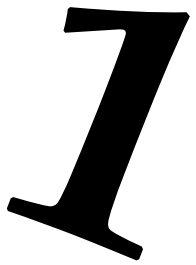
Part 1

Introduction to the course

This is the number?

This is the number?



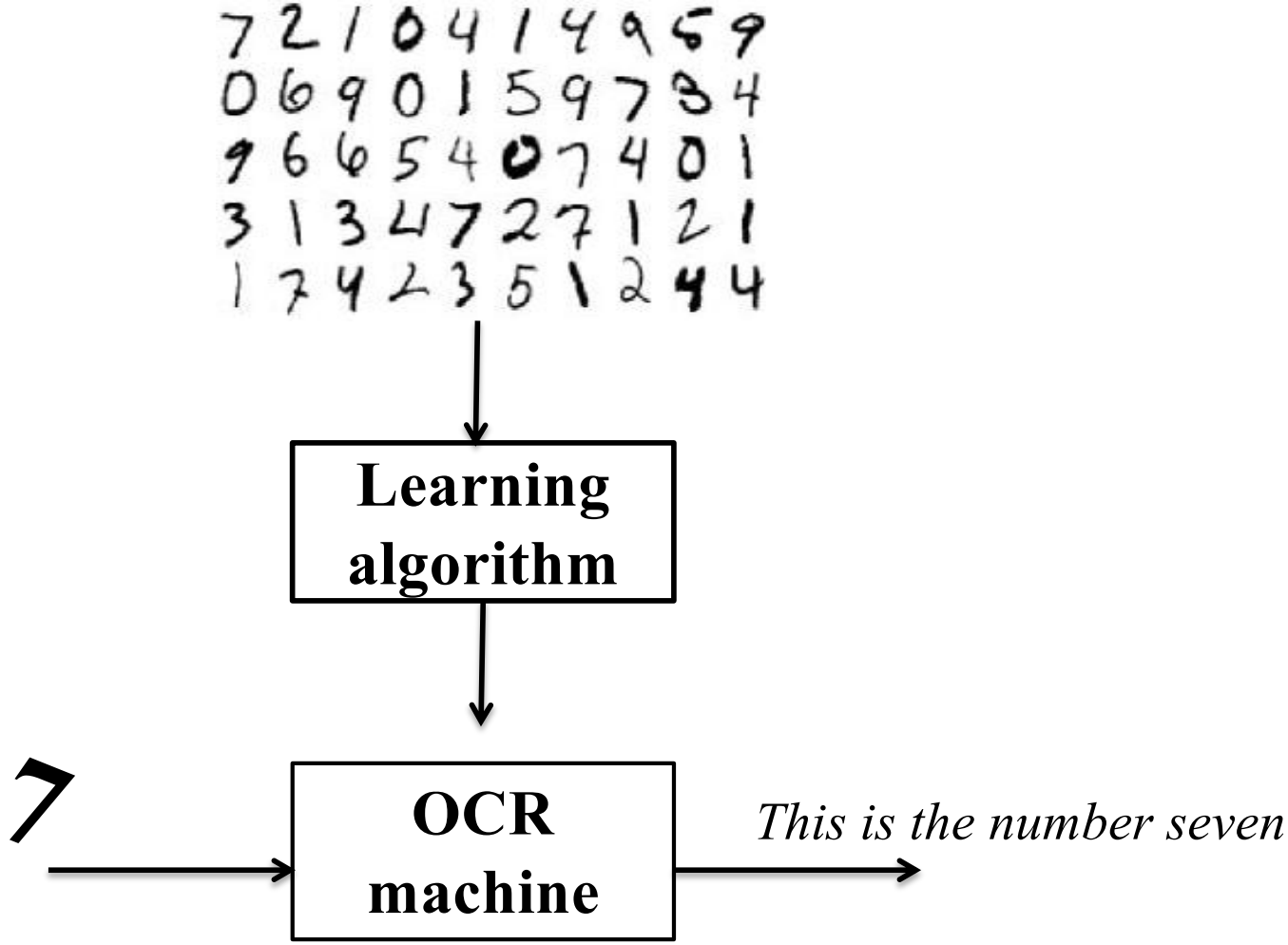
This is the number?

Are you able to write in Python (or any other language) the **exact algorithm** (step after step) that you use to recognize the above numbers?

Writing a **deterministic** algorithm to recognize numbers from images is very difficult…

But we can collect easily many example images…

If we could design a machine that learns from examples…



So, what is machine learning ?

Machine learning is the technology that we use to solve a problem by learning the solution by examples

"The goal of machine learning is to build computer systems that automatically improve with experience"

[Tom M. Mitchell, The discipline of Machine Learning, 2006]

First take-home message

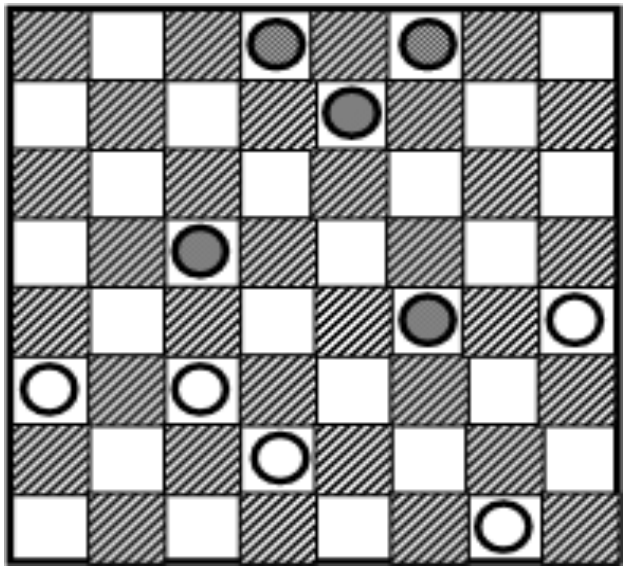
Machine learning is very useful when no ***algorithmic solution*** *is known. It also avoids a detailed algorithm to overfit known cases, reducing errors*

Second take-home message

When you are able to devise algorithmic solutions (step after step through every possible corner case) that work 100% of the time, you should ***not use*** *machine learning !*

Machine learning at the beginning…

Arthur Samuel (1959) wrote a program that **learnt** to play draughts ("checkers" if you're American).

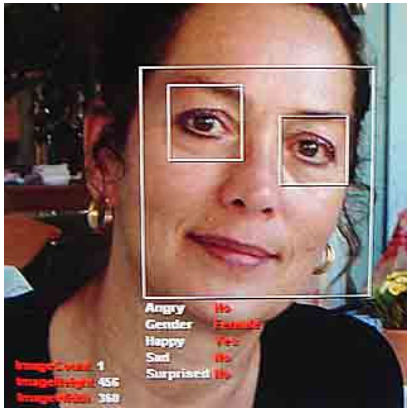


R.S. Michalski,J.G. Carbonell,T.M. Mitchell, Machine Learning: An Artificial Intelligence Approach, 1985

What is machine learning today?

It is mostly learning from (big) data for recognizing patterns

amazon.com



Google

Google Privacy & Terms

Overview Privacy Policy Terms of Service Technologies and Principles FAQ My Account

Technologies

Advertising

How Google uses cookies

How Google uses pattern recognition

How Google uses pattern recognition

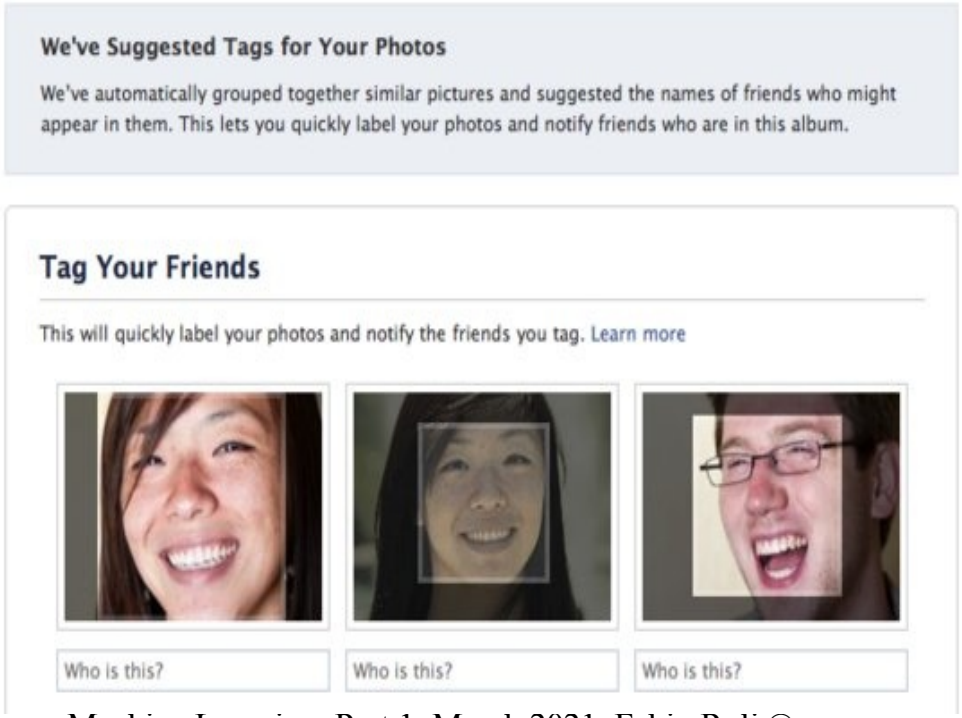
How Google uses pattern recognition to make sense of images

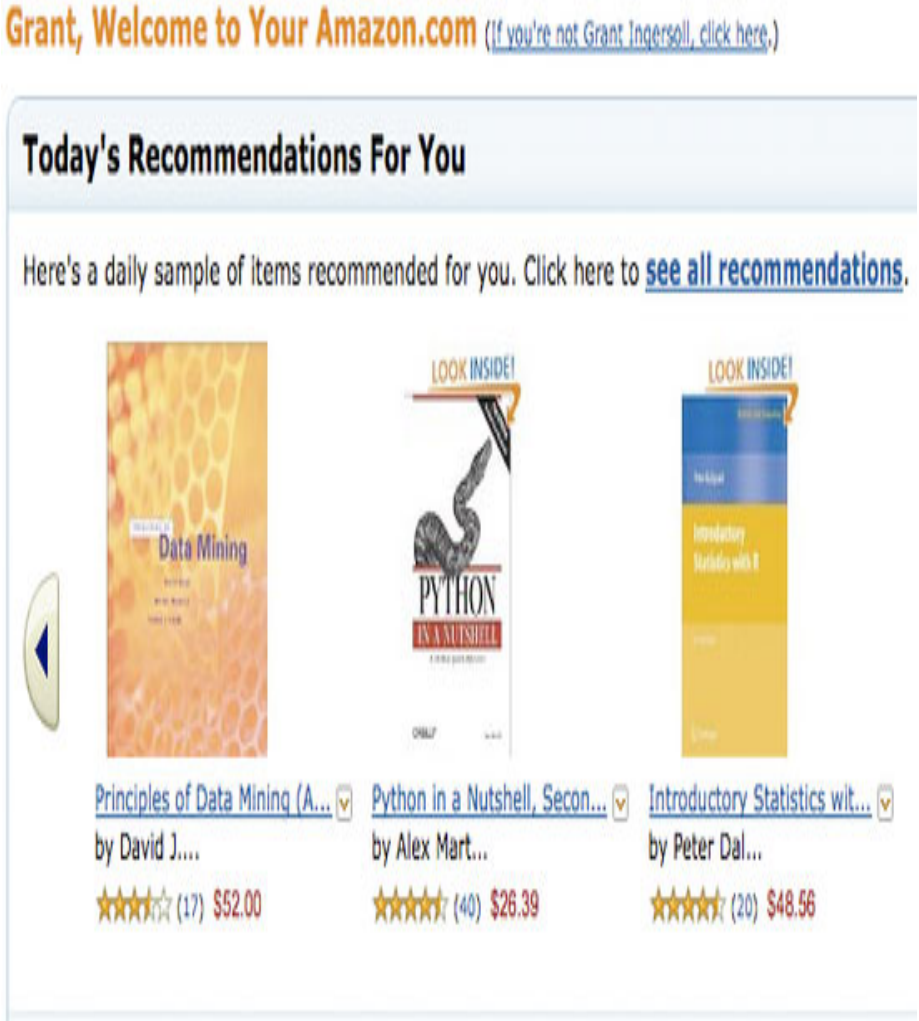
Computers don't "see" photos and videos in the same way that people do. When you look at a photo, you might see your best friend standing in front of her house. From a computer's perspective, that same image is simply a load of data that it may interpret as shapes and information about colour values. While a computer won't react like you do when you see that photo, a computer can be trained to recognise certain patterns of colour and shapes. For

Baidu 百度

What is machine learning today?

It is **Pattern Recognition** !





Today machine learning is pattern recognition

Therefore, this course is focused on machine learning for **pattern recognition** (often called pattern **classification**), on the design of **learning-based machines** for **pattern recognition**

In the next slides, I explain you briefly what pattern recognition is…

What is pattern recognition?

- During their evolution, animals and human beings have developed sophisticated skills for recognition of "patterns" acquired by their sense organs.
- Skills for fast recognition of predators, fast detection of features which distinguish friends and foes, etc.
- Pattern recognition skills have been initially developed to struggle for existence, afterwards they have been refined to develop high-level abilities (e.g., writing, painting).

- It is worth noting that pattern recognition is an activity largely **subconscious** for human beings.

The challenge of pattern recognition

[Theo Pavlidis, Why general AI is so hard?, http://theopavlidis.com]

We would like to replicate with computers complex transformations that the human/animal brain has evolved over millions of years.

We have to deal with the fact the pattern-recognition processing is not unidirectional and also affected by other factors than the input (the **"context"**).

What Do You See?

[Theo Pavlidis, Why general AI is so hard?, http://theopavlidis.com]

Context exploitation in human beings

[Theo Pavlidis, Why general AI is so hard?, http://theopavlidis.com]

Tentative binding on the letter shapes (bottom up) is finalized once a word is recognized (top down). Word shape and meaning over-ride early cues.

The challenge of pattern recognition

[Theo Pavlidis, Why general AI is so hard?, http://theopavlidis.com]

The human visual system does pattern recognition incredibly well

But it has evolved from animal visual systems over a period of more than 100 million years.

Should we try to replicate it as it is?

- All in all, what should we mean when we speak about building a machine that does pattern recognition?

An engineering definition of pattern recognition

- Pattern recognition aims to build machines able to recognize patterns like aereonautical engineering aims to build airplanes able to fly

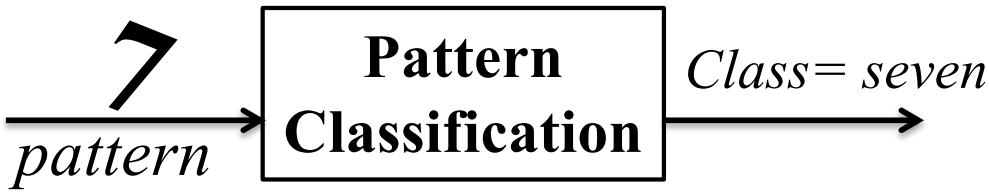
Pattern recognition can be defined as the scientific discipline that studies theories and methods for designing machines that are able to recognise patterns in noisy data…omiss…Pattern recognition has an "engineering" nature, as its final goal is the design of "machines" (R.P.W. Duin, F. Roli D. de Ridder, Pattern Recognition Letters, 2002)

- Nowadays, replicating the human pattern recognition performance for a large variety of tasks (building a **general** pattern recognition system) is still impossible.
- But we can be successful on limited and well understood tasks!

Pattern recognition as "classification"

- This machine learning course focuses on **pattern classification**. We use the term recognition instead of classification if the context makes the meaning clear and there is no ambiguity.

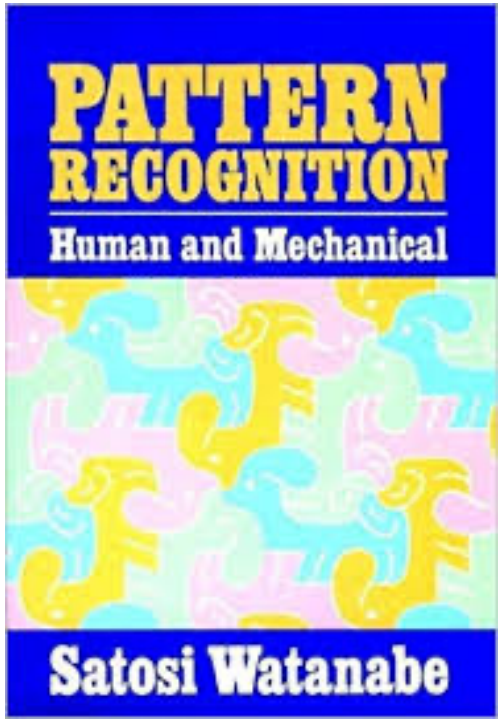
  – Pattern Classification: assigning a *"pattern" (a particular grouping of data)* to a category/class



In this picture, the pattern is the particular **grouping** of pixels that represent the number seven !

What is a pattern…

Satosi Watanabe defined pattern recognition as *"seeing one in many"*, namely, the capability of recognizing the *unity* in the *multiplicity*, the capability of recognizing one face in a huge collection of pixels or recognizing the concept of tree despite the huge variety of sizes, shapes, and colours of the different individualities.



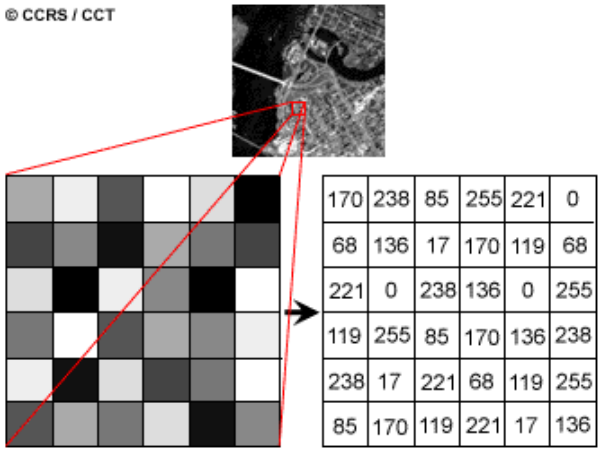
[Satosi Watanabe, Pattern Recognition: Human and Mechanical, 1985]

Pattern recognition as “classification”

Pattern classification is about assigning class *labels* to patterns.

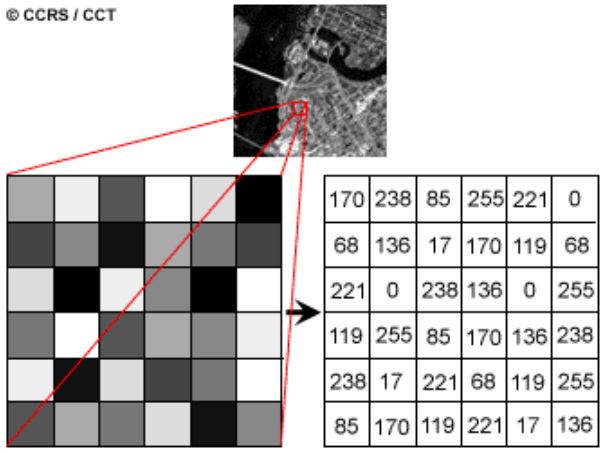
Patterns are described by a set of measurements called also **features** (or attributes, inputs).

- If we are working with image data, **feature values** could correspond simply to the **brightness** of each **pixel**.



Basic concepts: class and feature

In this course, we assume that each pattern is described by a feature vector with "d" elements: $\mathbf{x} = (x_1, x_2, \ldots, x_d)$.



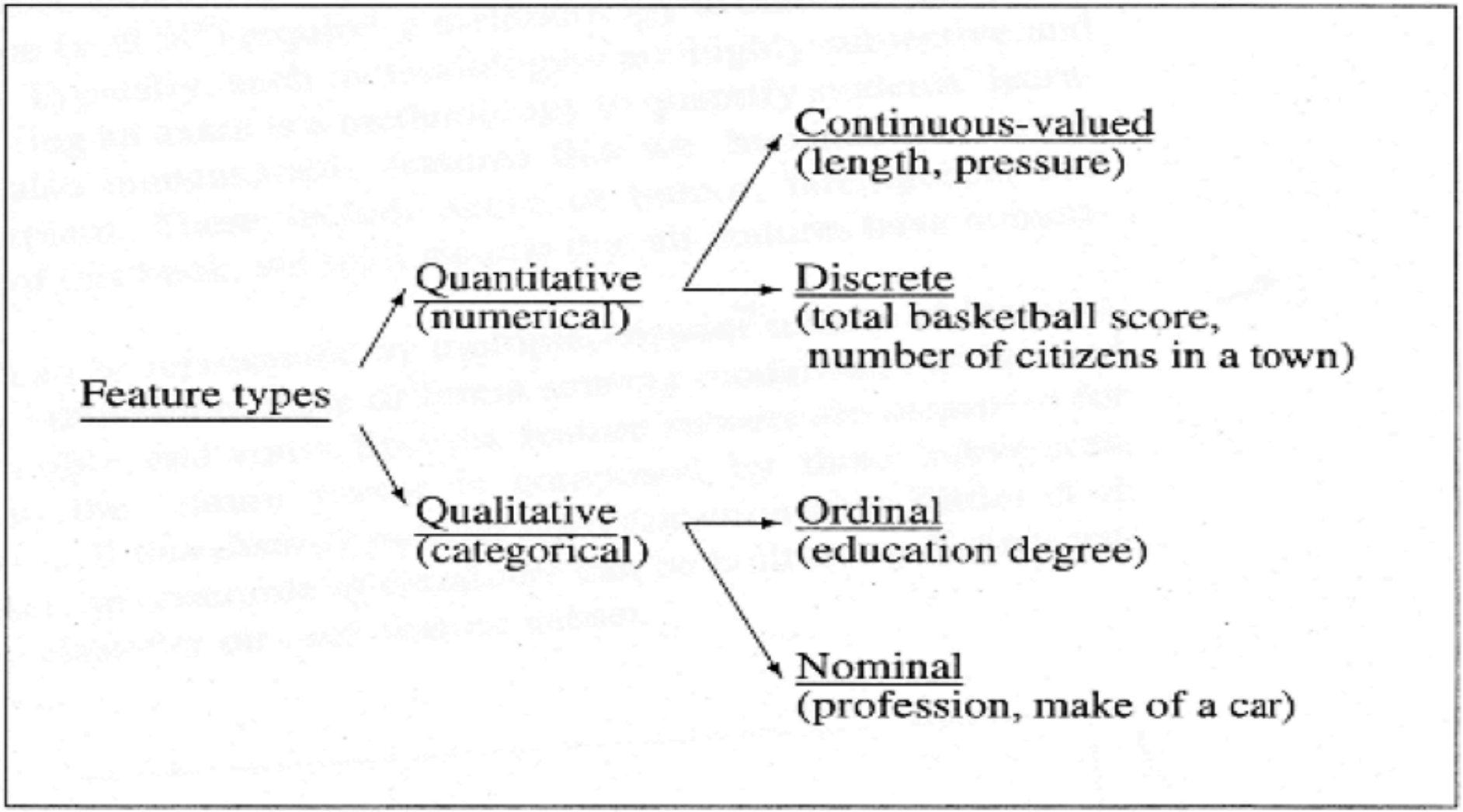
$\mathbf{x} = (x_1, x_2, \ldots, x_d) = (170, 238, 85 \ldots 136)$

Class: intuitively, a class contains similar patterns, whereas patterns from different classes are dissimilar (e.g., dogs and cars).

In this course, we assume that there are *c* possible classes, and we denote that as: $\boldsymbol{\Omega = \{\omega_1, \omega_2, \ldots, \omega_c\}}$, each pattern belongs to one of the "c" classes of the set Ω. We say that each pattern has a class **label**.

Different feature types

[L. Kuncheva, Combining pattern classifiers, Wiley, 2004]

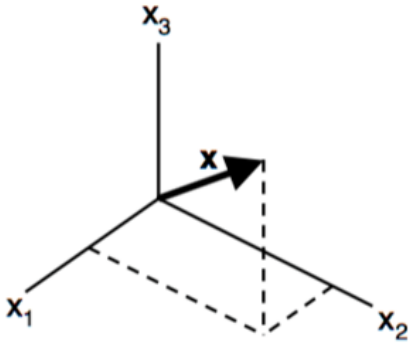


- Statistical pattern classification uses **numerical** features.

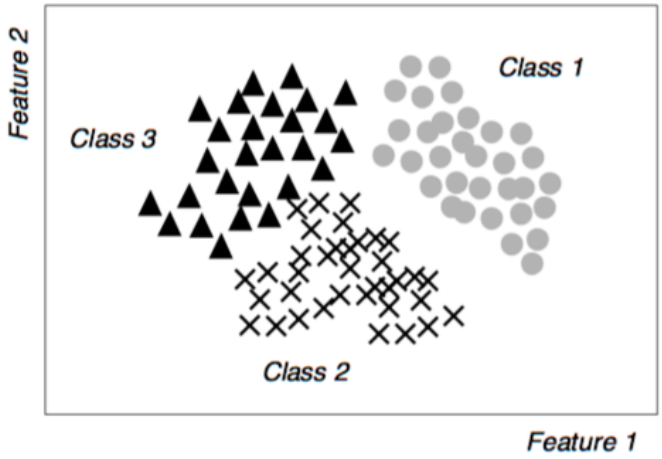
Basic concepts: feature vector, feature space

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \\ x_d \end{bmatrix}$$

Feature vector

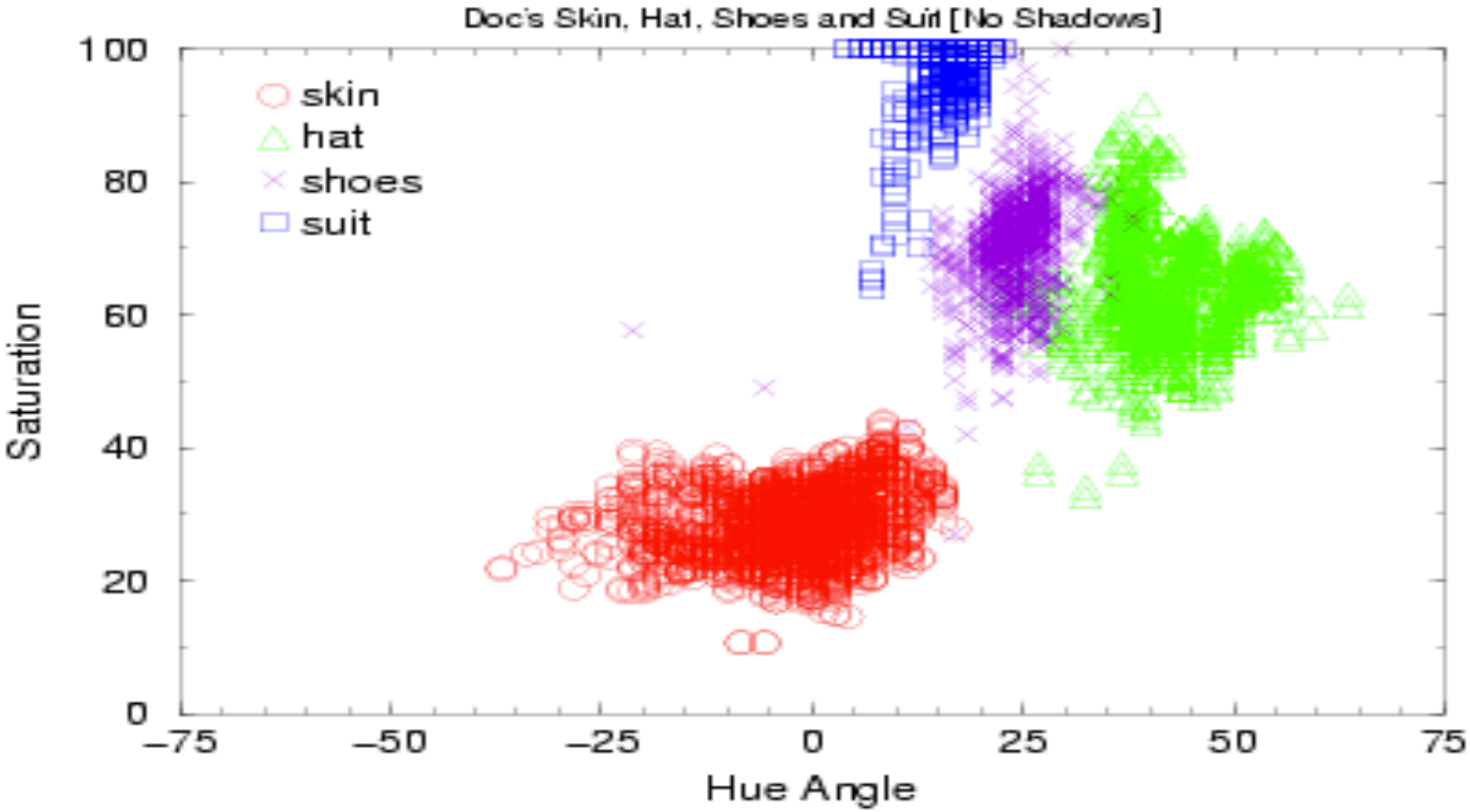


Feature space (3D)



Scatter plot (2D)

Basic concept: feature space



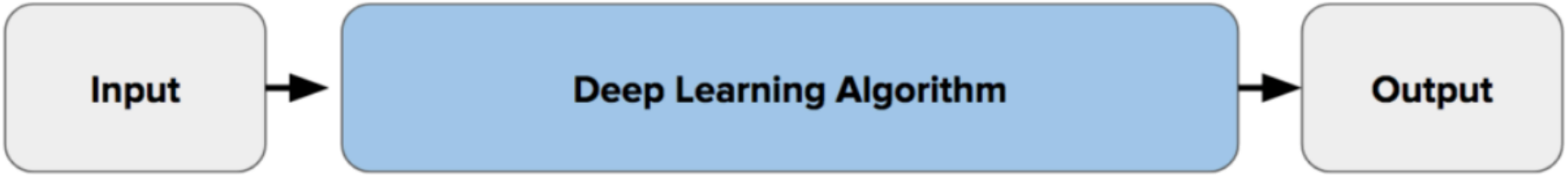
The feature values are arranged as a d-dimensional vector.

The real space is called the **feature space**, each axis corresponding to a physical feature.

Hand-crafted vs. non-handcrafted (learned) features

- In the previous example, we have seen what is named **«handcrafted»** features that are manually engineered by the human designer.
- Today, we can extract **non-handcrafted** features that are automatically learned from a machine learning algorithm.

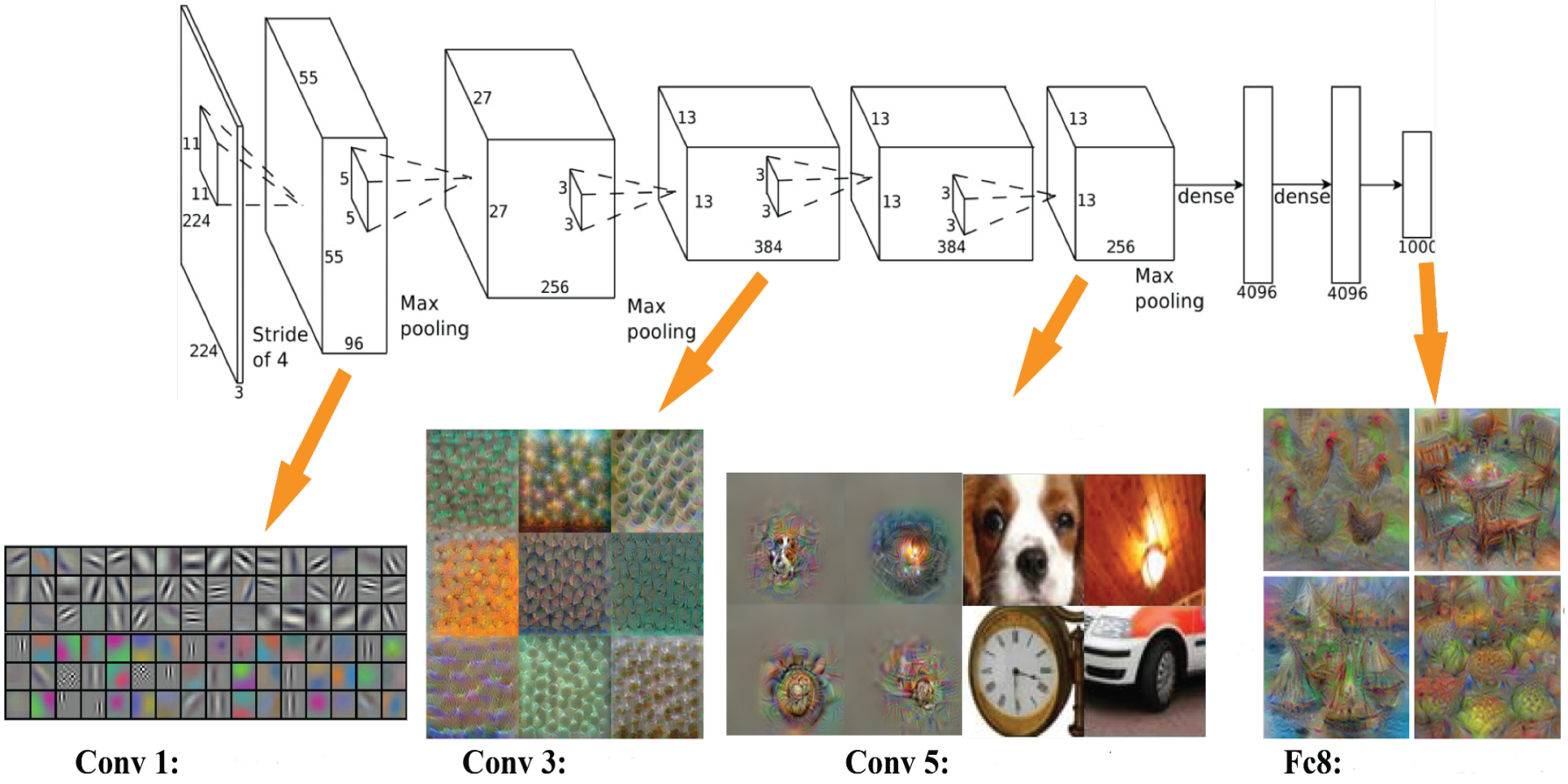
Processing flow for extraction of **handcrafted»** features



Processing flow for learning **non-handcrafted** features («learned» features)

Learning non-handcrafted features

- **Non-handcrafted** features can be automatically learned with **deep neural networks** (we will see them later).

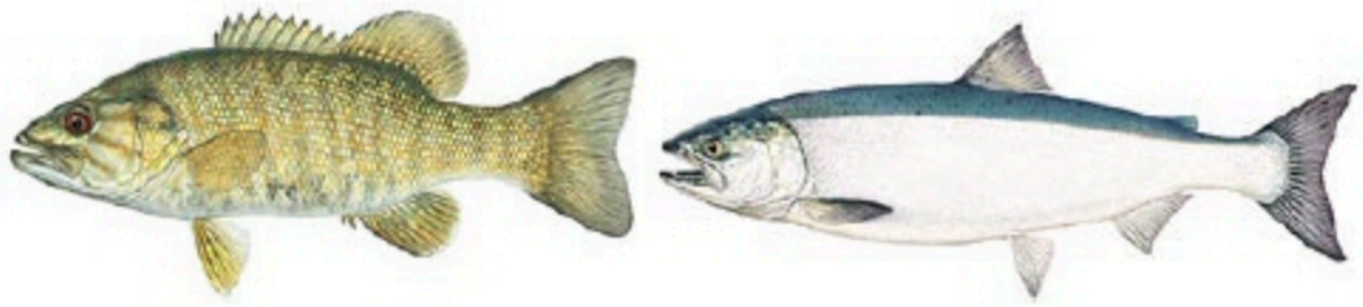


Classification model

[Pattern Classification, R. O. Duda, P. E. Hart, D. G. Stork, John Wiley & Sons, 2000]

- **Classification:** after the extraction of a set of features to characterize patterns, we should select a **classification model** using such features to classify patterns.

- Let us assume that want to recognize 2 classes of fish: **salmon** and **sea bass**.
- We use only one feature: **lenght** value (random variable l).



Sea bass Salmon

Classification model

•A very simple classification **model** based on a simple heuristic rule could be:

- *A sea bass is generally longer than a salmon*
- We can rewrite more formally this heuristic rule as follows:

  if $l > l^*$ *then* fish=sea bass , *else* fish=salmon

•The threshold value l^* can be an heuristic value that we know, otherwise we should estimate it.

•*How can we estimate* l^* *?* We need a set of samples/examples of the two fish types (called **"design o training set"**)

Basic concept: design or training data set

The information to design a pattern classifier is usually in the form of a labeled data set ***D*** (called design or training set):

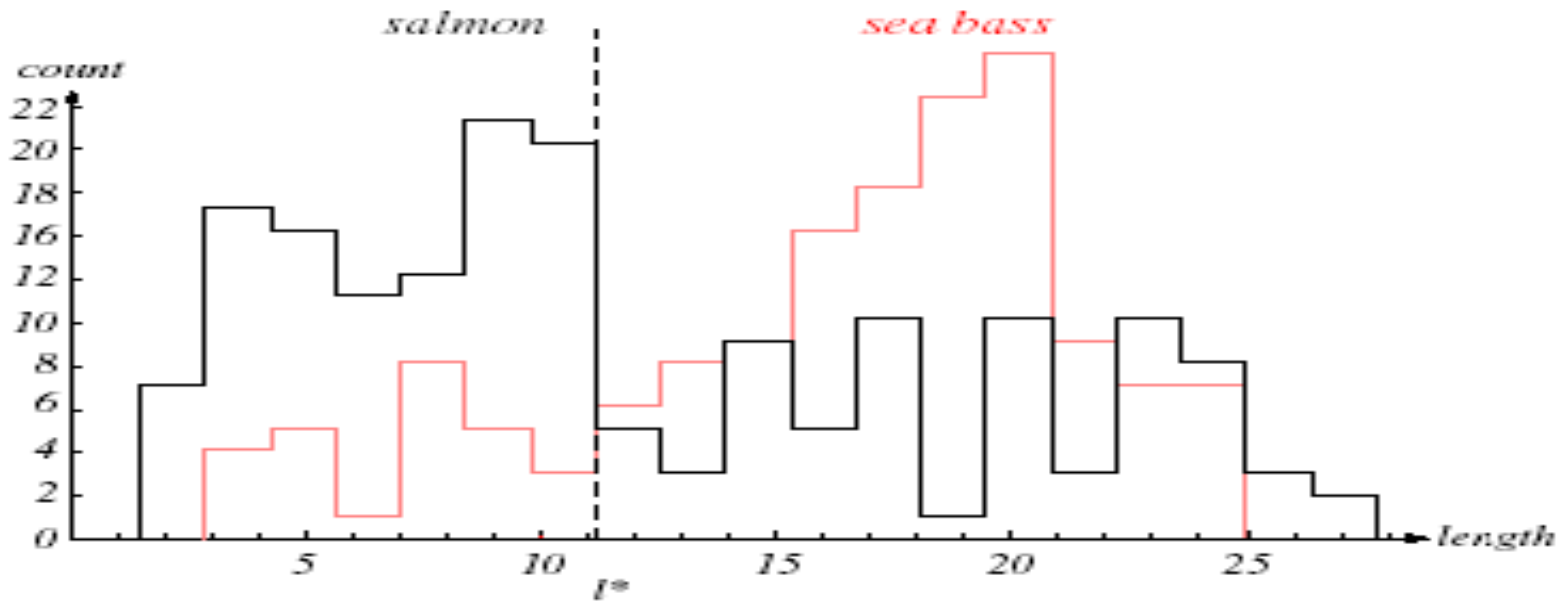
$$\boldsymbol{D} = [\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_n}]$$

$$\boldsymbol{x}_i = (\boldsymbol{x}_{i1}, \boldsymbol{x}_{i2}, \ldots, \boldsymbol{x}_{id}) \text{ i=1,..,n}$$

$\boldsymbol{x}_i$ belongs to one of the "c" classes ($\boldsymbol{x}_i \, \varepsilon \, \omega_j \;\; j=1,\ldots,c$)

In the previous example, ***D*** is the data set used to compute the empirical distributions of the length of the two fish types.
This allows us to estimate the threshold value l* that discriminates between salmon and sea bass.

Classification models



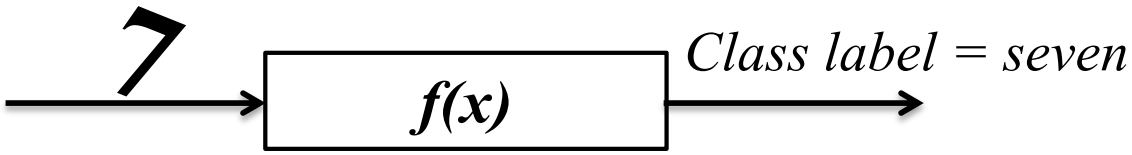
This simple example suggests us a more general classification model. We could estimate the two probability functions:

P(lenght / salmon) and *P(lenght / sea bass)*

and then make a probabilistic decision…

- We will discuss this in greater detail in Part 2 !

Classification models in general…

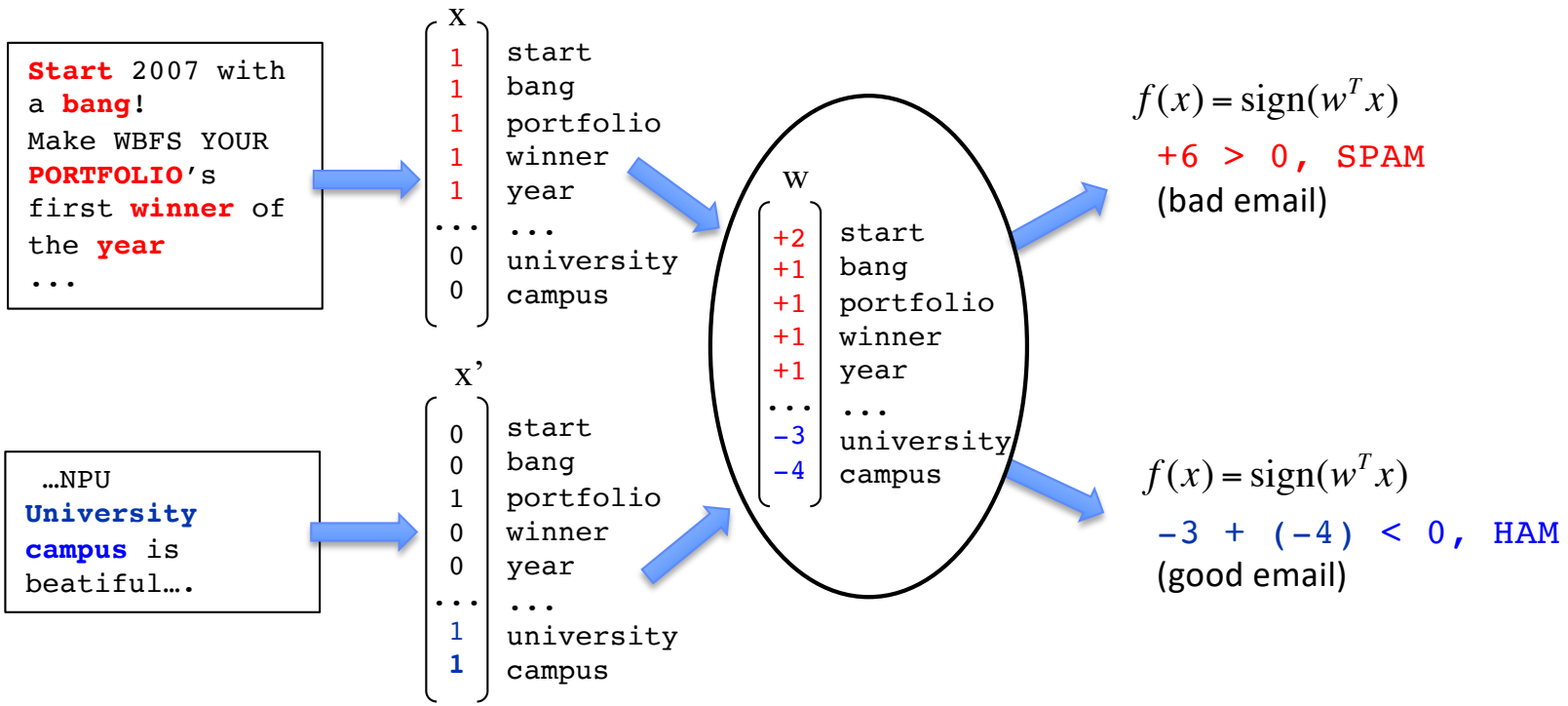


In general, a classification model can be regarded as a **function** $f(\boldsymbol{x})$, that takes as input the vector $\boldsymbol{x}$ (representing the pattern) and provides as output the classification (class label)

For example, the classification model could be a **linear function**: $f(\boldsymbol{x}) = \boldsymbol{w}^T\boldsymbol{x} + b = \sum_{j=1}^{d} w_j x_j + b$

Example: Spam filtering

$$f(\boldsymbol{x}) = sign(\boldsymbol{w}^T \boldsymbol{x})$$



Learning as optimization

We said that machine learning is “learning from experience”.

In other words, improving classification performances over time

How do we evaluate if we are improving?

In order to develop a formal mathematical system of learning machines, we need to have formal measures of how good (or bad) our models are.

To this end, we use ***loss functions*** (or *cost functions)* to evaluate how good (or bad) our classification models are.

Example of loss function

$$L(D, \boldsymbol{\theta}) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(\boldsymbol{x}_i; \boldsymbol{\theta}))$$

D: training set containing «n» examples

y_i: is the class label for training example $\boldsymbol{x}_i$

$f(\boldsymbol{x}_i; \boldsymbol{\theta})$ is the classification model

$\ell(y_i, f(\boldsymbol{x}_i; \boldsymbol{\theta}))$ could be the zero-one loss function

- equal to 0 for correct predictions and 1 otherwise

$$\ell\big(y_i, f(\boldsymbol{x}_i; \boldsymbol{\theta})\big) = \begin{cases} 0, classification\ is\ correct \\ 1, classification\ is\ incorrect \end{cases}$$

Learning as an Optimization Problem

Given a **linear function**: $f(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + b = \sum_{j=1}^{d} w_j x_j + b$

How do we estimate the classifier parameters $\boldsymbol{w}$ and b?

Modern approaches formulate the learning problem as an **optimization problem**

- This is generally true also for nonlinear classification functions $f(\boldsymbol{x}; \boldsymbol{\theta})$, including modern deep-learning approaches and neural networks

$$\boldsymbol{w}^\star, b^\star = \underset{\boldsymbol{w},b}{\operatorname{argmin}} \underbrace{\frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(\boldsymbol{x}_i))}_{\text{loss term } L(D,\boldsymbol{\theta})} + \underbrace{\lambda\, \Omega(\boldsymbol{w})}_{\text{regularization term } \Omega(\boldsymbol{w})}$$

λ: regularization hyperparameter

Optimization algorithms

$$\boldsymbol{w}^{\star}, b^{\star} = \underset{\boldsymbol{w},b}{\operatorname{argmin}} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(\boldsymbol{x}_i)) + \lambda\, \Omega(\boldsymbol{w})$$

- In machine learning, we need an optimization algorithm (also called ***«solver»***) capable of searching for the best possible parameters for minimizing the loss function.
- The most popular optimization algorithms follow an approach called **gradient descent**.

- We will discuss it later…

Generalization error, overfitting

- The best values of our model's parameters are learned by minimizing the loss incurred on a ***training set*** consisting of some number of *examples* collected for training. However, doing well on the training data does not guarantee that we will do well on (**unseen**) **test** data. So we will typically want to split the available data into two partitions: the training data (for fitting model parameters) and the test data (which is held out for evaluation), reporting the following two quantities:

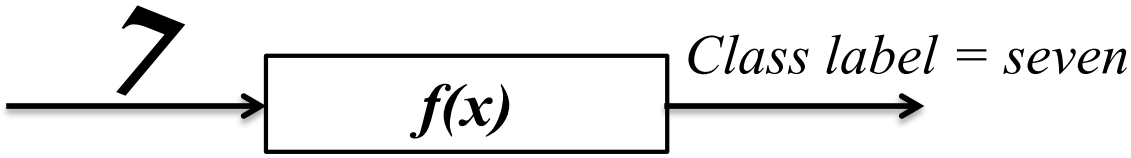
- **Training Error:** The error on that data on which the model was trained.

- **Test Error:** This is the error incurred on an **unseen** test set (**generalization error**). This can deviate significantly from the training error. When a model performs well on the training data but fails to generalize to unseen data, we say that it is ***overfitting***.

[https://d2l.ai, Chapter 1]

Two main kinds of Machine Learning

Supervised learning



(in this course, we mainly focus on this kind of learning and, in particular, on supervised pattern classification)

Unsupervised learning

Learning from a set of unlabeled samples. The goal of **unsupervised learning** (also called "clustering") is basically to find groupings in the data ("clusters") which actually reflect the ground truth and the "natural properties" of the domain the data comes from.

Other kinds of machine learning problems

[https://d2l.ai, Chapter 1]

Regression

- for example, predicting the rating that a user will assign to a movie can be thought of as a regression problem

Tagging

- for example, assigning muliple lables to one image can be thought of as a tagging problem

Search and ranking

- for example, determining whether a particular web page is relevant for a user's query can be thought of as a search and ranking problem

Recommendation

- for example, providing movie recommedations to web users can thought of as a recommendation problem

Other kinds of machine learning problems

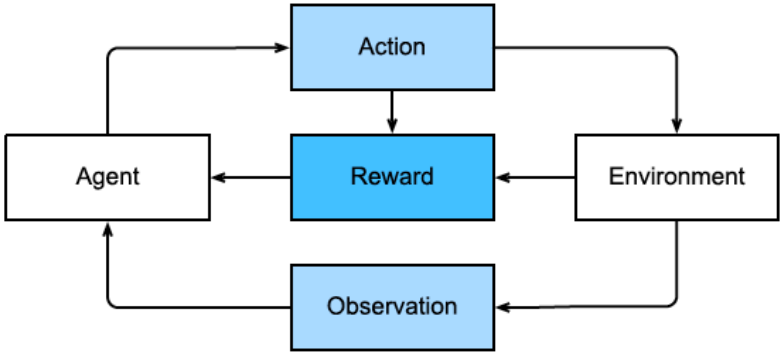
[https://d2l.ai, Chapter 1]

Sequence learning

- When you have a sequence of inputs and you have to provide a sequence of outputs; for example, speech recognition, text-to-speech, language translation, can be thought of as a sequence learning problems

Reinforcement learning (learning by interacting with an environment)

- Game of chess, driving a car, can be thought of as reinforcement learning problems



Course objectives and outcome

Objectives

The objective of this course is to provide students with the fundamental elements of **machine learning** and its applications to **pattern recognition**. The main concepts and methods of machine learning and statistical pattern recognition are presented, as well as basic methods to design and evaluate the performance of a pattern recognition system.

Outcome

An understanding of fundamental concepts and methods of machine learning, statistical pattern recognition and its applications. An ability to analyse and evaluate simple algorithms for pattern classification. An ability to design simple algorithms for pattern classification, code them with Python programming language and test them with benchmark data sets.

Machine Learning (6 CFU) - Tentative course outline

1. Introduction (2 hours)
2. Bayesian Decision Theory and Gaussian Pattern Classifiers (10 hours)
3. Non parametric methods and k-nn classifier (4 hours)
4. Linear discriminant functions and support vector machines (6 hours)
5. Artificial neural networks (4 hours)
6. Performance evaluation (2 hours)
7. Clustering Methods (2 hours)
8. Adversarial machine learning (2 hours)
9. Exercises (12 hours)
10. Python Programming language and computer exercises (16 hours)

Course grading and material

- Home computer-exercise assignment + Oral examination
  - You can do intermediate assessments instead of the oral examination
  - You can do the oral examination only after the computer exercise
  - Teams of 3 students maximum can do the home computer exercise
- **Grading policy = Computer exercise (10/30) + Oral examination (20/30)**

- **Reference book:** Pattern Classification (2^ edizione), R. O. Duda, P. E. Hart, e D. G. Stork, John Wiley & Sons, 2000
- All the course material is available on the web site
- **Course web site:** https://unica-ml.github.io/#

Homework and bonus policy

•If you do your **homework** well and send them to the instructors within 1 week from the date of assignment of the homework, you can obtain 1/30 additional score at the end of the course. **Doing homework is optional, not mandatory!** It's just an incentive to study on a regular basis, very useful if you want to do intermediate assessments. The instructors will evaluate homework at the end of the course and will assign or not the 1/30 additional score. **No guarantee to get the 1/30 additional score !**

•Bonus: sometimes the instructors assign homework with a "bonus". The first three students who send right answers get the bonus (usually, one additional score between 0.5/30 and 1/30, higher than 1/30 for very difficult homework).

Homework 1a - send me your answer to roli@unica.it

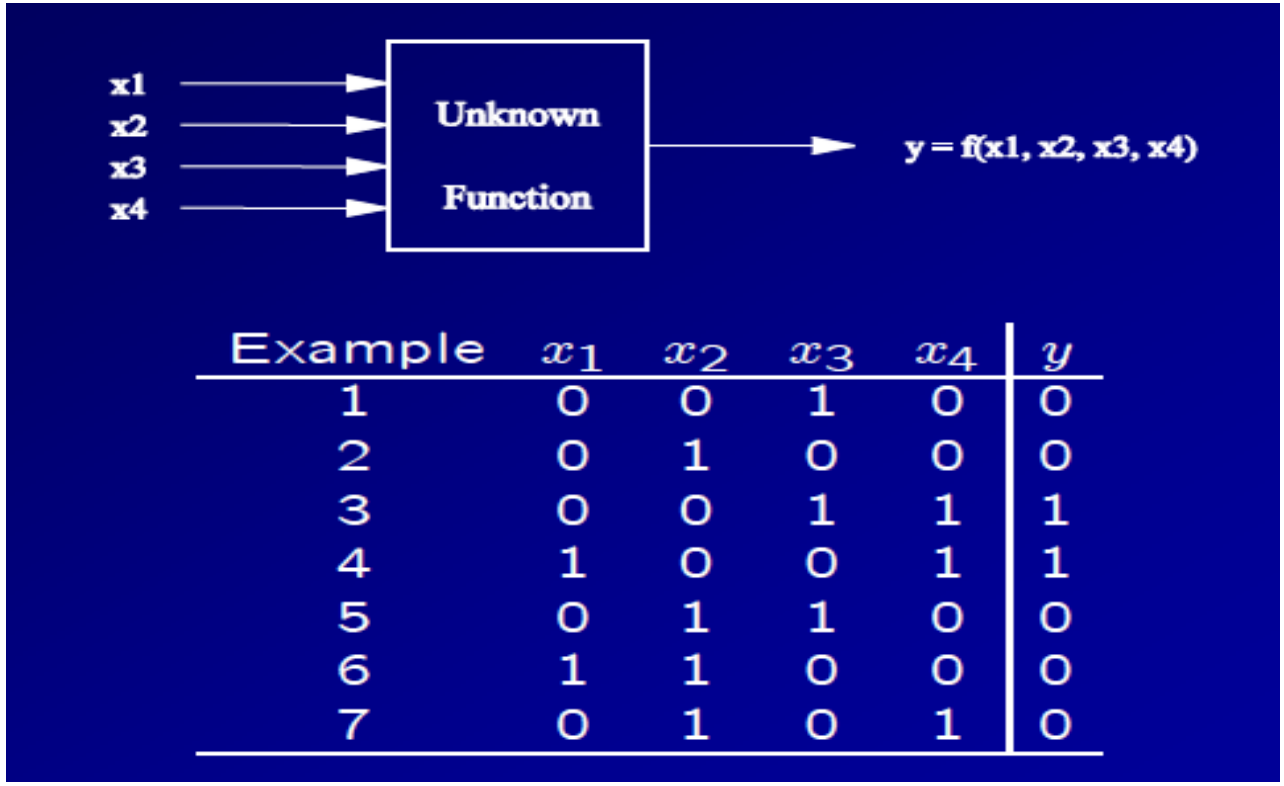
Consider this boolean function **f(x1, x2, x3)**

x1	x2	x3	f(x1, x2, x3)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Do you need machine learning to find the logical expression of this boolean function **f(x1, x2, x3)**? Yes or no? Why?

Homework 1b - send me your answer to roli@unica.it

[Thomas Dietterich, Machine learning course, CS534, 2005]



Example	x_1	x_2	x_3	x_4	y
1	0	0	1	0	0
2	0	1	0	0	0
3	0	0	1	1	1
4	1	0	0	1	1
5	0	1	1	0	0
6	1	1	0	0	0
7	0	1	0	1	0

You have the above examples of the outputs of the Boolean function f(x1, x2, x3, x4)

Do you need machine learning to find the logical expression of this boolean function? Yes or no? Why?

Homework 1c (Bonus 0.5/30)- send me your answer to roli@unica.it

Which problems you know that have many examples for how to solve them, and, therefore, you could try to use machine learning to solve them? Give 3 examples.

Which problems you know that don't need machine learning to solve them? Give 3 examples.

In ten years, which jobs you think that will be automatized (totally or partially) using machine learning?

Viewing the development of artificial intelligence as a new industrial revolution, what is the relationship between algorithms and data? Is it similar to steam engines and coal? what is the fundamental difference?

[https://d2l.ai, Chapter 1]