ML-PREP
Training Session
PROGRAM

Mark Asch - ML-PREP

2025

Outline

Day	Topic	Links
1	Machine Learning	Lectures Examples
2	Geostatistics	Lectures Examples
3	Wave propagation	Lectures Examples
4	Research projects	Directory

1. PLEASE CLICK ON THE LINKS
2. The mornings will be dedicated to lectures.
3. The afternoons will concentrate on practical examples and exercises.

DAY 1:

MACHINE LEARNING

Advanced ML

- Pre-requisites
  - ⇒ Basics of Machine Learning: lecture notes and examples are here.
- Theory:
  - ⇒ how to choose a method?
  - ⇒ cross-validation and tuning
  - ⇒ evaluation and performance metrics
  - ⇒ causality and correlation
  - ⇒ features and model selection
  - ⇒ PINN
- Examples and Exercises:

DAY 2:

GEOSTATISTICS
and
MACHINE LEARNING

Geostatistics & ML

- Pre-requisites:
  - ⇒ Basic course lecture notes
- Theory:
  - ⇒ Geostatistics
    - → probability and stochastic processes
    - → variograms
    - → kriging
  - ⇒ Geospatial data analysis and machine learning
    - → model evaluation
    - → spatial cross-validation
- Examples and Exercises

DAY 3:

WAVE PROPAGATION

Wave Propagation

- Pre-requisites

- Theory:
  - $\Rightarrow$ basics of seismic wave propagation: harmonic waves, acoustic waves, seismic waves
  - $\Rightarrow$ finite difference method
  - $\Rightarrow$ finite element method
  - $\Rightarrow$ spectral element method

- Examples and Exercises

DAY 4:

RESEARCH PROJECTS

Propositions

1. Landslide inventory using ensemble, tree-based machine learning.
2. Extensive machine learning study of parameters in the Factor of Safety formula of Newmark.
3. Coupling seismic wave propagation with landslide triggering (Newmark equation).

TRAINING OVERVIEW

Where do we begin?

- Different starting points and prior experience:
  - $\Rightarrow$ mathematics, statistics, probability theory
  - $\Rightarrow$ data science and machine learning
  - $\Rightarrow$ GIS and geospatial data

Where do we arrive?

- Unique end point that ensures everyone is at the same level of knowledge of:

  $\Rightarrow$ machine learning,
  $\Rightarrow$ geospatial data analysis,
  $\Rightarrow$ basic seismic wave propagation.

How do we get there?

- Presentation of tools in a big toolbox.
- Understanding why, and not just how!
- Ethical, reproducible and responsible science...

ML for Science

- Motivation: see initial lecture.

- Objective: find the mapping (function, pattern) f that relates outcomes Y (observations, measurements) to explanatory variables (inputs, features, causes) X, such that
$$Y = f(X) + \epsilon,$$
where ϵ represents the intrinsic uncertainty (noise) of the underlying phenomenon.

- In practice, we will seek an approximation $\hat{f}$ to f such that the resulting
$$\hat{Y} = \hat{f}(X)$$
is as close as possible to Y.

- This is done by minimizing a suitable loss function
$$\mathcal{L} = \left\| Y - \hat{Y} \right\|.$$

The PTP method

- PTP = Please the Professor... (or the Boss)
- In Machine Learning we are strongly tempted to do our best by minimizing the model error:

  $\Rightarrow$ RMSE (root mean-squared error) for regression problems,

  $\Rightarrow$ CEE (cross-entropy error) for classification problems.

- Recall the XKCD cartoon: "stir the pile until the answer looks right"

  $\Rightarrow$ This is NOT reproducible, NOT responsible, NOT ethical.

A simple example that brings out many points

- **Basic fact**: most of ML is about pattern recognition—see C. M. Bishop, PRML, Springer, 2006.

  $\Rightarrow$ we have a bunch of data, $\mathbf{x}$ and corresponding observations, $\mathbf{y}$

  $\Rightarrow$ we seek the relation (recognize the pattern), f, that relates $\mathbf{x}$ to $\mathbf{y}$

  $$\mathbf{y} = f(\mathbf{x}),$$

  or,

  $$f : \mathbf{x} \longmapsto \mathbf{y}$$

  $\Rightarrow$ our goal: make predictions for new (unseen) data (inputs)

Polynomial Curve Fitting

- suppose that

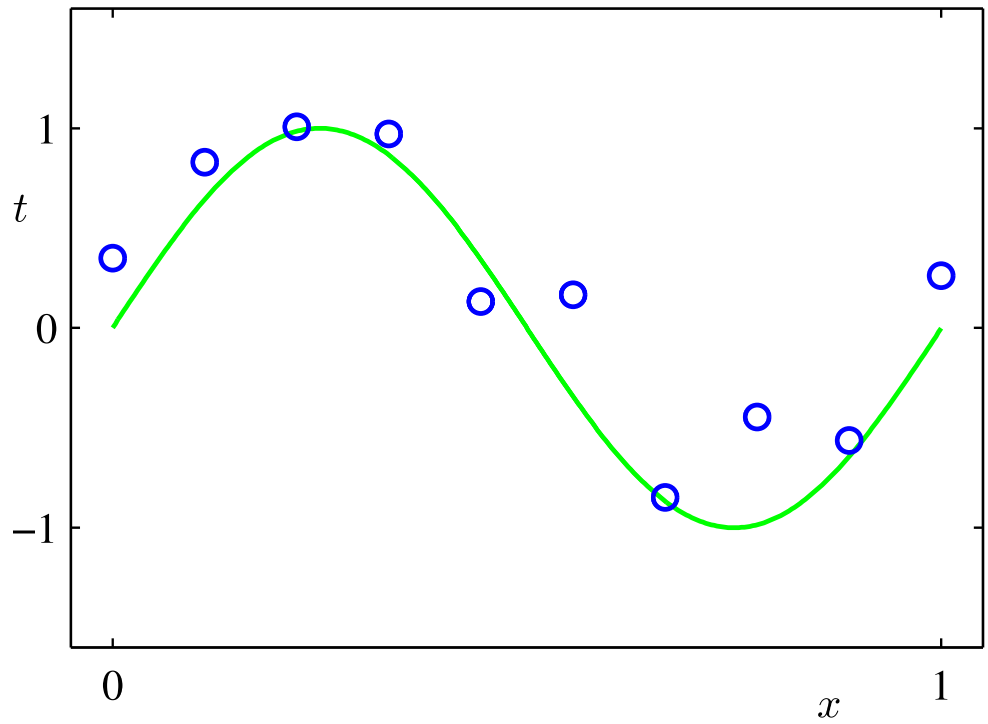
$$y = \sin(2\pi x) + \text{noise}$$

- given: a training set of N observations of x,

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}$$

and the corresponding (noisy) observations,

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$



- data generation from

$$y_i = \sin(2\pi x_i) + \epsilon,$$

where the noise term,

$$\epsilon \sim \mathcal{N}(0, \sigma)$$

is due to intrinsic/natural variability and other uncertainties from the data collection methods.

- **Objective:** from the training set, learn/discover an approximation $\hat{f}$ of the relation f that will enable us to make predictions based on:

  ⇒ probability theory,

  ⇒ (polynomial) curve fitting.

- PCF: suppose that the data can be represented by the polynomial or "linear model"

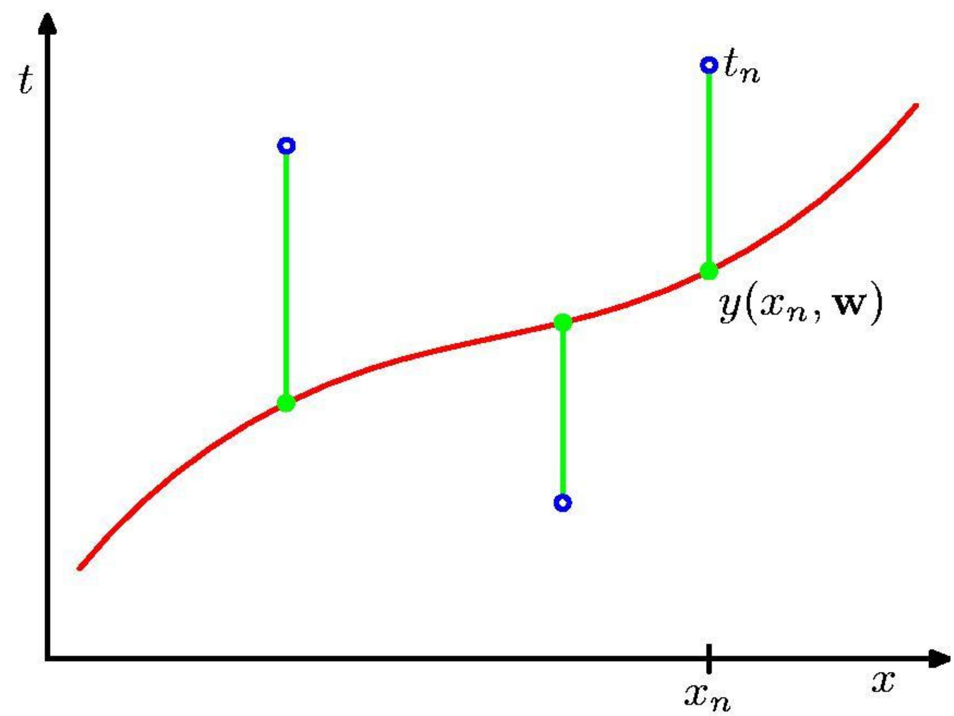
$$y(x, \mathbf{w}) = w_0 + w_1 x + w_2 x^2 + \cdots + w_M x^M = \sum_{k=0}^{M} w_k x^k$$

with weight vector (coefficients) $\mathbf{w}$.

  ⇒ compute the coefficients by minimizing an error/loss function—this is (often) a least-squares approach, where the loss function is defined as

$$\mathcal{L}(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^{N} \left[\hat{y}(x_i, \mathbf{w}) - y_i \right],$$

  where $\hat{y}_i$ is the approximation obtained from our polynomial model.



- in the case of a quadratic loss function, the gradient is linear and the solution to the optimization problem exists and can be found in closed form using the normal form, or psuedo-inverse,

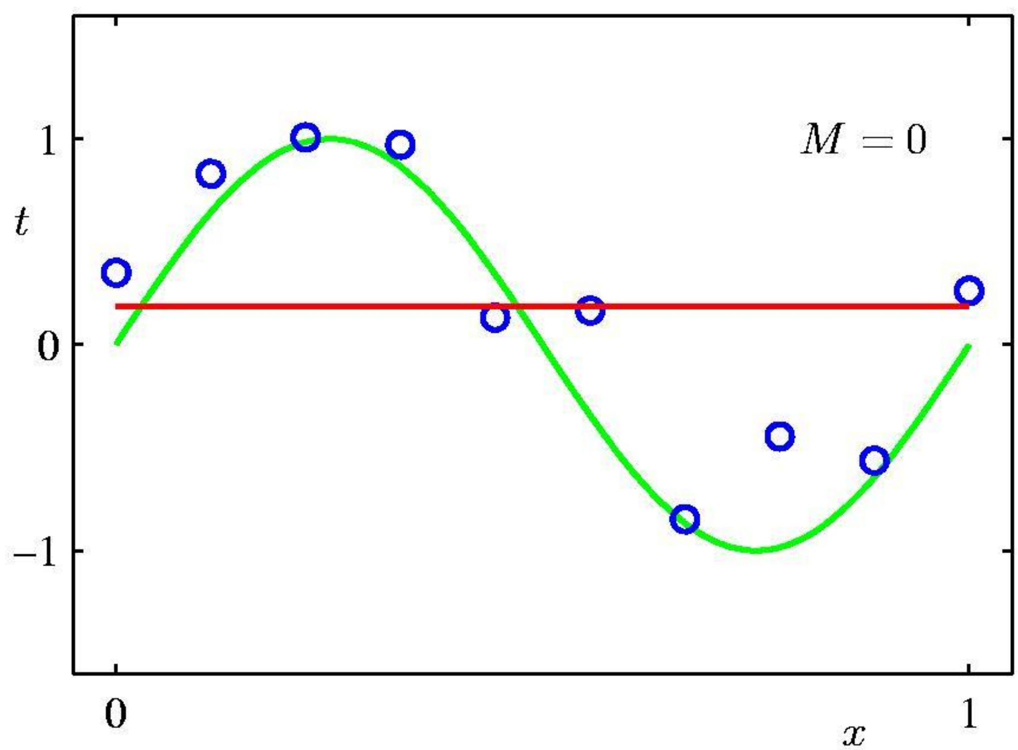
$$\mathbf{w}_* = \left(X^{\mathrm{T}} X\right)^{-1} X^{\mathrm{T}} \mathbf{y},$$

where X is the (rectangular) coefficient matrix of dimension $(N \times M)$ with $N > M$ and $\mathbf{y}$ is the data/observation vector.

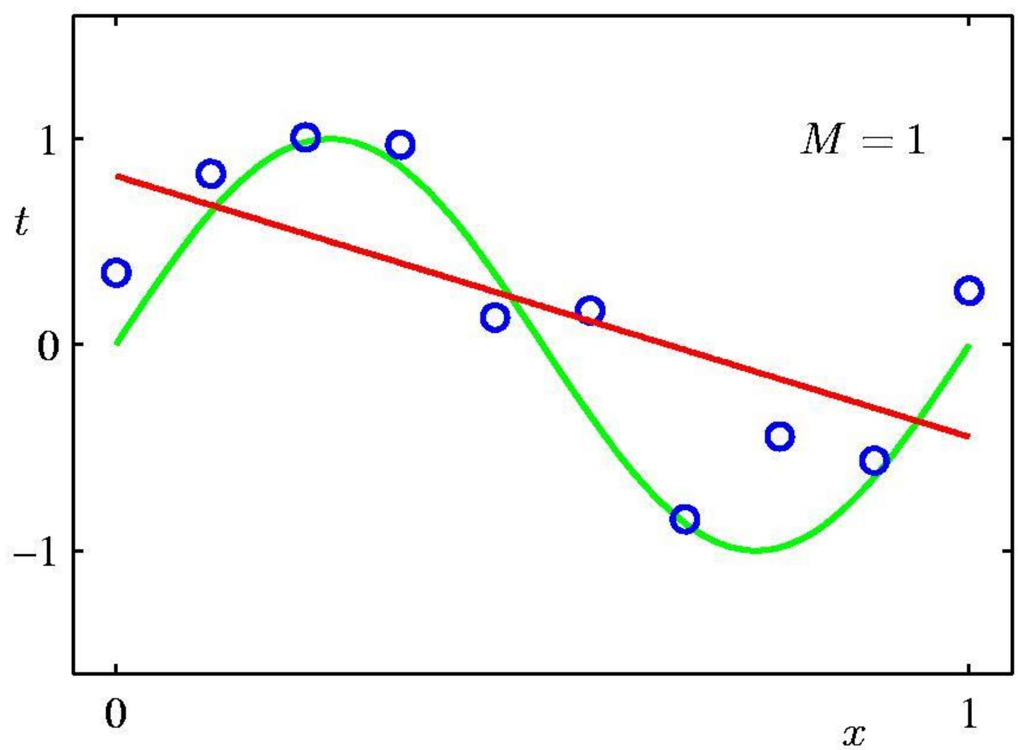
- the choice of M is a model selection problem
  - ⇒ we compute the solution for 4 values: $M = 0, 1, 3, 9$—see below
  - ⇒ Model $M = 0$ produces underfitting—it is too simple and does not find the trend
  - ⇒ Model $M = P$ produces overfitting—it fits the data points perfectly, but is "brittle" and does not generalize

- **Conclusion:** we must seek a compromise between the two.

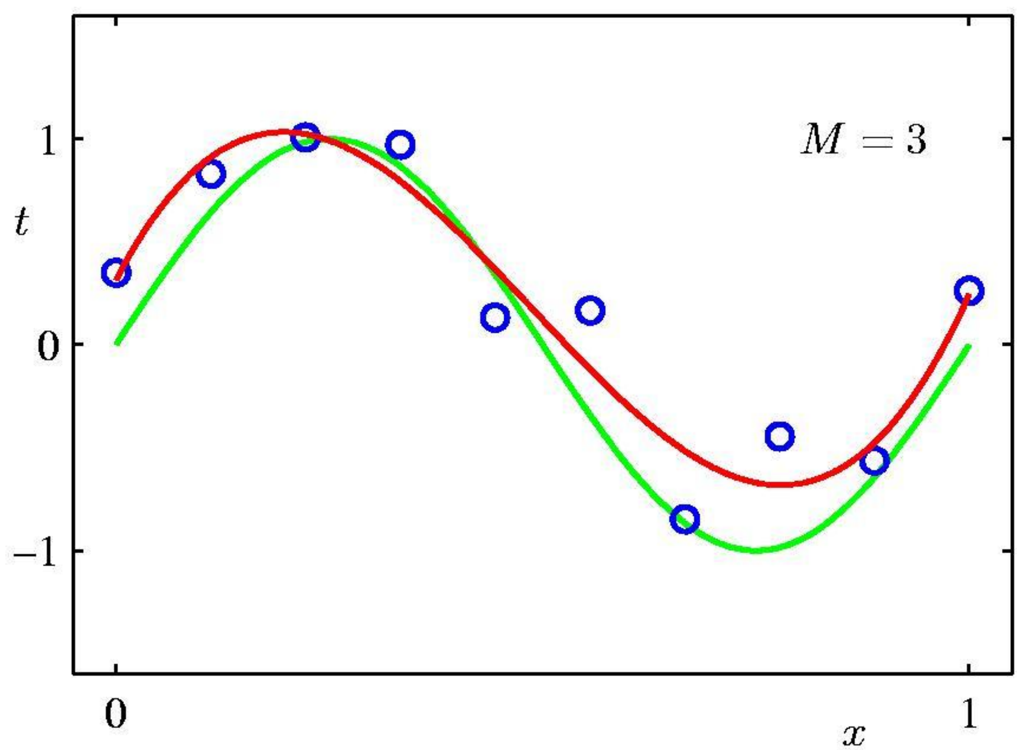
0^{th} Order Polynomial



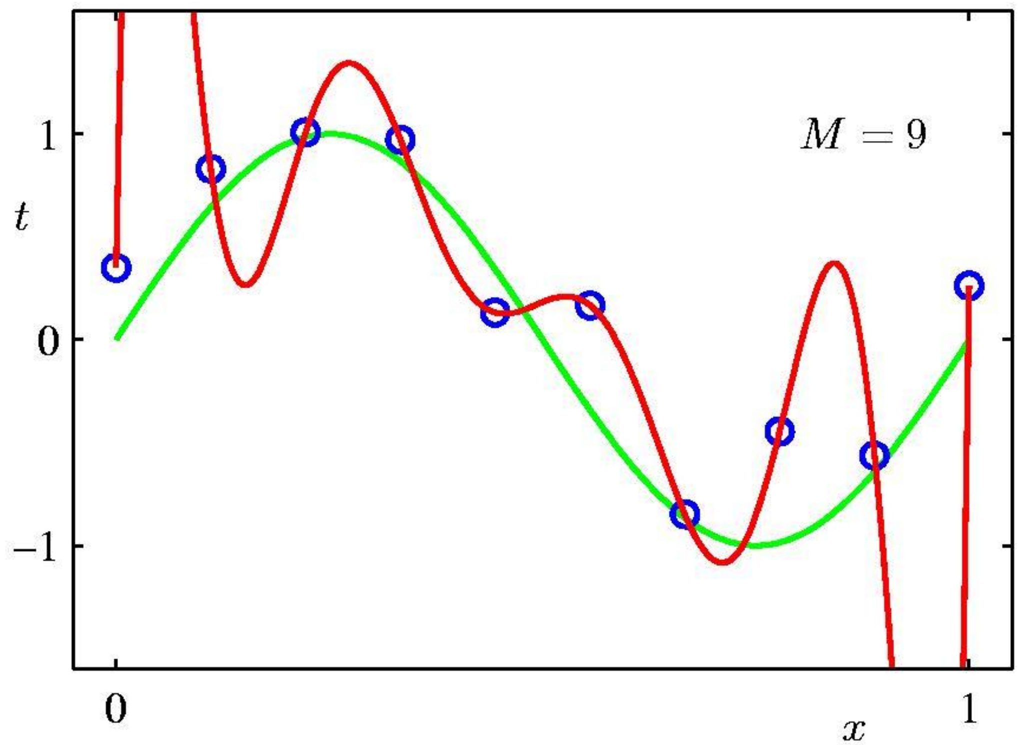
1st Order Polynomial



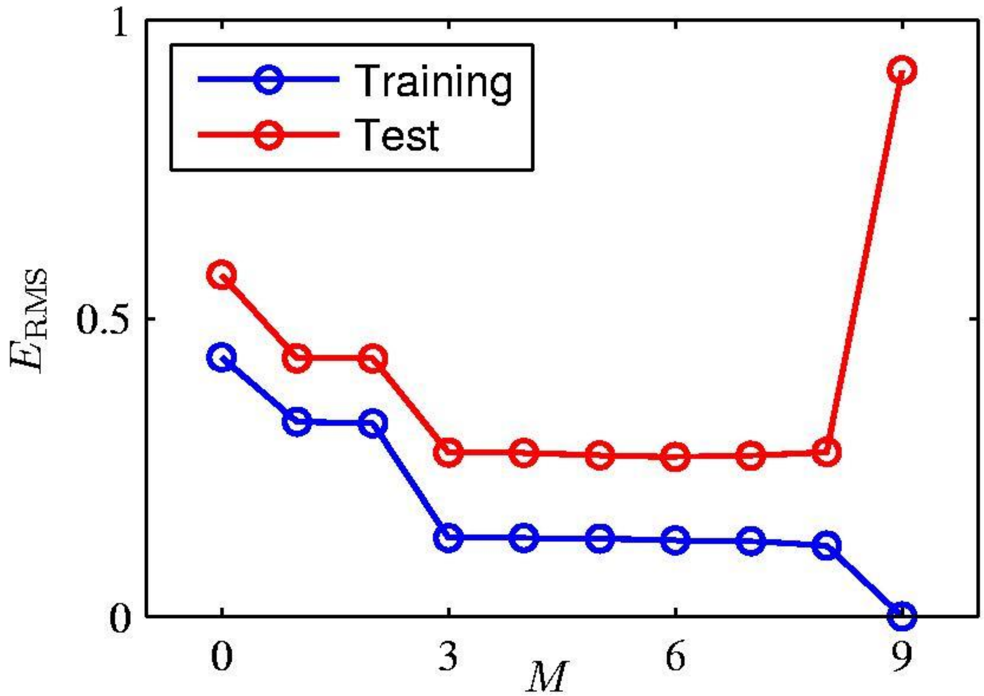
3rd Order Polynomial



9th Order Polynomial



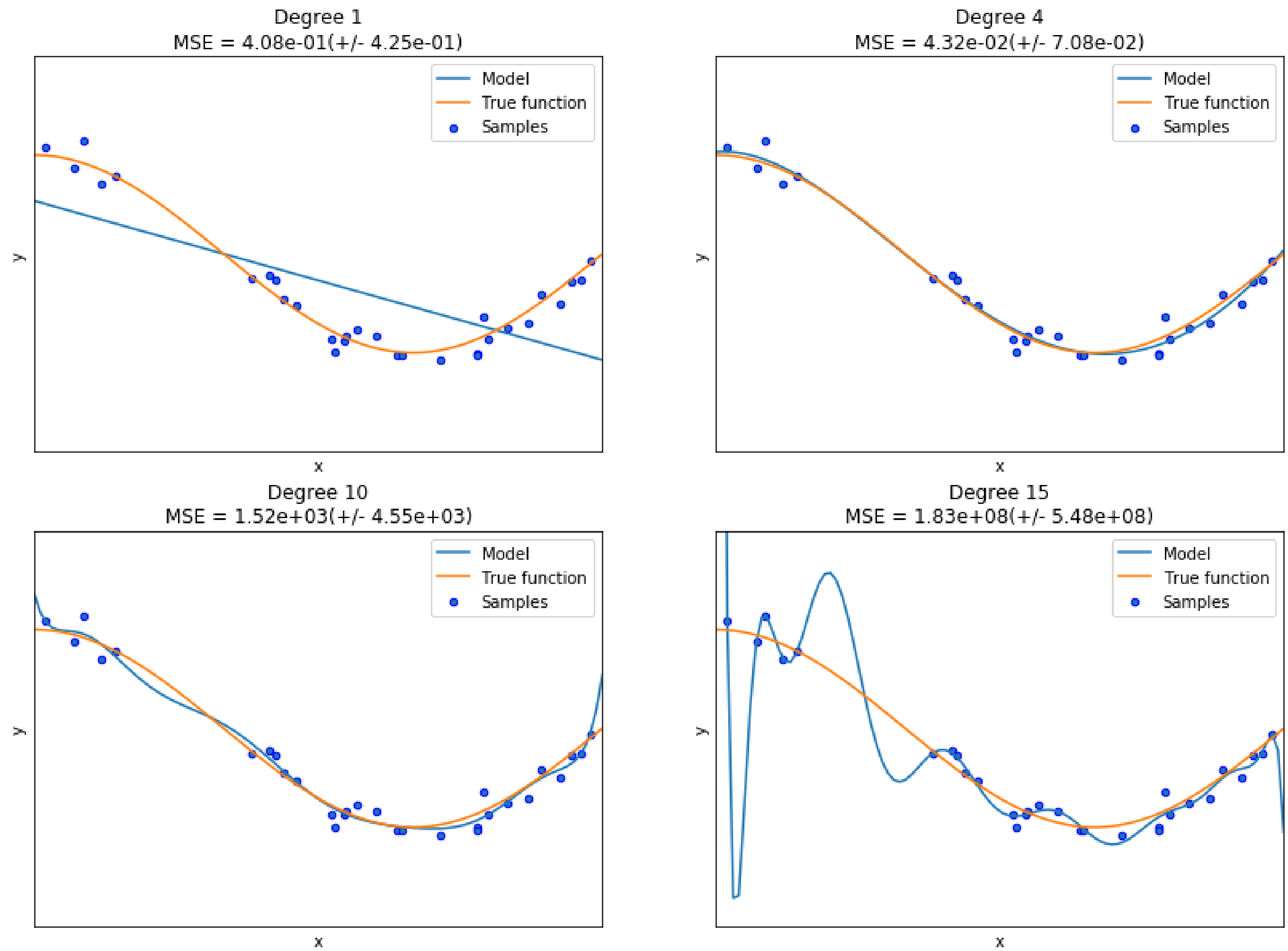
Over-fitting



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

Polynomial Coefficients

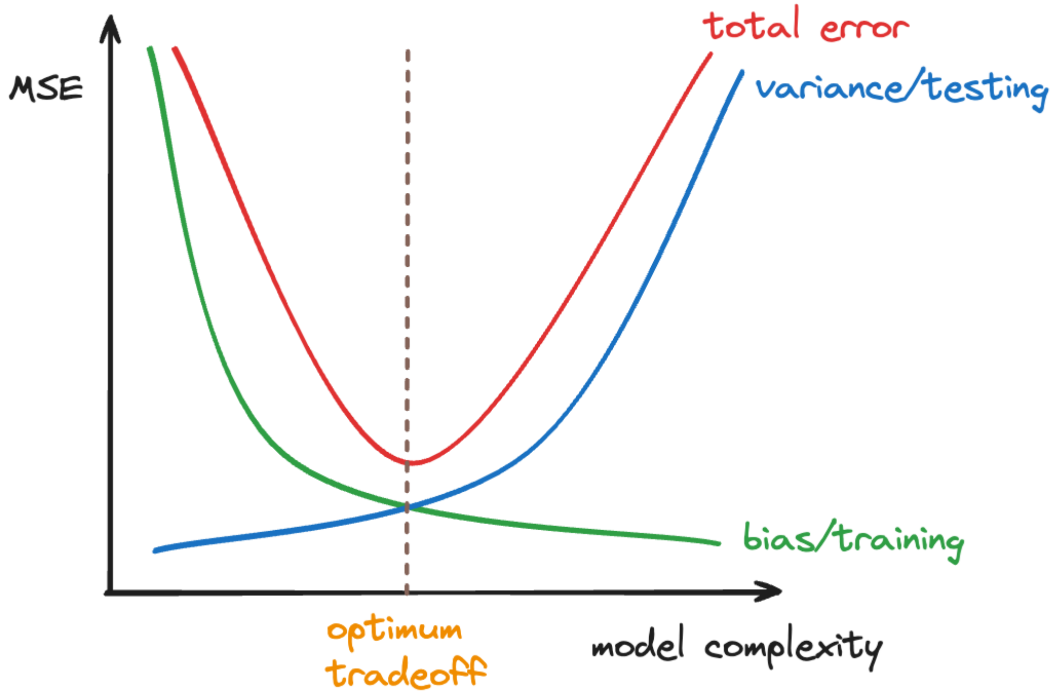
	$M = 0$	$M = 1$	$M = 3$	$M = 9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43



M	MSE
1	0.04
4	0.004
10	1520
15	1.8e8

What goes wrong here?

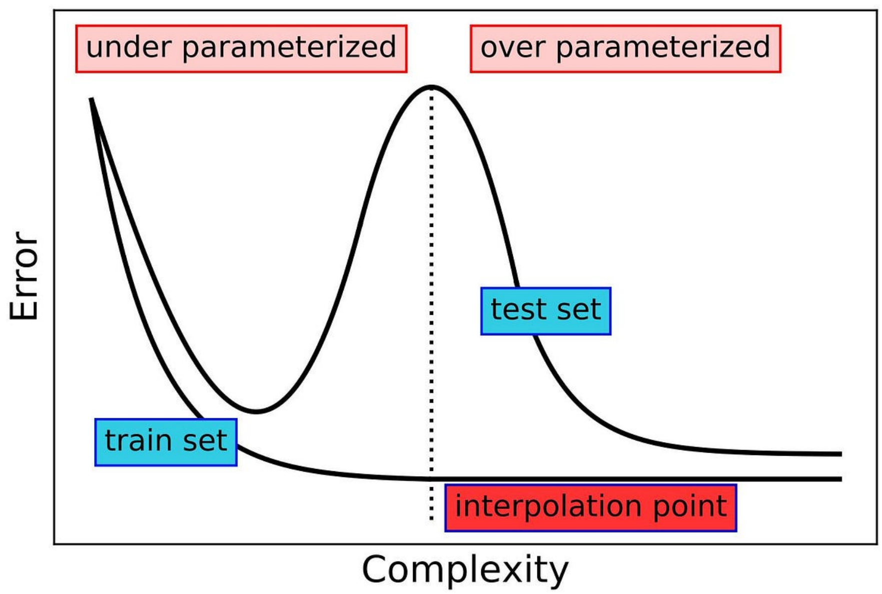
- The bias-variance tradeoff:



- Option 1: fit the noise $\Rightarrow$ high accuracy (low bias), but large uncertainty (high variance)—overfitting.

- Option 2: compromise $\Rightarrow$ lower accuracy (higher bias), but lower uncertainty (lower variance)

- Double-descent phenomenon (for DNNs):



What is the objective?

- The principal objective is good predictive performance, NOT good training accuracy.

- Mathematically, recall the trained ML model for $Y = f(X)$, f unknown,

$$\hat{f}: X \to \hat{Y}$$

- Prediction: for $(X^*, Y^*) \notin (X, Y)$, how accurate is $\hat{f}(X^*)$?

- In other words, if

$$\hat{f}: X^* \to \hat{\hat{Y}},$$

then how big is the error

$$\left\| \hat{\hat{Y}} - Y^* \right\| ?$$

How do we avoid the trap?

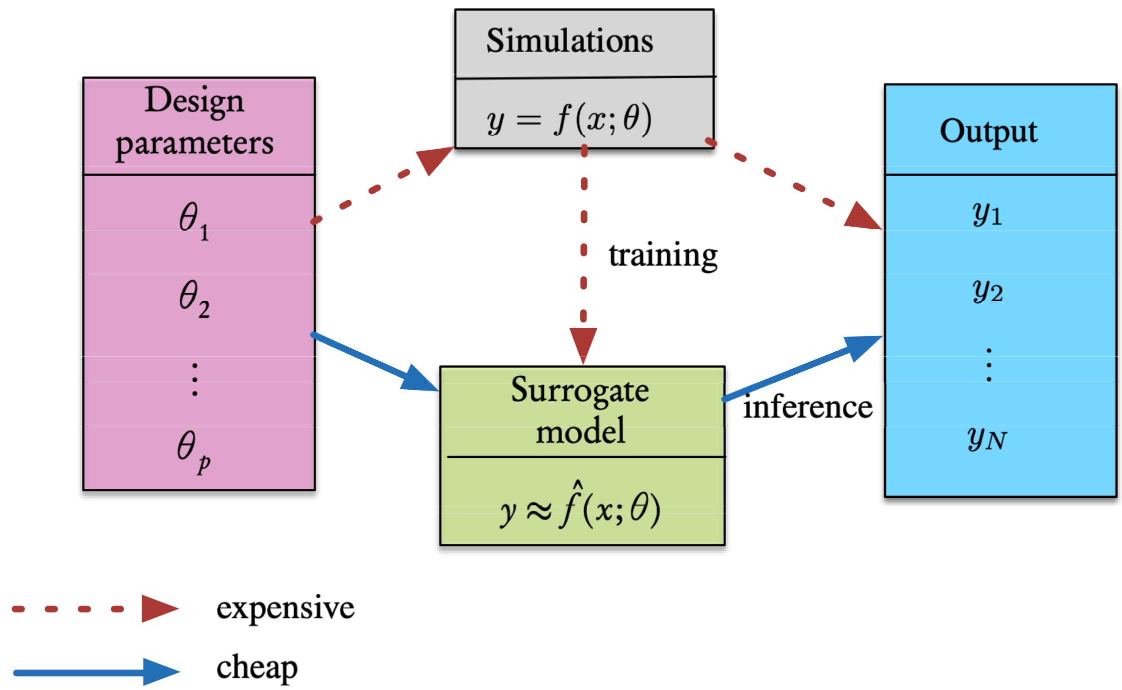
- By definition, unseen data has not been seen in the training...

- Best effort in this case is to use

  ⇒ nested cross-validation for tuning and testing
  ⇒ repeated cross-validation for training and testing

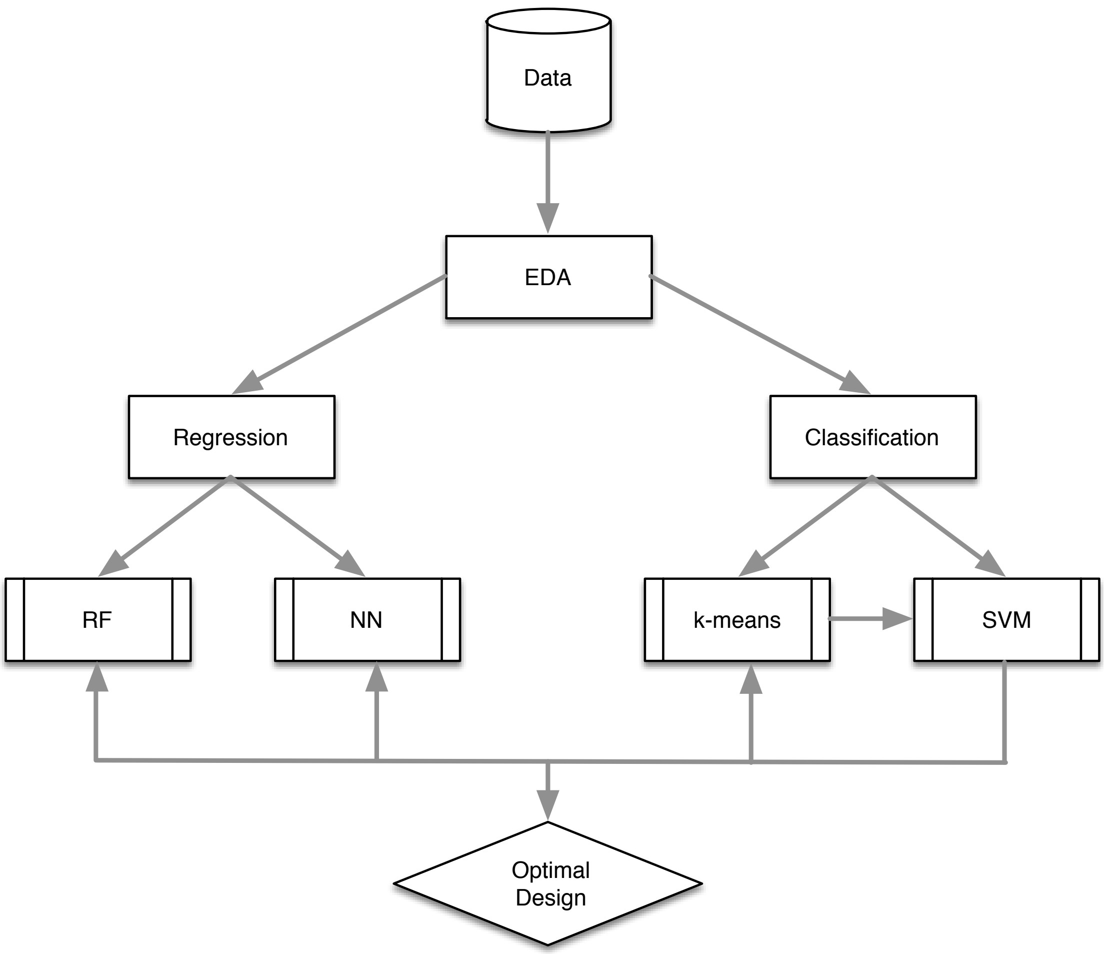
- Then to report confidence intervals, taking uncertainty into account—not just the "best" result (see PTP metod above).

SUMO

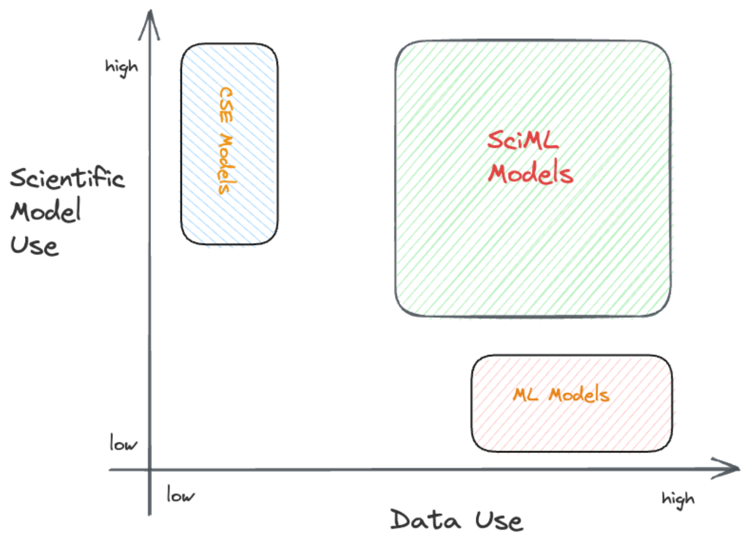


Simulations
$y = f(x;\theta)$
Design parameters
θ_1
θ_2
$\vdots$
θ_p
training
Surrogate model
$y \approx \hat{f}(x;\theta)$
inference
Output
y_1
y_2
$\vdots$
y_N
expensive
cheap

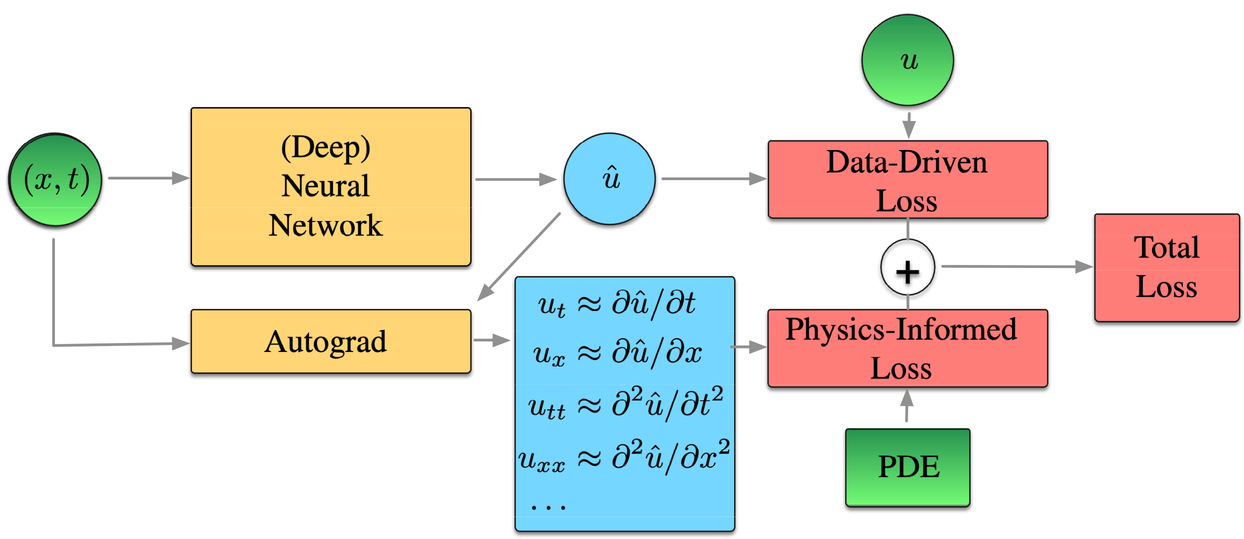
EDA


Data
EDA
Regression
Classification
RF
NN
k-means
SVM
Optimal
Design

SciML & PINN



high
Scientific
Model
Use
CSE Models
SciML
Models
ML Models
low
low
Data Use
high



u
(x, t)
(Deep)
Neural
Network
$\hat{u}$
Data-Driven
Loss
Total
Loss
+
Autograd
$u_t \approx \partial\hat{u}/\partial t$
$u_x \approx \partial\hat{u}/\partial x$
$u_{tt} \approx \partial^2\hat{u}/\partial t^2$
$u_{xx} \approx \partial^2\hat{u}/\partial x^2$
$\dots$
Physics-Informed
Loss
PDE

WARNING: ML is not everything!

- ML depends on data availability and data quality—without these we cannot obtain good models and reliable predictions.

- ML should be coupled with classical modeling approaches
  - ⇒ statistics
  - ⇒ differential equations
  - ⇒ empirical knowledge.

- ML (IMHO) will never replace human researchers, and we should not be fooled by apparent fluency as exhibited by LLMs, where we forgo explainability, transparency and reproducibility.

References

1. G. James, D. Witten, T. Hastie, R. Tibshirani, J. Taylor. *An Introduction to Statistical Learning*. Springer. 2023. ISL Site

2. C. Wikle, A. Zammit-Mangion, N. Cressie. *Spatio-Temporal Statistics with R*. CRC Press. 2019. https://spacetimewithr.org/

3. H.P. Langtangen. *Finite Difference Computing with PDEs: A Modern Software Approach.* Springer Cham. 2017. Download pdf

4. H. Igel. *Computational Seismology: A Practical Introduction.* Oxford University Press, 2017. website