Introduction to Machine Learning

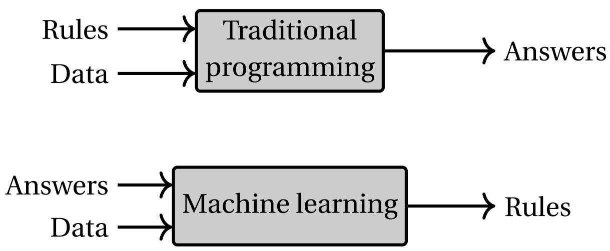
Session 1

Intro

Maxime Ossonce

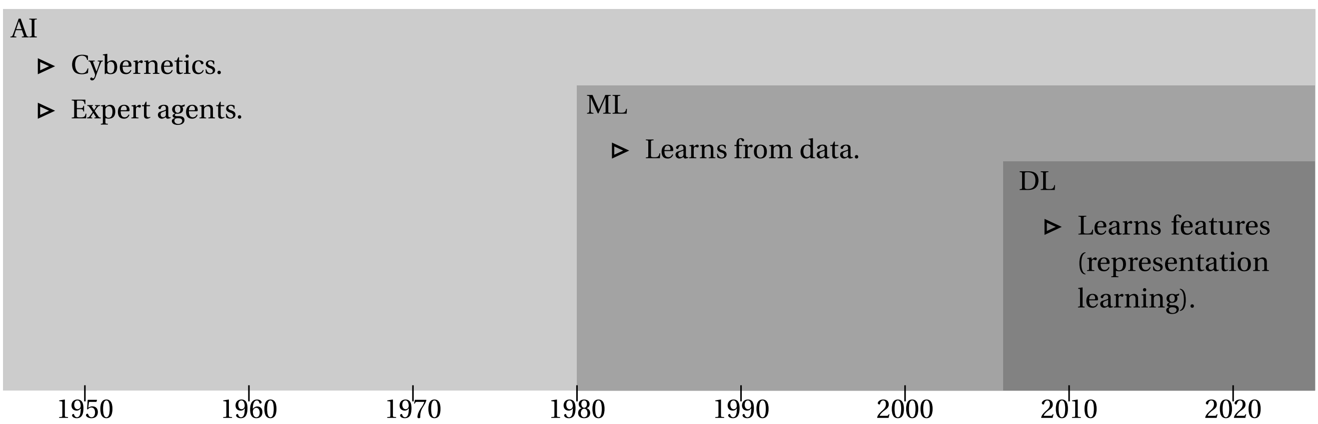
What is machine learning (ML)?

- Data based programming: improve **performance** at some **task** with **experience**



- Model inference.

ML and artificial intelligence (AI)

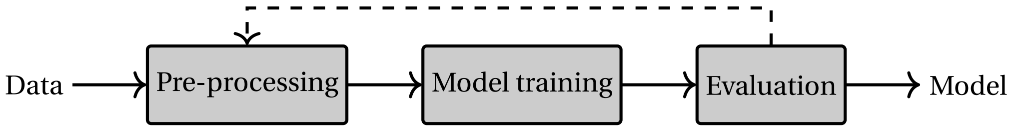


Applications

- Image classification, object detection.
- Medical diagnosis.
- Audio captioning.
- Chatbot.
- Product recommendation.

ML project

Data engineering process



1. Visualize, pre-processing of the data.
2. Identify the approach.
3. Design a model.
4. Evaluate the model.
5. Go back to **??** if needed.

Approaches

- Supervised learning:
  - classification,
  - regression.
- Unsupervised learning:
  - clustering,
  - dimension reduction.

Supervised learning

- Given a **training set** of N samples $\left\{(x^i, y^i) : i \in 1, \cdots, N\right\}$ we want to **estimate** the function $y = f(x)$.
- Examples: image classification, object detection....

Unsupervised learning

- Given a **training set** of N samples $\left\{x^i : i \in 1, \cdots, N\right\}$ we want to **describe** how the data is organized.
- Examples: image segmentation, customers clustering....

Supervised learning principle

- $\mathcal{X}, \mathcal{Y}$ two sets with an unknown **distribution** $p(x, y)$.
- Find a function $f : \mathcal{X} \mapsto \mathcal{Y}$ which estimates y associated to x.
- f belongs to $\mathcal{H}$ the **hypothesis class**.
- *E.g.* linear regression:

$$\mathcal{H} = \left\{ f : x \mapsto y = ax + b : (a, b) \in \mathbb{R}^2 \right\}.$$

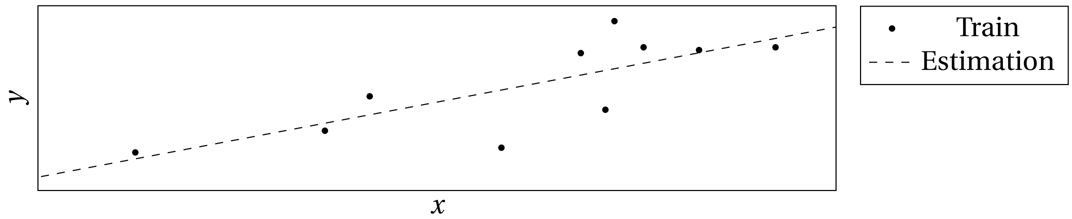


Figure: Simple linear regression.

Loss function

- The loss function L evaluates the relevance of $f(x)$:
  - $L(y, f(x)) = 0$ if $y = f(x)$.
  - $L(y, f(x)) > 0$ if $y \neq f(x)$.
- For **regression** ($\mathcal{Y} = \mathbb{R}$):
  - absolute deviation $|y - f(x)|$,
  - quadratic loss $(y - f(x))^2$.
- For **classification** ($\mathcal{Y}$ is a finite set):
  - $0 - 1$ loss,
  - hinge loss,
  - cross-entropy (CE).

Risk

- **True risk**
$$R(f) = \mathbb{E}_{(\mathsf{X},\mathsf{Y})\sim p(x,y)}\left[L(\mathsf{Y}, f(\mathsf{X}))\right].$$
- Objective find f^*:
$$f^* = \arg\min_{f\in\mathcal{H}} R(f).$$
- Be R^* the minimal risk attained over all measurable functions. The **approximation** error is
$$R(f^*) - R^*.$$

Empirical risk

- The distribution $p(x,y)$ is **unknown**.
- Only a finite **training set** is available $\{(x^i, y^i)\}_{i=1}^N$.
- **Empirical risk**
$$\hat{R}(f) = \frac{1}{N}\sum_{i=1}^{N} L(y^i, f(x^i)).$$
- Training: minimization of the **empirical risk**:
$$\hat{f} = \arg\min_{f\in\mathcal{H}} \hat{R}(f).$$
- The **estimation error** is
$$R(\hat{f}) - R(f^*).$$

Overfitting

$$R(\hat{f}) - R^* = \underbrace{R(\hat{f}) - R(f^*)}_{\text{estim. error}} + \underbrace{R(f^*) - R^*}_{\text{approx. error}}.$$

- Low complexity: approximation error (**bias**) high.
- High complexity: estimation error (**variance**) high.

Overfitting: example

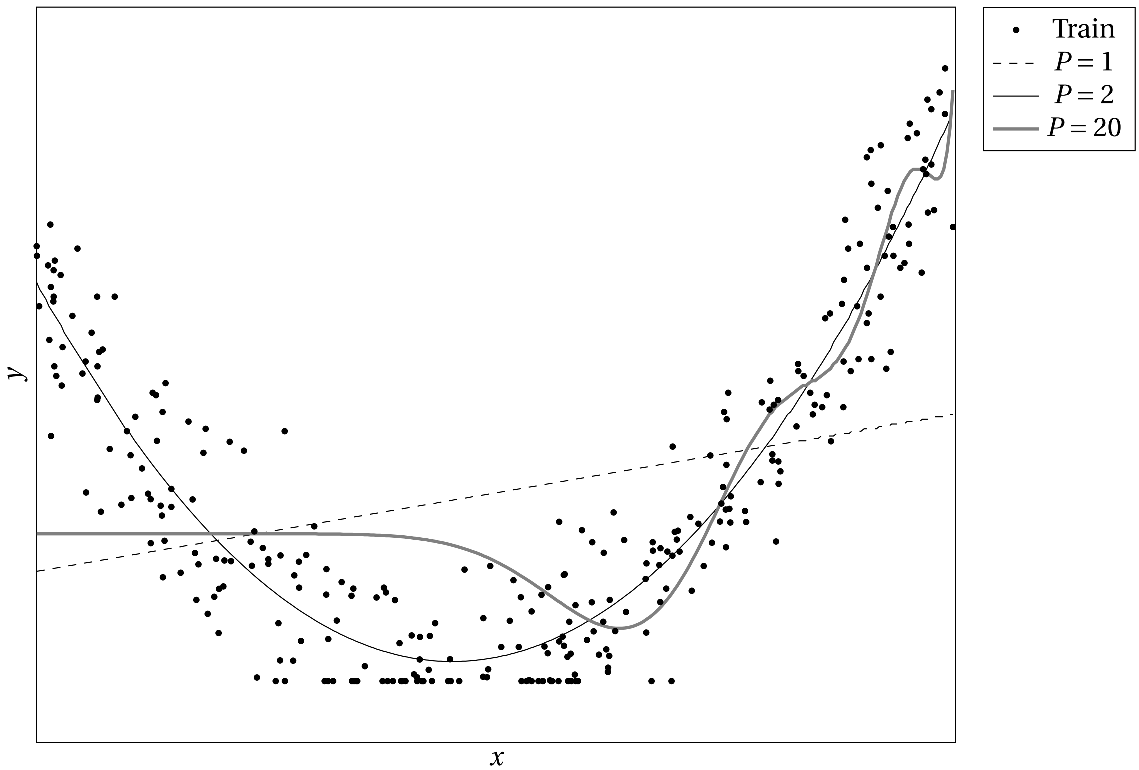
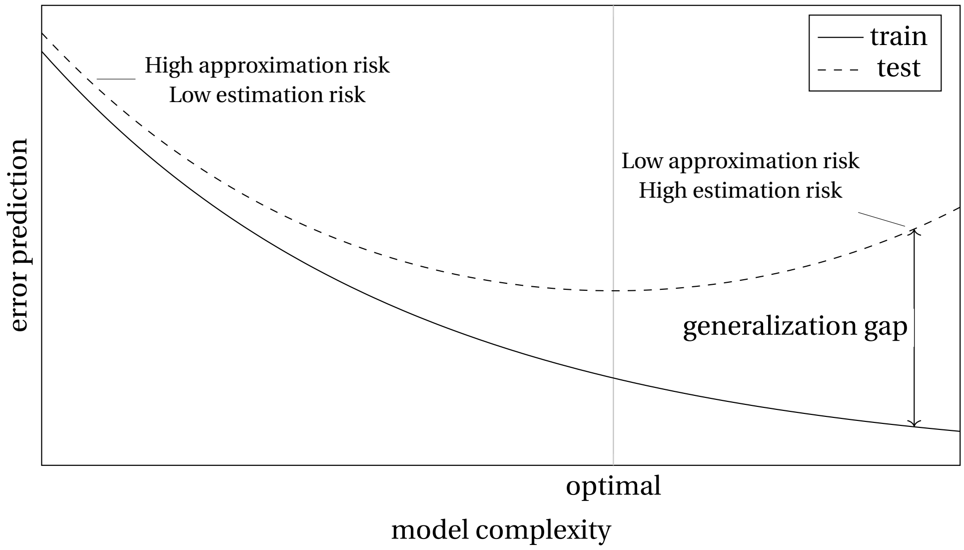


Figure: Polynomial regression.

Generalization gap



- Generalization gap: $R(f) - \hat{R}(f)$.
- $\hat{R}(f) \to 0$ when model complexity $\to \infty$.

Model selection

- Split dataset $\mathcal{D}$ in three sets $\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{valid}}, \mathcal{D}_{\text{test}}$
- For each **hyperparamter** α (*e.g.* polynom order), train the model and estimate f_α.
- Evaluate f_α on $\mathcal{D}_{\text{valid}}$.
- **Select** α with best **validation** performance.
- **Test** selected model on $\mathcal{D}_{\text{test}}$.

$\boxed{\mathcal{D}_{\text{test}} \text{ is only used once.}}$