* * * Artificial Intelligence and Machine Learning * * *

Machine Learning Introduction

Daniel Wehner, M.Sc.

SAP SE / DHBW Mannheim

Summer term 2025

Find all slides on GitHub (DaWe1992/Applied_ML_Fundamentals)

Lecture Overview

- I Machine Learning Introduction
- II Optimization Techniques
- III Bayesian Decision Theory
- IV Non-parametric Density Estimation
- V Probabilistic Graphical Models
- VI Linear Regression
- VII Logistic Regression
- VIII Deep Learning
- IX Evaluation
- X Decision Trees
- XI Support Vector Machines
- XII Clustering
- XIII Principal Component Analysis
- XIV Reinforcement Learning
- XV Advanced Regression

Agenda for this Unit

Section:

Introduction

Notation and Symbols
Motivation
Definition of Machine Learning

Notation

- **Scalars:**
  - Lower case letters, e. g. a, b, c, α, β, γ
  - Indices in subscript, e. g. x_1, x_2
- **Vectors:**
  - Lower case, bold face letters, e. g. $\boldsymbol{a}$, $\boldsymbol{b}$, $\boldsymbol{\theta}$
  - Indices in superscript, e. g. $\boldsymbol{\theta}^1$, $\boldsymbol{x}^1$
  - Components of vectors: θ_m references the m-th component of $\boldsymbol{\theta}$; $x_m^{(n)}$ is the m-th component of the vector $\boldsymbol{x}^n$ (braces avoid confusion with powers)
- **Matrices:**
  - Upper case, bold face letters, e. g. $\boldsymbol{X}$, $\boldsymbol{W}$, $\boldsymbol{\Theta}$, $\boldsymbol{\Sigma}$
  - Indices in subscript, e. g. $\boldsymbol{\Sigma}_k$

Symbols

- The symbols $\boldsymbol{x}$ (or x for scalar data points) and y denote data points and labels
- The total number of data points in a set is N, where we use n as the index variable, e. g. $\boldsymbol{x}^n$ (or x_n for scarlar data points) is the n-th data point in the training set, y_n the corresponding label $(1 \leqslant n \leqslant N)$
- It should be clear from context whether x_1 references the first scalar data point in a set or the first component of the vector $\boldsymbol{x}$
- We use M to denote the number of features and m as the index variable
- We use K (with index variable k) to denote the total number of classes
- Finally, $\mathcal{C}_k$ denotes the k-th class in a set of K classes

Why Machine Learning?

- *'We are drowning in information and starving for knowledge.'*

  – John Naisbitt

- **Era of big data:**
  - In 2017 there are about **1.8 trillion** web-pages on the internet
  - **20 hours** of video are uploaded to YouTube every minute
  - Walmart handles more than **1 million** transactions per hour and has data bases containing more than **2.5 peta-bytes** (2.5×10^{15}) of information

No human being can deal with this data avalanche!

Why Machine Learning? (Ctd.)

*'I keep saying the sexy job in the next ten years will be **statisticians** and **machine learners**. People think I'm joking, but who would've guessed that computer engineers would've been the sexy job of the 1990s? The ability to take data - to be able to understand it, to process it, to extract value from it, to visualize it, to communicate it - that's going to be a hugely important skill in the next decades.'*

***– Hal Varian**, Chief Economist at Google, 2009*

Definition of Machine Learning

- *'[Machine Learning is the] field of study that gives computers the ability to learn without being explicitly programmed.'*

  ***– Arthur Samuel**, 1959*

- *'A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.'*

  ***– Tom Mitchell**, 1997*

A more abstract Definition

- Our task is to learn a mapping from the input domain $\mathcal{I}$ to the output domain $\mathcal{O}$:

$$h : \mathcal{I} \mapsto \mathcal{O}$$

- Put differently, we want to predict the output from the input
- The symbol 'h' stands for **hypothesis** *(also known as model function)*
- The model function is parameterized by a set of adjustable parameters $\boldsymbol{\theta}$ *(they are learned on a training dataset)*

Model function:

$$\widehat{y} := h(\boldsymbol{x}; \boldsymbol{\theta}) \qquad \text{also:} \quad \widehat{y} := h_{\boldsymbol{\theta}}(\boldsymbol{x})$$

General Paradigm

Training

Testing

Section:

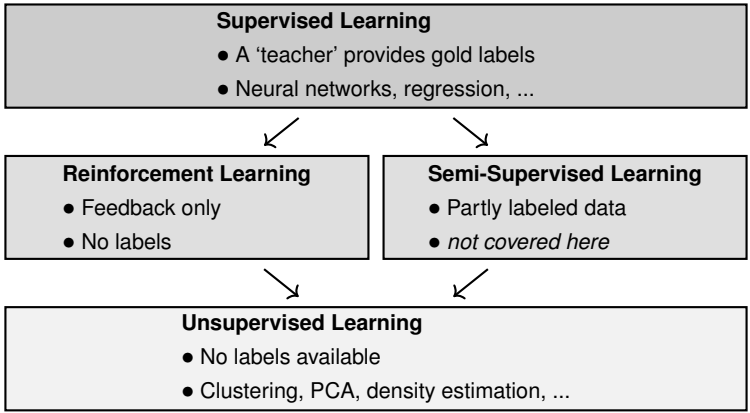
Problem Types in Machine Learning

Type of Training Information
Availability of Training Examples
Type of Target Variable

Type of Training Information

- **Supervised learning**
  - A ‘teacher’ provides **gold labels**
  - E. g. neural networks, decision trees, linear regression
- **Unsupervised learning**
  - The labels are **not** known during training
  - E. g. density estimation, clustering, dimensionality reduction
- **Reinforcement learning**
  - The environment provides rewards for actions, but the correct action is unknown
  - E. g. policy-iteration, value-iteration, Q-learning, SARSA
- **Semi-supervised learning** *(partly labeled data)*

Type of Training Information (Ctd.)

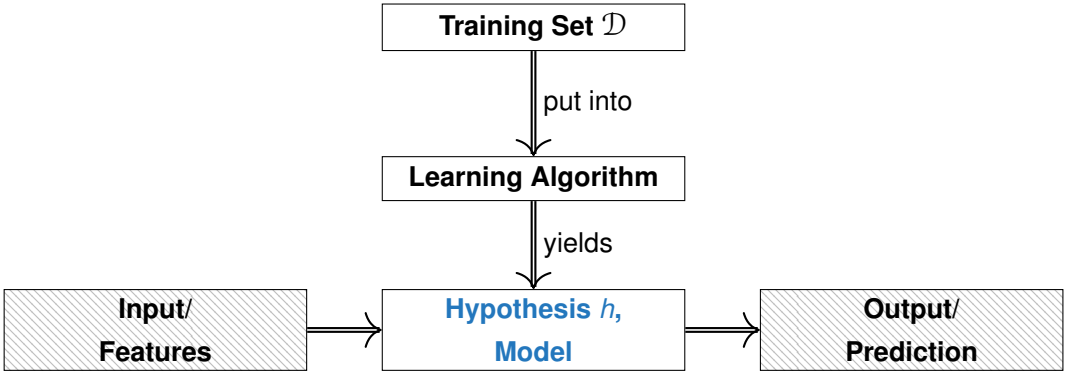


Supervised Learning

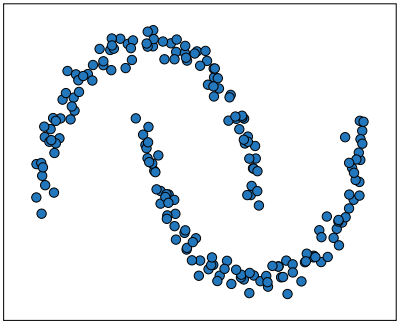
- A single row is called **example** or **instance**
- **Predictors:**
  - `Outlook` $\in$ {sunny, overcast, rainy}
  - `Temperature` $\in$ {hot, mild, cool}
  - `Humidity` $\in$ {high, normal}
  - `Wind` $\in$ {weak, strong}
- **Label:**
  - `PlayGolf` $\in$ {yes, no}
  - Given a new instance, we want to predict its label
- **Label of the new instance???**

Outlook	Temperature	Humidity	Wind	PlayGolf
sunny	hot	high	weak	**no**
sunny	hot	high	strong	**no**
overcast	hot	high	weak	**yes**
rainy	mild	high	weak	**yes**
rainy	cool	normal	weak	**yes**
rainy	cool	normal	strong	**no**
overcast	cool	normal	strong	**yes**
sunny	mild	high	weak	**no**
sunny	cool	normal	weak	**yes**
rainy	mild	normal	weak	**yes**
sunny	mild	normal	strong	**yes**
overcast	mild	high	strong	**yes**
overcast	hot	normal	weak	**yes**
rainy	mild	high	strong	**no**
rainy	mild	normal	strong	**???**

Supervised Learning: General Approach



Unsupervised Learning



- There are **no** labels
- Try to find regularities in the data
- Examples for unsupervised learning:
  - **Density estimation**
  - **Clustering**
  - **Dimensionality reduction**

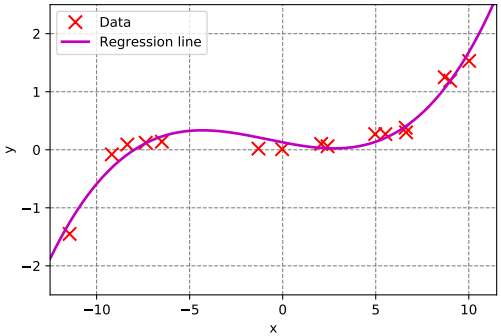
Availability of Training Examples

- **Batch learning**
  - The learner is provided with a **fixed set** of training examples
  - Cf. weather dataset
  - E. g. neural networks, decision trees
- **Incremental / online learning**
  - **Constant stream** of training examples
  - The model is updated as new training examples arrive
  - E. g. k-nearest-neighbors
- **Active learning** (*not covered*)

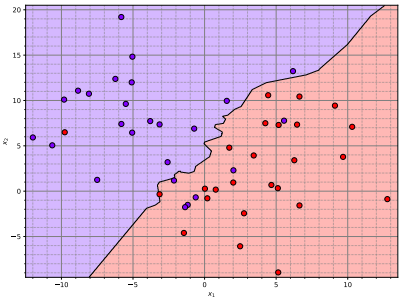
Type of Target Variable: Regression

Regression

- Learn a mapping into a **continuous space**, e. g.
  - $\mathcal{O} = \mathbb{R}$
  - $\mathcal{O} = \mathbb{R}^3$
- E. g. curve fitting, financial analysis, housing prices, ...



Type of Target Variable: Classification



Classification

- Learn a mapping into a **discrete space**, e. g.
  - $\mathcal{Y} = \{0, 1\}$ (binary classification)
  - $\mathcal{Y} = \{0, 1, 2, 3, ...\}$
  - $\mathcal{Y} = \{\text{verb, noun, adverb, ...}\}$
- Examples:
  - Spam / no spam (ham)
  - Digit recognition
  - Part of speech tagging

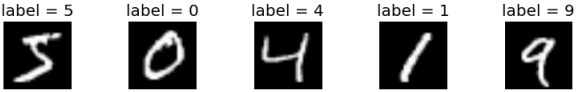
Section:

Key Challenges in Machine Learning

Generalization from Training Data
Feature Selection / Feature Engineering
Performance Measurement
Model Selection
Computation

Generalization from Training Data

- Learning does not mean memorizing the training data by heart
- What if we see input that we **haven't seen before**?
- Example OCR (**O**ptical **C**haracter **R**ecognition):



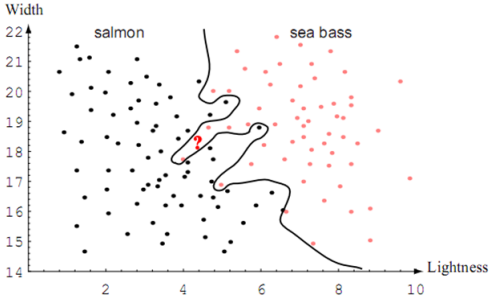
(Hand-written digits from the *MNIST* dataset)

  - Predict the character given the input image
  - **People have different hand-writings**

Generalization from Training Data (Ctd.)

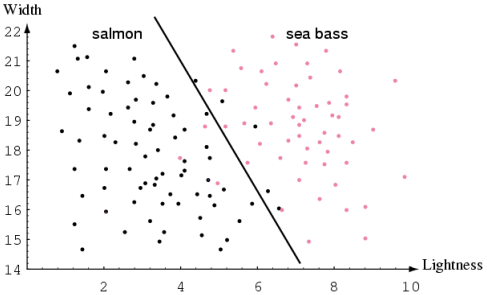
What is the problem here?

- Complex decision boundary
- This leads to **Overfitting**
  - The model is too expressive...
  - ...and adapts to **idiosyncrasies** of the training data



Solution: Choose a simpler model (cf. **Occam's razor**)

Generalization from Training Data (Ctd.)



- Linear (less complex) model
- Allow for **misclassifications** of some training examples
- **Better generalization** to unseen instances

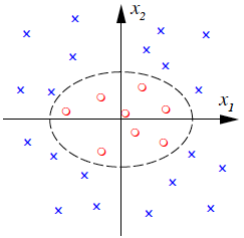
A prominent Example of Overfitting

Choosing the right Features

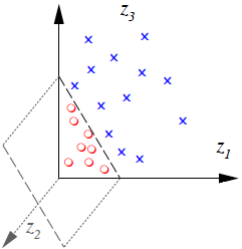
'When stuck, move to a different perspective!'

$\mathbb{X}$

Input space



$\varphi : \mathbb{X} \mapsto \mathbb{F}$



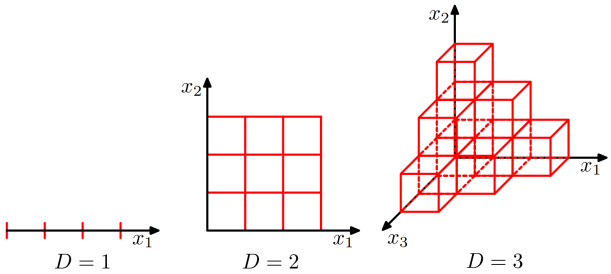
$\mathbb{F}$

Feature space

$$\varphi(x_1, x_2) = \left(x_1^2, \sqrt{2}x_1x_2, x_2^2\right) =: (z_1, z_2, z_3)$$

Choosing the right Features

But: Beware of the curse of dimensionality!



- Too many features significantly **slow down** the machine learning algorithm
- It requires an **exponential** amount of training data
- Dimensionality reduction might be useful!

cf. BISHOP.2006, page 35

Performance Measurement

- How do we measure performance?
  - 99 % correct classification in speech recognition: What does it really mean?
  - *We understand the meaning of the sentence?*, *We understand every word?*, *For all speakers?*
- We need more **concrete numbers**:
  - % of correctly classified letters
  - Average distance driven (until accident, ...)
  - % of games won
  - % of correctly recognized words, sentences, etc.
- **Training vs. testing performance**

Training vs. Testing Performance

- Evaluate on data which was **not used** for training **(out-of-sample testing)**
- **Two-way split:**

  `Train - Test`
- Even better: **Three-way split:**

  `Train - Dev - Test`

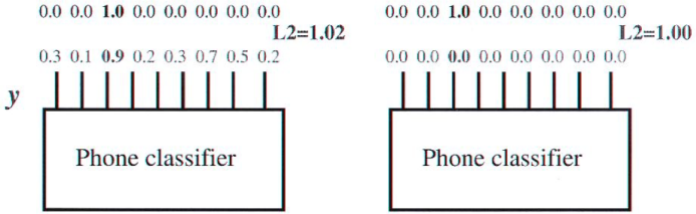
  `Train:` Train model

  `Dev:` Tune hyperparameters

  `Test:` Test final model

Performance Measurement (Ctd.)

- We also need to define the right **error metric**:

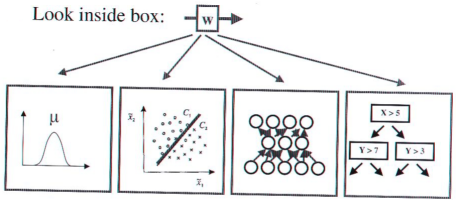


- Which is better?
- EUCLIDean distance (L2-norm) might be useless

Model Selection

- What is the **right model**?
- The learned parameters (here: $\boldsymbol{w}$) can mean a lot of different things:

- $\boldsymbol{w}$ may characterize a family of functions
- $\boldsymbol{w}$ may be parameters of a probability distribution
- $\boldsymbol{w}$ may be a vector, adjacency matrix, graph, etc.



Computation

Even if the other problems are solved, **computation is usually quite hard**

- Learning involves optimization of model parameters θ
- Search for the best model parameters
  - Often GPUs (**G**raphics **P**rocessing **U**nit) are needed
  - Google invented TPUs (**T**ensor **P**rocessing **U**nit)
- Often we have millions of parameters (GoogleNet has approximately 6.5 million parameters)
- Often we have to deal with thousands, millions, ... of training examples
- Given a model, the prediction has to be computed efficiently

Section:

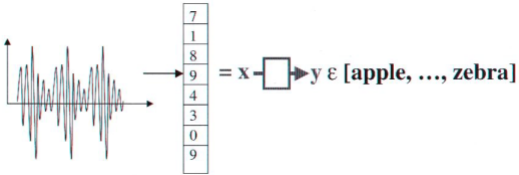
Machine Learning Applications

Natural Language Processing
Computer Vision
Robotics

Applications in Natural Language Processing

- **E-mail filtering:**

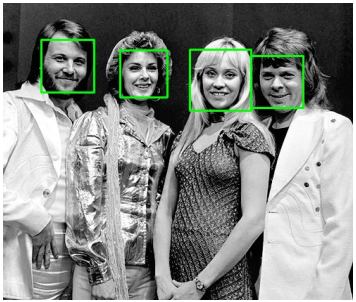
- **Speech Recognition:**



ChatGPT

Applications in Computer Vision

Face detection:



Traffic sign detection:

Applications in Computer Vision (Ctd.)

Optical character recognition:

Cf. demonstration LeNet

Applications in Robotics

Robot control:

Autonomous driving:

Section:

Wrap-Up

Summary
Recommended Literature
Self-Test Questions
Lecture Outlook

Summary

- Vast amounts of data are prohibitive for manual inspection
- Machine learning algorithms learn **without being explicitly programmed**
- **Important distinctions:**
  - Supervised learning $\Longleftrightarrow$ Unsupervised learning
  - Classification $\Longleftrightarrow$ Regression
- There are several **challenges** to be mastered
  - Generalization
  - Feature selection
  - Performance measurement
  - Model selection
  - Computation

Recommended Literature

1. [BISHOP.2006], chapter 1
2. [MURPHY.2012], chapter 1

(For free PDF versions, see list in GitHub readme!)

Self-Test Questions

1. What is the difference between supervised and unsupervised learning?
2. What is regression?
3. What does generalization mean?
4. *'The more features the better.'* Is this statement correct or not? Give reasons for your answer.
5. Why do we need `train`, `dev` and `test` sets?
6. True or false: 100 % train accuracy is desirable.
7. State some applications of machine learning.

What's next...?

	I	Machine Learning Introduction
•	II	Optimization Techniques
	III	Bayesian Decision Theory
	IV	Non-parametric Density Estimation
	V	Probabilistic Graphical Models
	VI	Linear Regression
	VII	Logistic Regression
	VIII	Deep Learning
	IX	Evaluation
	X	Decision Trees
	XI	Support Vector Machines
	XII	Clustering
	XIII	Principal Component Analysis
	XIV	Reinforcement Learning
	XV	Advanced Regression

Thank you very much for the attention!

*** * * Artificial Intelligence and Machine Learning * * ***

Topic: Machine Learning Introduction

Term: Summer term 2025

Contact:
Daniel Wehner, M.Sc.
SAP SE / DHBW Mannheim
daniel.wehner@sap.com

Do you have any questions?