Chapter 1
Introduction

supplementary slides to

Machine Learning Fundamentals

published by Cambridge University Press

August 2020

Outline

Machine Learning

- **artificial intelligence** (AI):
  - history of AI
  - broad definition of AI: mimic human intelligence
  - narrow definition of AI: rule-based symbolic approaches
- **machine learning** (ML): data-driven statistical methods
- AI vs. ML
  - AI: manual construction of knowledge bases
  - ML: automatic learning from training data
- paradigm shift: knowledge-based $\rightarrow$ data-driven
- machine learning pipeline:

Basic Concepts in Machine Learning

- classification vs. regression
- supervised vs. unsupervised learning
- simple vs. complex models
- parametric vs. non-parametric models
- over-fitting vs. under-fitting
  - bias-variance tradeoff

Machine Learning: classification vs. regression

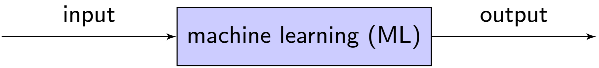


Figure: A system view of any machine learning problem

- **classification** problems: outputs are discrete and finite
- **regression** problems: outputs are continuous
- **structured prediction**: outputs are structured objects

Machine Learning: supervised vs. unsupervised learning

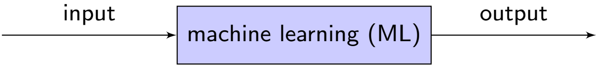


Figure: A system view of any machine learning problem

- supervised learning
- unsupervised learning
- semi-supervised learning
- weakly-supervised learning
  - self-supervised learning

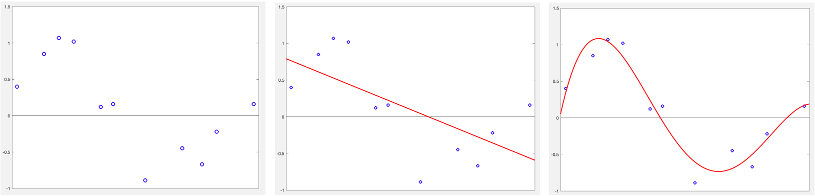
Simple vs. Complex Models

- crucial to choose a **right** model in machine learning
- simple vs. complex models
- model complexity depends on the function form and the number of free parameters.
- simple models: linear models
  - less training data; less computing resources
  - mediocre performance in practice
- complex models: nonlinear models (e.g. *neural networks*, *decision trees*)
  - superior performance when sufficient training data are available
  - more training data require more computing resources
  - difficult to analyze and interpret

Simple vs. Complex Models

Example: curve fitting

- a regression problem: $x \mapsto y$
- a simple model: a linear model $y = a_0 + a_1\,x$
- a complex model: a 4th-order polynomial
  $y = a_0 + a_1\,x + a_2\,x^2 + a_3\,x^3 + a_4\,x^4$



(a) training data (b) simple model (c) complex model

Parametric vs. Non-parametric Models

- parametric models: *a.k.a.* finite-dimensional models
  - the function form is given
  - the model is fully determined by a *fixed number of parameters*
- non-parametric models: *a.k.a.* distribution-free models
  - the function form is not specified
  - the model complexity depends on the available data

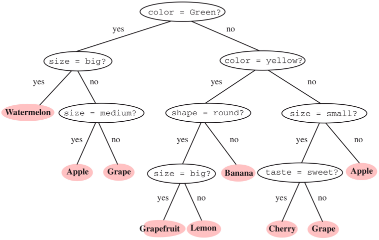
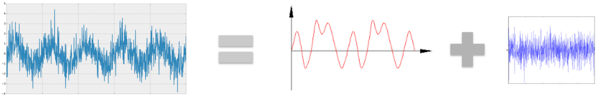


Figure: Decision trees: a non-parametric model

Over-fitting vs. Under-fitting

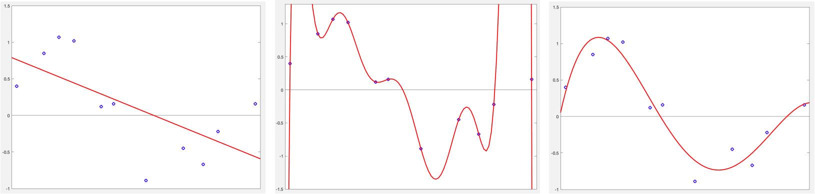


data signal noise

$$\text{data} = \text{signal} + \text{noise}$$

- simple models $\Longrightarrow$ under-fitting
  - too weak to capture the regularities in data
  - increase model complexity
- complex models $\Longrightarrow$ over-fitting
  - perfectly fit random noises
  - totally useless to fit noises as they vastly change each time
  - decrease model complexity; add more data; *regularization*

Over-fitting vs. Under-fitting



(a) under-fitting (b) over-fitting (c) good-fitting

Figure: under-fitting vs. over-fitting in regression

Over-fitting vs. Under-fitting

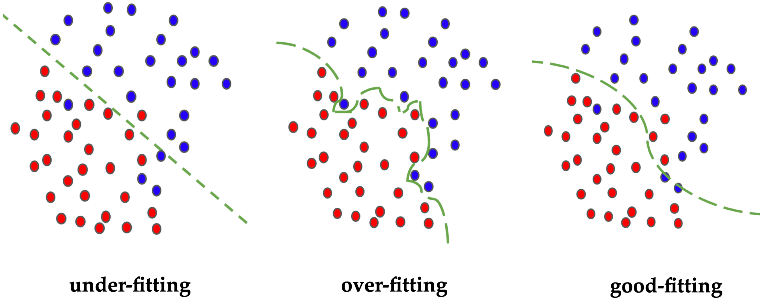
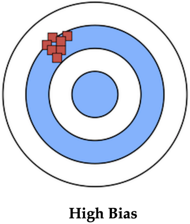


Figure: under-fitting vs. over-fitting in classification

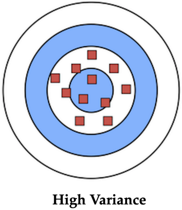
Bias-Variance Tradeoff

- simple models $\Longrightarrow$ under-fitting $\Longrightarrow$ high biases
- complex models $\Longrightarrow$ over-fitting $\Longrightarrow$ high variances
- bias and variance decomposition:

$$\text{average learning error} = \text{bias}^2 + \text{variance}$$



(a) high learning bias



(b) high learning variance

Bias-Variance Tradeoff

- cannot simultaneously reduce both bias and variance when learning from a fixed amount of data
- tradeoff between bias and variance for the lowest total error

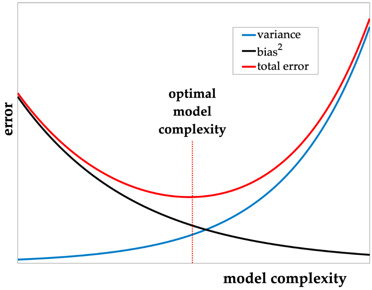


Figure: bias-variance tradeoff as a function of model complexity

General Principles in Machine Learning

- Occam's Razor
- No Free Lunch Theorem
- Law of the Smooth World
- Curse of Dimensionality
- Blessing of Non-uniformity

Occam's Razor

- a general principle in philosophy
  - *the simplest solution is most likely the right one*
- a preference for simplicity in model selection
- it suggests the **minimum description length** (MDL) principle
  - an important learning criterion in machine learning
  - the best model to describe the regularities in data is the one that can compress the data most.

No Free Lunch Theorem

- no learning method is universally superior to other methods for all possible learning problems
- no machine learning algorithm can learn anything useful **merely** from the training data
- a successful machine learning algorithm must have explicitly or implicitly used some knowledge beyond the training data

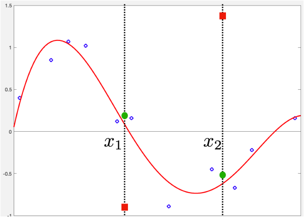


Figure: An illustration of No Free Lunch Theorem

Law of the Smooth World

- physical processes are smooth due to energy/power constraints
- real-world data are smooth, e.g. audio/speech/images/video
- the smoothness of the ground-truth is mathematically quantified by *Lipschitz continuity* or *bandlimitedness*

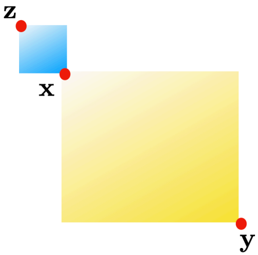
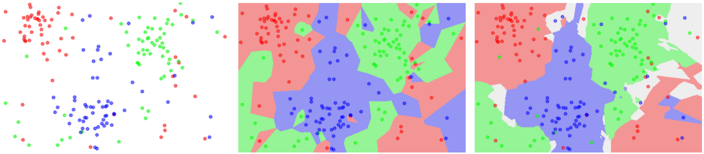


Figure: How the law of the smooth world helps in machine learning

k-nearest neighbors (k-NN)

- the law of the smooth world suggests the k-nearest neighbors (k-NN) method:
    - an unknown object is classified based on its k nearest neighbors in the training set
- k-NN is simple and intuitive
- how to measure distance? e.g. *metric learning*
- whether training data are enough to cover the whole space?



(a) training data (b) k-NN ($k = 1$) (c) k-NN ($k = 5$)

Curse of Dimensionality

- **curse of dimensionality**: the dilemma of learning in high-dimensional spaces
  - as the dimensionality grows, it requires the exponentially increasing amount of training data and computing resources to ensure the effectiveness of learning
- e.g. the k-NN method requires N training samples to ensure classification error ϵ $(0 < \epsilon < 1)$ in a d-dimensional space:

$$N \propto \left(\frac{\sqrt{d}}{\epsilon}\right)^{d+1}$$

Assume $\epsilon = 0.01$, if it requires $N = 100$ when $d = 3$. When $d = 10$, it needs $N = 2 \times 10^8$, and it requires $N = 7 \times 10^{123}$ when $d = 100$.

Blessing of Non-uniformity

- the worst-case scenarios predicted by the curse of dimensionality normally occur when the data are uniformly distributed in high-dimensional spaces
- **blessing of non-uniformility**: real-world data never spreads evenly throughout the high-dimensional spaces but rather congregates on
  - linear subspaces
  - lower-dimensional nonlinear subspaces, called *manifolds*.
- it makes machine learning in high-dimensional spaces feasible
- it suggests **dimensionality reduction**:
  - linear dimensionality reduction
  - manifold learning

Advanced Topics in Machine Learning

- reinforcement learning
- meta-learning (*a.k.a.* learning to learn)
- causal inference
- transfer learning (*a.k.a.* domain adaptation)
- online learning
- active learning
- imitation learning