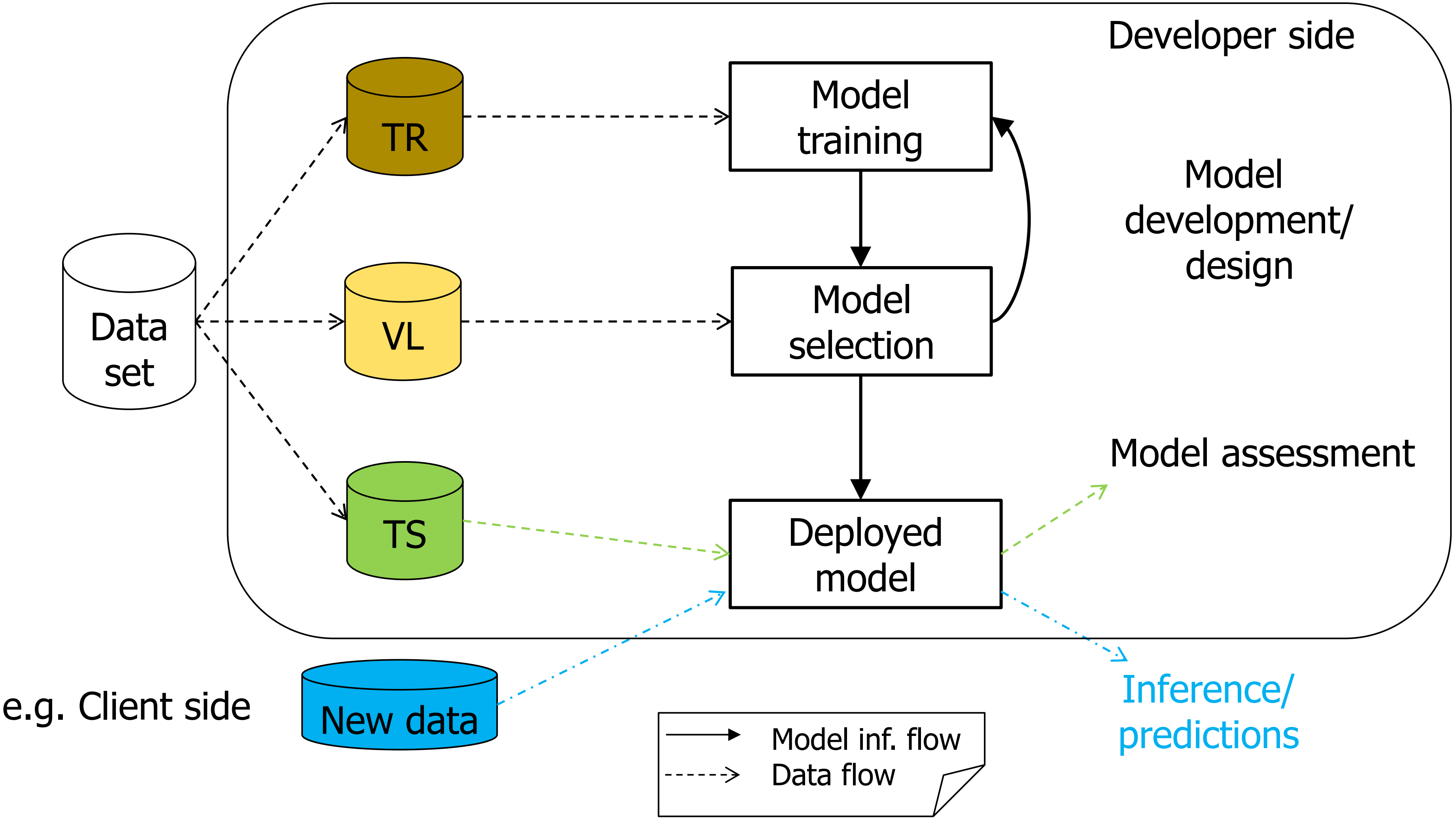
Validation

Test

D_{t+v} ***Development/design set***

$D\backslash D_{t+v}$

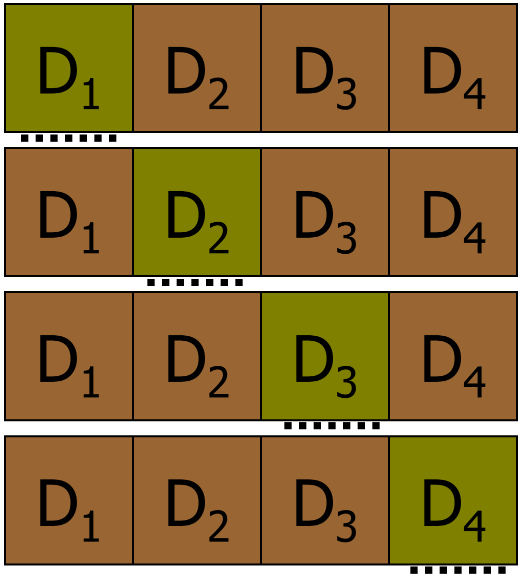
TR/VL/TS by a schema



Hold out and **K-fold** cross validation

preview

Hold out CV *can make insufficient use of data*



D_1 D_2 D_3 D_4

D_1 D_2 D_3 D_4

D_1 D_2 D_3 D_4

D_1 D_2 D_3 D_4

K-fold Cross-Validation

(we will see later in a specific lecture)

- Split the data set D into k mutually exclusive subsets $D_1, D_2, \ldots, D_K$
- Train the learning algorithm on $D \backslash D_i$ and test it on D_i
- Can be applied for both VL or TS splitting
- *It uses all the data for training and validation or testing*

Issues:

- How many folds? 3-fold, 5-fold , 10-fold, …., 1-leave-out
- Often computationally very expensive
- Combinable with validation set, double-K-fold CV, ….

Classification Accuracy

Def Confusion matrix

Predicted Actual	Positive	Negative
Positive	TP	FN
Negative	FP	TN

false positive (FP) :eqv. with false alarm

Specificity = TN / (FP + TN)

(True Negative rate =1 -FPR)

Sensitivity = TP / (TP + FN)

(True Positive rate or Recall)

(Precison= TP/(TP+FP))

Accuracy: % of correctly classified patterns = TP +TN / total

Note: for binary classif.: 50% correctly classified = “coin” (random guess) predictor!

Other topics:

- unbalanced data (e.g. 99% +) → trivial classifier exists ,
- ….

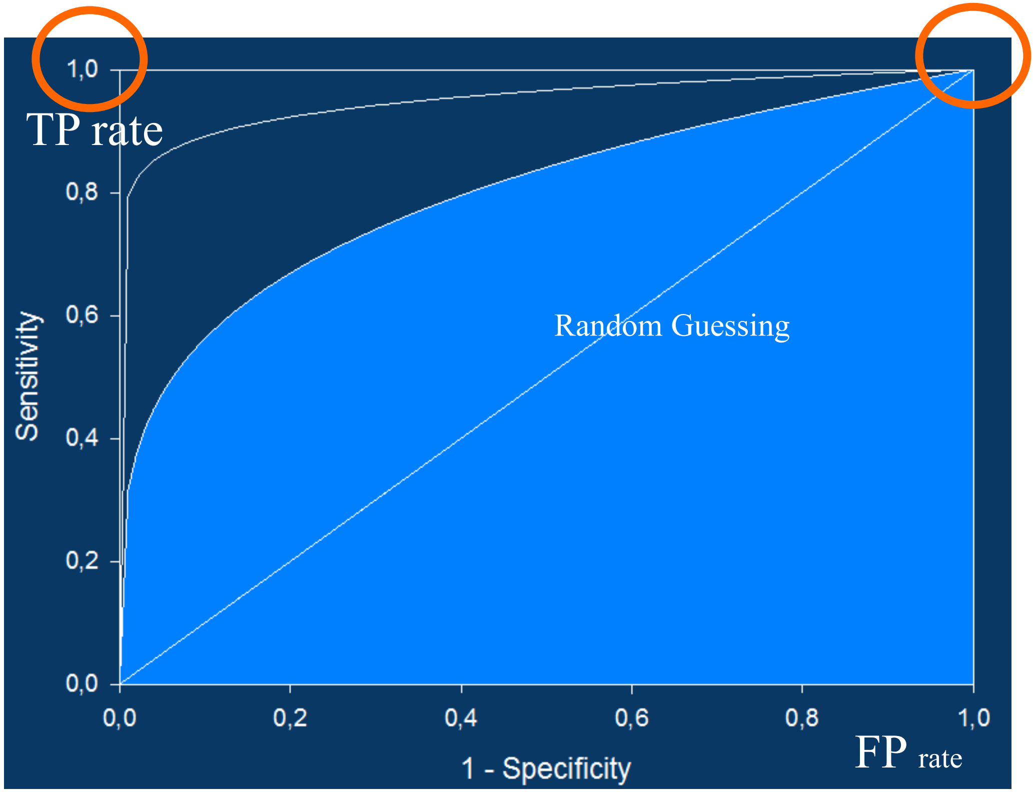
ROC curve

Confusion matrix

Predicted Actual	Positive	Negative
Positive	TP	FN
Negative	FP	TN

Specificity = TN / (FP + TN)

TPr or Sensitivity = TP / (TP + FN)

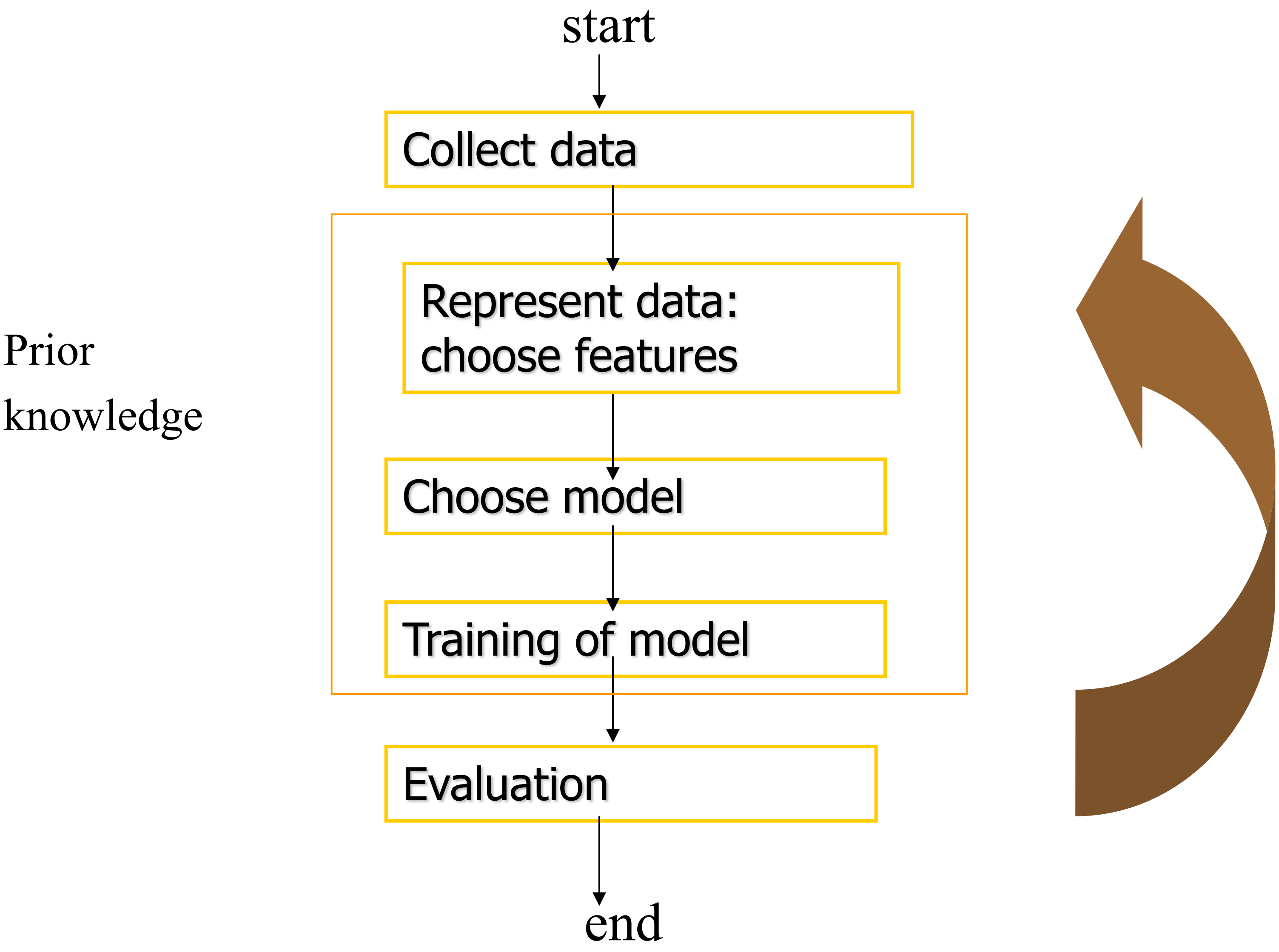


ROC curve

The diagonal corresponds to the worst classificator.

- Better curves have higher AUC (Area Under the Curve).

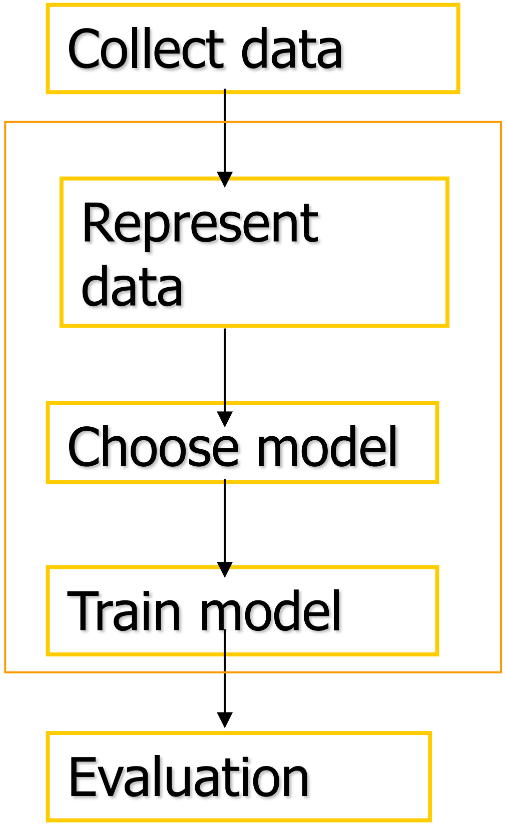
The Design Cycle



Design cycle (I)

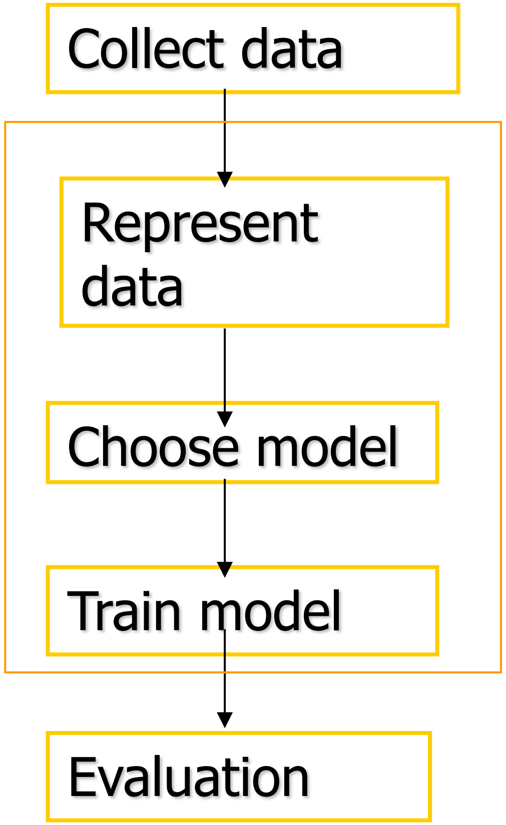
- Data collection:
  - Selection, integration, data cleaning etc.
  - Adequately large and representative set of examples for training and test
- Data representation
  - Domain dependent, exploit prior knowledge of the application expert
  - Feature selection
  - Outliers detection
  - Other preprocessing: variable scaling, missing data,..

  Often the most critical phase for an overall success!
- Model choice:
  - Statement of the problem
  - Hypothesis formulation
    - You must know the limits of applicability of your model
  - complexity control



Design cycle (II)

- Building of the model (core of ML):
  - through the learning algorithm using the training data
- Evaluation:
  - Performance = predictive accuracy !
  - Also *interpreation* of model outcomes and *explanation* of the results
  - "knowledge" extraction
- Deployment…



Misinterpretations

For every statistical models (including DM applications)

- Causality is (often) assumed and a set of data representative of the phenomena is needed.
  - Not for unrelated variables and for random phenomena (lotteries)
  - Uninformative input variables → poor modeling → Poor learning results
- Causality cannot be inferred from data analysis alone:
  - People in Florida are older(on av.) than in other US states.
  - Florida climate causes people to live longer ?
- May be there is a statistical dependencies for reasons outside the data

- More specifically for ML:
- Powerful models (even for "garbage" data) → higher risk !
- Not-well validated results: the predicted outcome and the interpretation can be misleading.

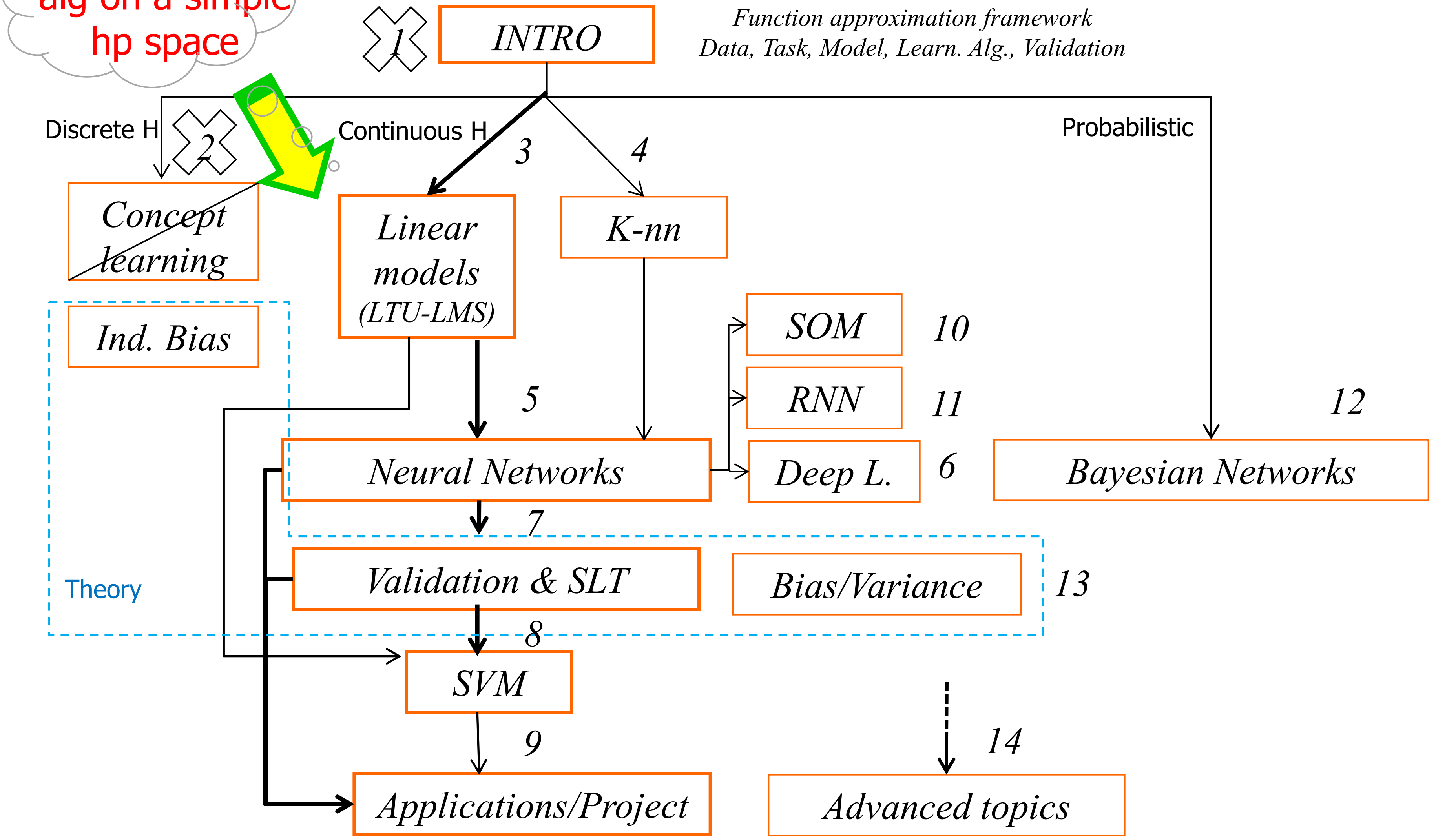
Bibliographic references (lect 1-2-3-4)

- **Course notes (slides copy): lectures 1-4** without specific textbook materials
  - ***Readings :*** Mitchell. *The discipline of Machine Learning.* July 2006. CMU-ML-06-108
  - And other in: *http://www.di.unipi.it/~micheli/DID/*
- *On the textbook:*
  - S. **Haykin**: *Neural Networks: a comprehensive foundation*, IEEE Press, 1998. (2nd. Ed.): **sez.1.7** (knowledge rep)
  - **(3rd ed): sez.1.7** (knowledge rep), **1.8, 1.9** (learning processes and tasks)
  - T. M. **Mitchell**, *Machine learning*, McGraw-Hill, 1997: **cap 1 and 2.**
- *Other references:*
  - Russell, Norvig: Intelligenza artificiale (AIMA), 2005 (in vol. 2)
    - E.g. background appendix: http://aima.cs.berkeley.edu/newchapa.pdf
  - Hastie, Tibshirani, Friedman, The Elements of Statistical Learning, Springer Verlag, 2001 (esiste New Ed.): **cap 1** e sez. 7.10
  - Cherkassky, Mulier, Learning from data : concepts, theory, and methods, Wiley, 1998 (esiste New Ed.): **cap1 e sez.2.1** (loss e tasks)
  - C.M. Bishop, Pattern Recognition and Machine Learning, Springer 2006: **sez. 1.1** (polynomial fitting example)
  - Duda, Hart, Stork, Pattern Classification, 2nd. ed. J. Wiley & Sons, 2001: **cap1** (design cycle)

ML Course structure
Where we go

Dip. Informatica
University of Pisa

Firsts learning alg on a simple hp space



1 INTRO

Function approximation framework
Data, Task, Model, Learn. Alg., Validation

Discrete H

2 ~~Concept learning~~

Continuous H

3 Linear models (LTU-LMS)

4 K-nn

Probabilistic

Ind. Bias

5 Neural Networks

10 SOM

11 RNN

6 Deep L.

12 Bayesian Networks

7 Validation & SLT

Bias/Variance 13

Theory

8 SVM

9 Applications/Project

14 Advanced topics

For infornation

Alessio Micheli

micheli@di.unipi.it

Dipartimento di Informatica
Università di Pisa - Italy

**Computational Intelligence &
Machine Learning Group**