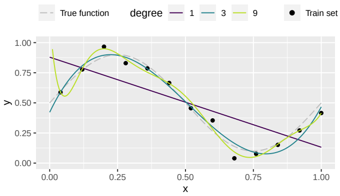
Introduction to Machine Learning

Evaluation: Training Error

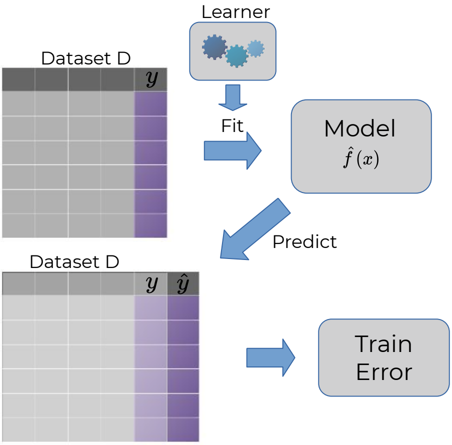


Learning goals

- Understand the definition of training error
- Understand why training error is no reliable estimator of future performance

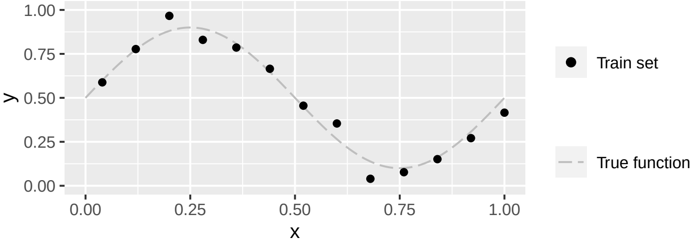
TRAINING ERROR

(also: apparent error / resubstitution error)



EXAMPLE: POLYNOMIAL REGRESSION

Sample data from sinusoidal function $0.5 + 0.4 \cdot \sin(2\pi x) + \epsilon$
with measurement error ϵ.

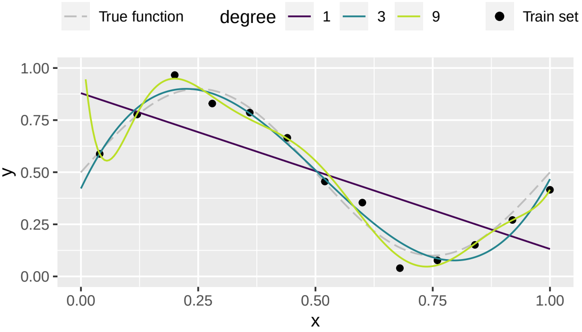


Assume data-generating process unknown.
Try to approximate with a d^{th}-degree polynomial:

$$f(\mathbf{x} \mid \boldsymbol{\theta}) = \theta_0 + \theta_1 x + \cdots + \theta_d x^d = \sum_{j=0}^{d} \theta_j x^j.$$

EXAMPLE: POLYNOMIAL REGRESSION

Models of different *complexity*, i.e., of different orders of the polynomial are fitted. How should we choose *d*?



- d=1: MSE = 0.036: Clear underfitting
- d=3: MSE = 0.003: Pretty OK?
- d=9: MSE = 0.001: Clear overfitting

Simply using the training error seems to be a bad idea.

TRAINING ERROR PROBLEMS

- Unreliable and overly optimistic estimator of future performance. E.g., training error of 1-NN is always zero as each observation is its own NN during test time.
- Goodness-of-fit measures like (classic) R^2, likelihood, AIC, BIC, deviance are all based on the training error.
- For models of restricted capacity, and given enough data, the training error may provide reliable information.
  E.g., LM with $p = 5$ features, 10^6 training points.
  But: impossible to determine when training error becomes unreliable.