Part 1

Introduction to the course

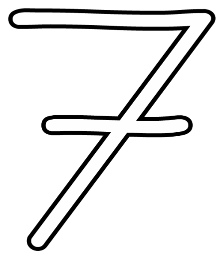
This is the number?

7

This is the number?

7

This is the number?



This is the number?

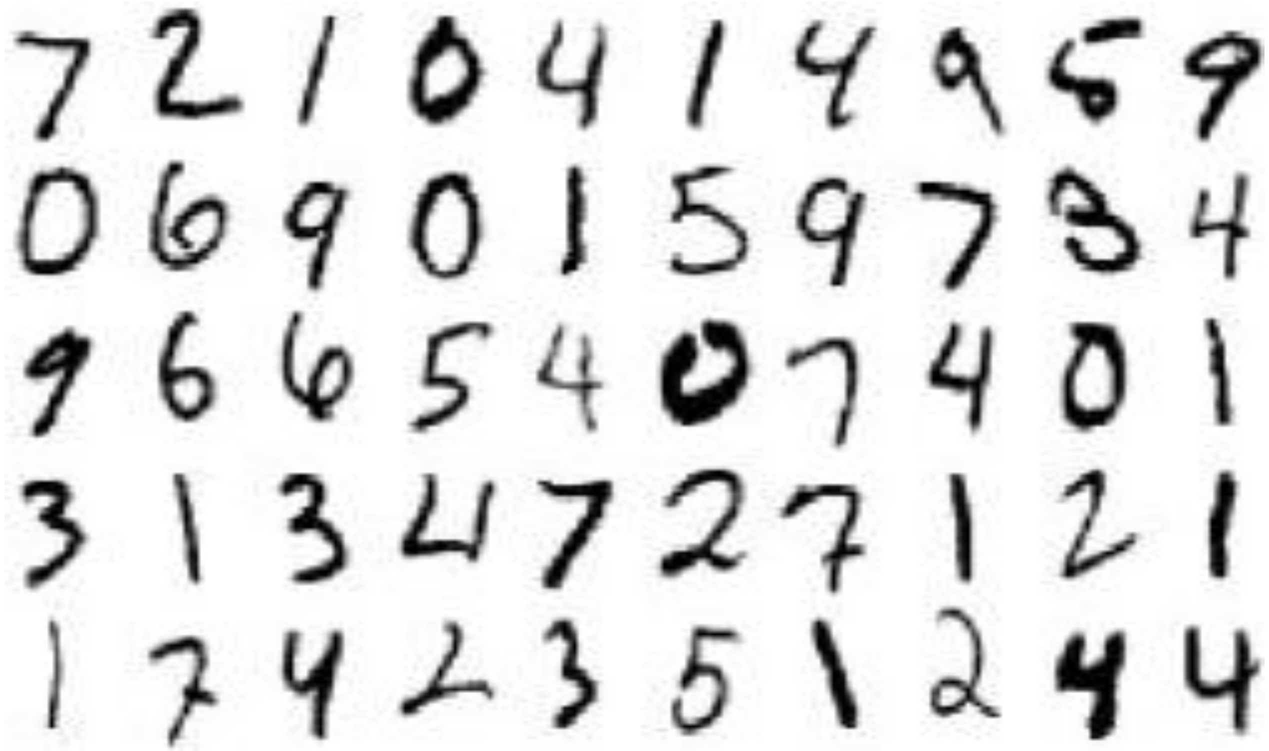
1

This is the number?

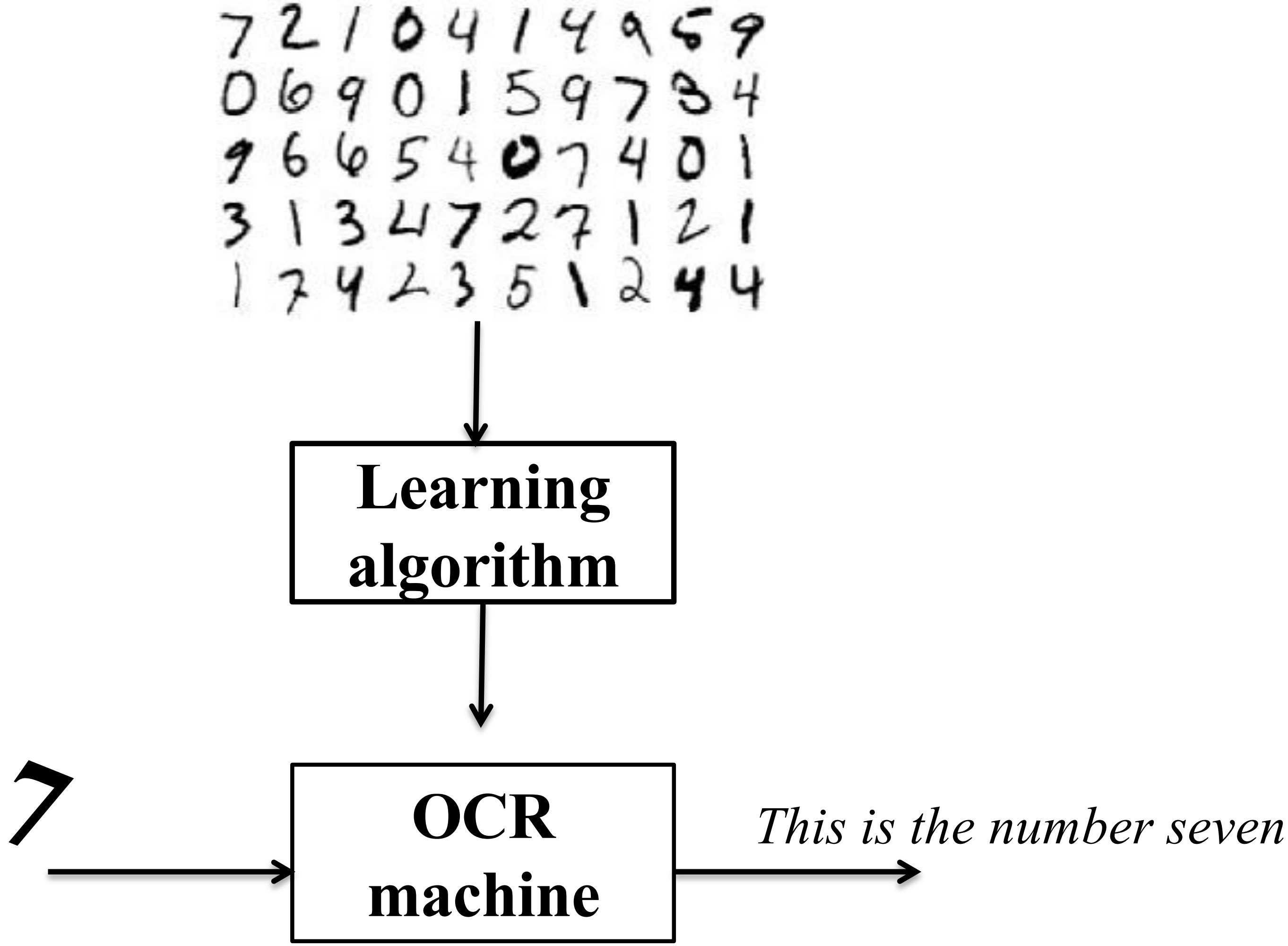
Are you able to write the exact algorithm that you use to recognize the above numbers?

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…



This is the number?

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 classification errors*

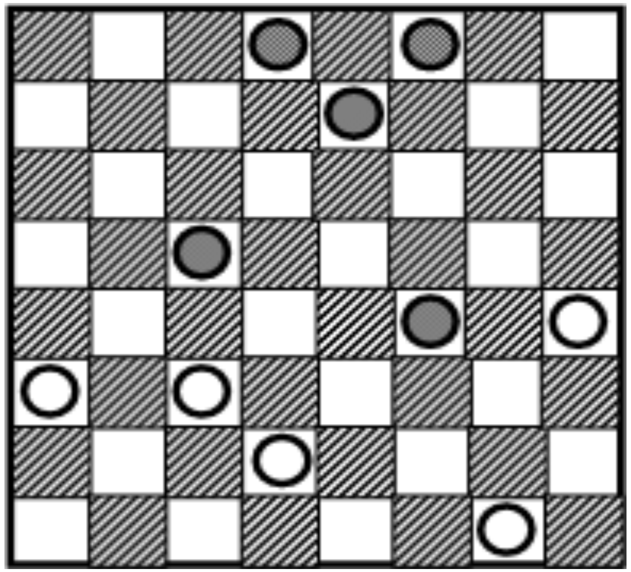
What is the goal of machine learning ?

"To build computer systems that automatically improve with experience"

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

Machine learning at the beginning…

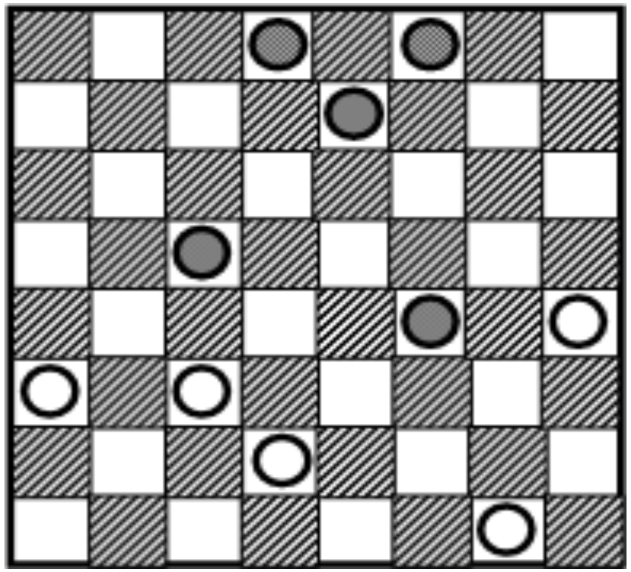
Arthur Samuel (1959) wrote a program that **learnt** to play draughts ("checkers" if you're American).

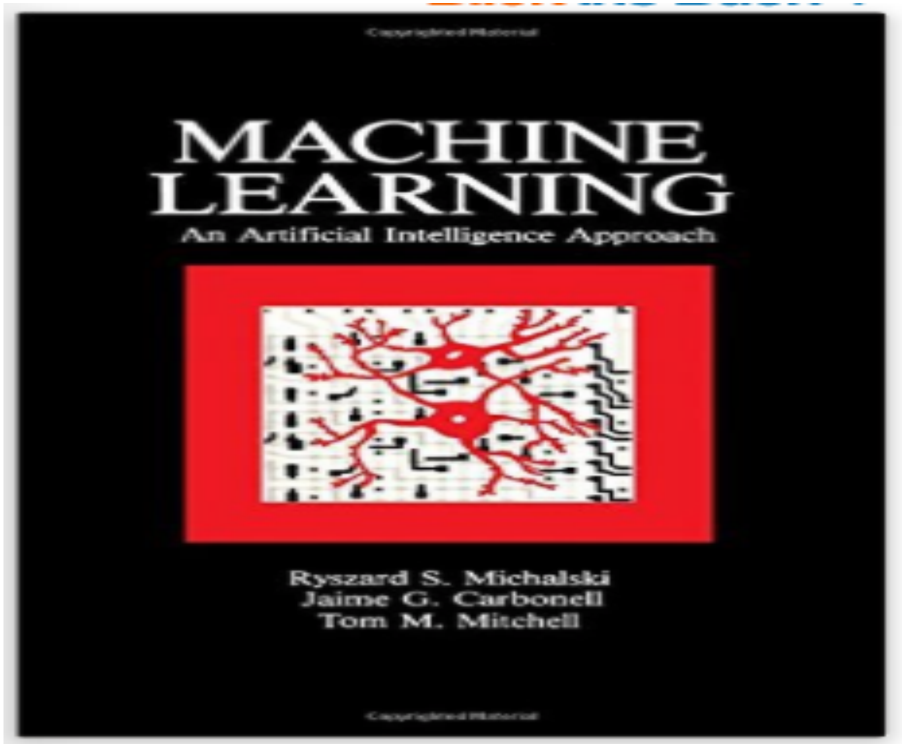


Machine learning at the beginning…

Other applications at the beginning (1980-1990):

- Learning and discovery of scientific laws
- Learning to make plans
- …



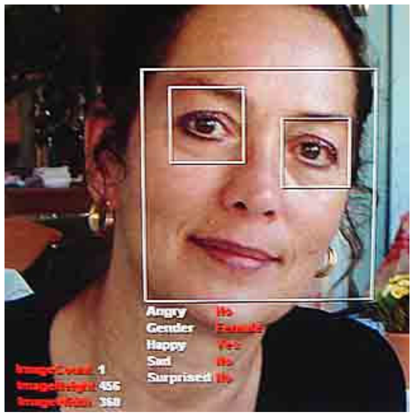


R.S. Michalski,J.G. Carbonell,T.M. Mitchell, Machine Learning: An Artificial Intelligence Approach, 1985

What is machine learning today?

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

amazon.com



Google

facebook

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

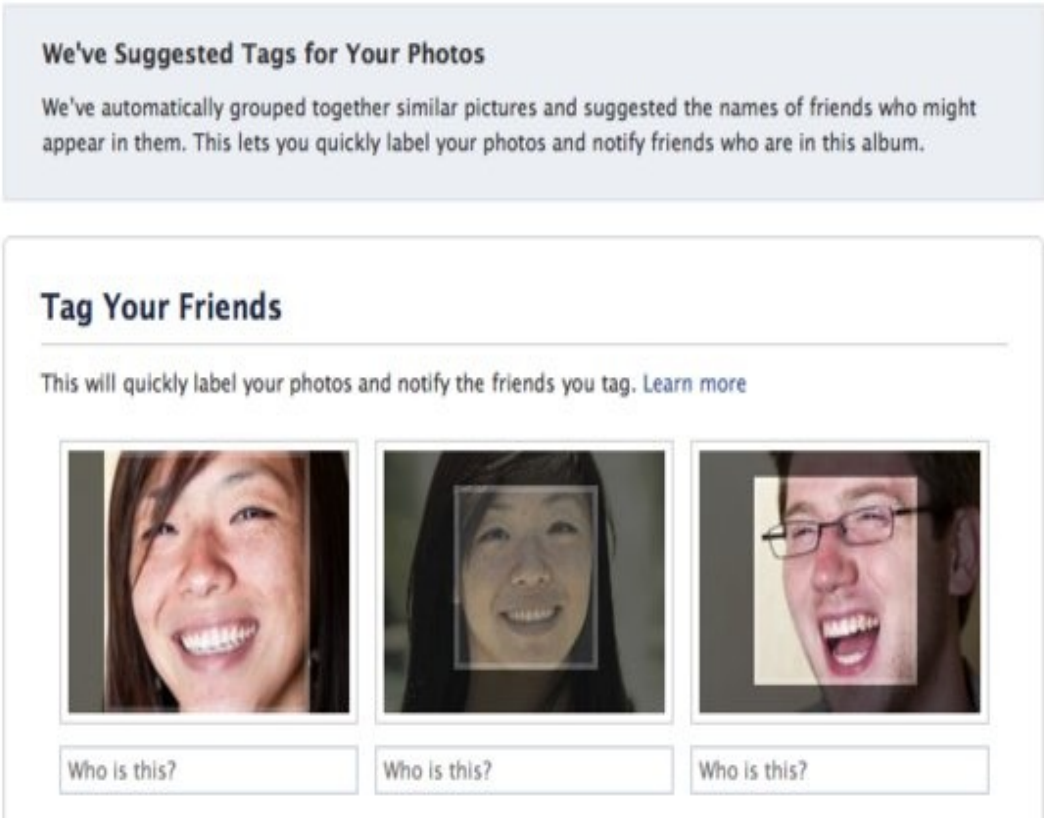
How Google uses pattern recognition

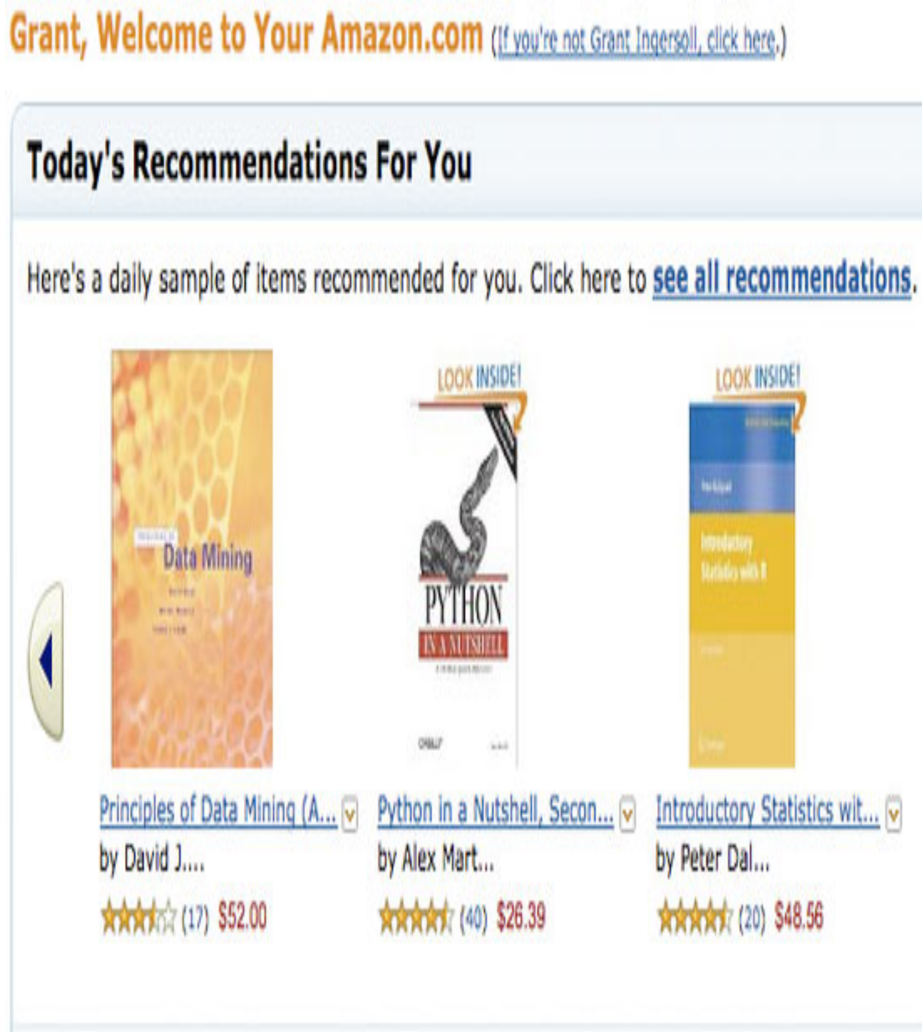
How Google uses pattern recognition to make sense of images

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

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, on the design of learning-based machines for pattern recognition

In the next slides, I explain you 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).
- Human pattern recognition may be regarded as the identification of patterns within the data collected by our sense organs (e.g., recognizing one particular object into a given image).
- 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 need to replicate 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.

Context exploitation in human beings

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

New York State lacks proper facilities for the mentally III.

The New York Jets won Superbowl III.

Human readers may ignore entirely the shape of individual letters if they can infer the meaning through context.

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?

Pattern Recognition: an engineering view

- What does it mean to design and build a machine able to recognize patterns?

- Should this machine replicate exactly the human pattern recognition system?

- Is it really possible to build a pattern recognition machine?

Questions for you:

Can machines fly?
Can machines swim?

Pattern Recognition Engineering

Questions for you:

Can machines fly?
Can machines swim?

Most people agree that airplanes can fly
But they do not call movement of boats through the water "swimming"

The same argument may hold for pattern recognition

Can machine recognize patterns?

Depends on the meaning that we give to the term "recognize"

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!

Classification, recognition, interpretation

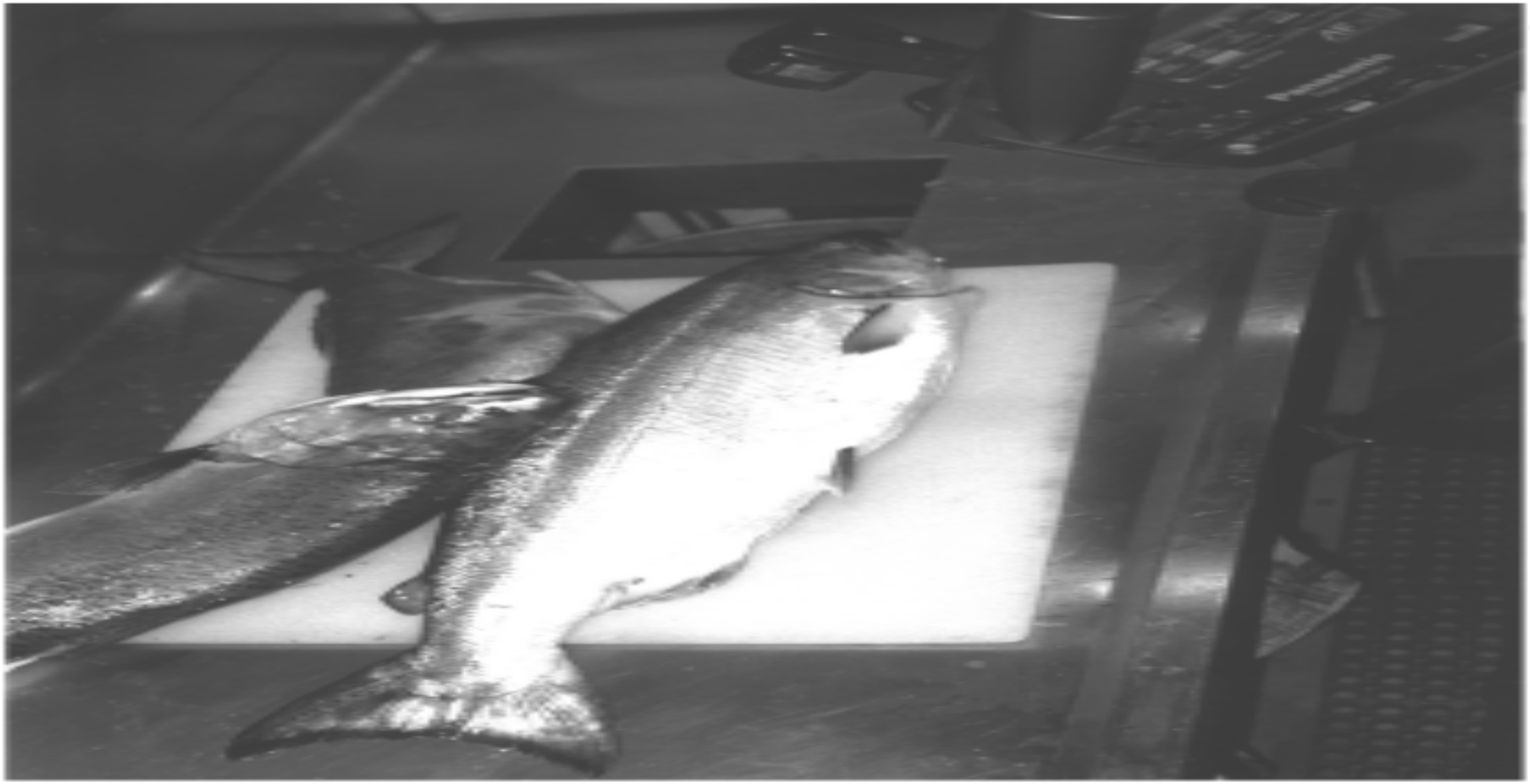
- Typically, three different phases are identified in a patter recognition process: classification, recognition, and interpretation
  - These three phases are strictly linked, and are carried out jointly, progressively, and in an unconscious way by human beings.

  - Classification: assigning a *"pattern"* to a category/class (e.g., assigning symbols in a document to alphabet letters);
  - Recognition: recognizing a complex object made up of component patterns (e.g., recognizing a "word" after having classified the letters which make the word);
  - Interpretation: understanding the meaning of the whole stream of input data. For instance, interpreting the semantic meaning of a text, after the recognition of the words, and understanding if the text deals with "economy" or "sport".

- This 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.

Let us start with a hypothetical example of pattern classification, to introduce basic concepts…

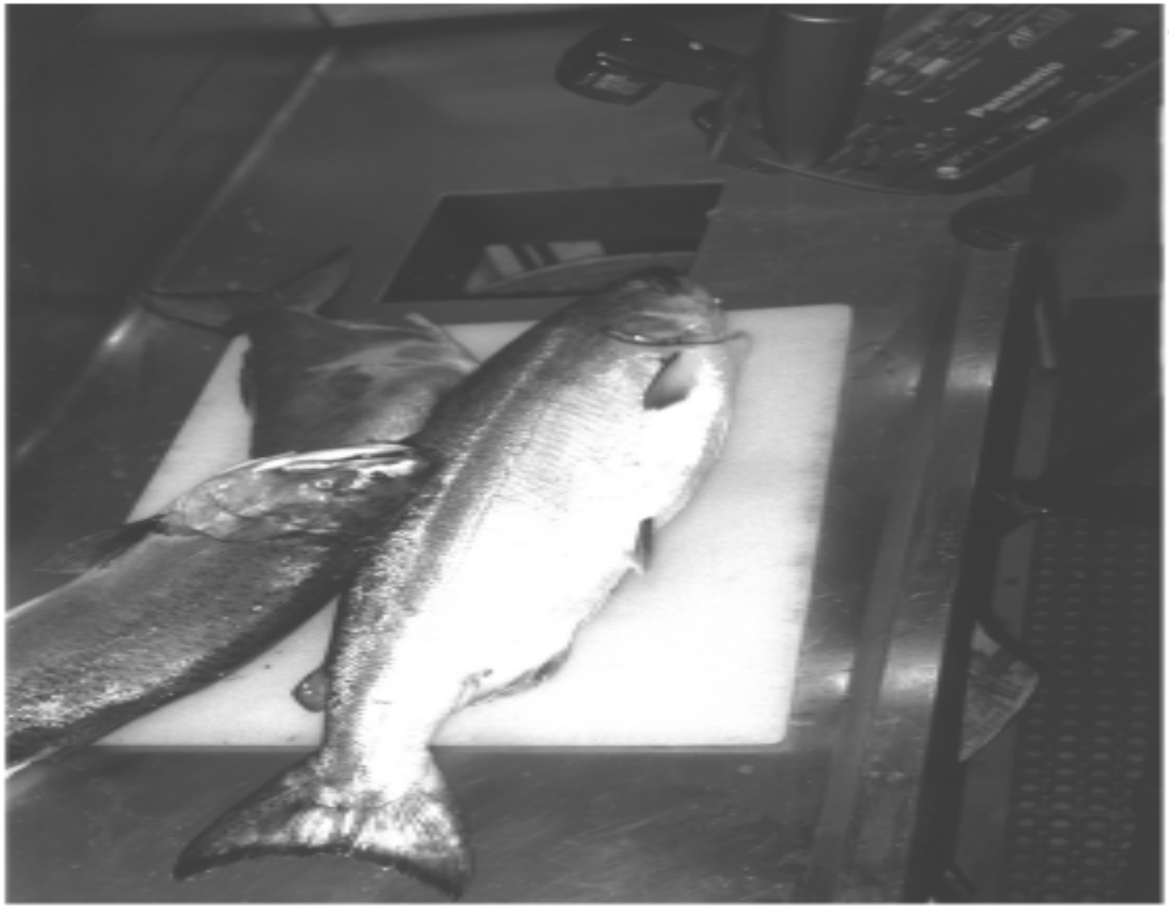
- Suppose that a fish packing plant wants to automate the process of sorting incoming fish on a conveyor belt, in order to discriminate salmons from sea basses by processing images acquired by a digital camera…[Example from DHS book; Duda, Hart, and Stork book]



Classification of salmon and sea bass….

•It is easy to see that the a large part of the image acquired is not so useful to classify the objects of interest, as a large part is “background”…

•All the image background is not interesting for our classification task.

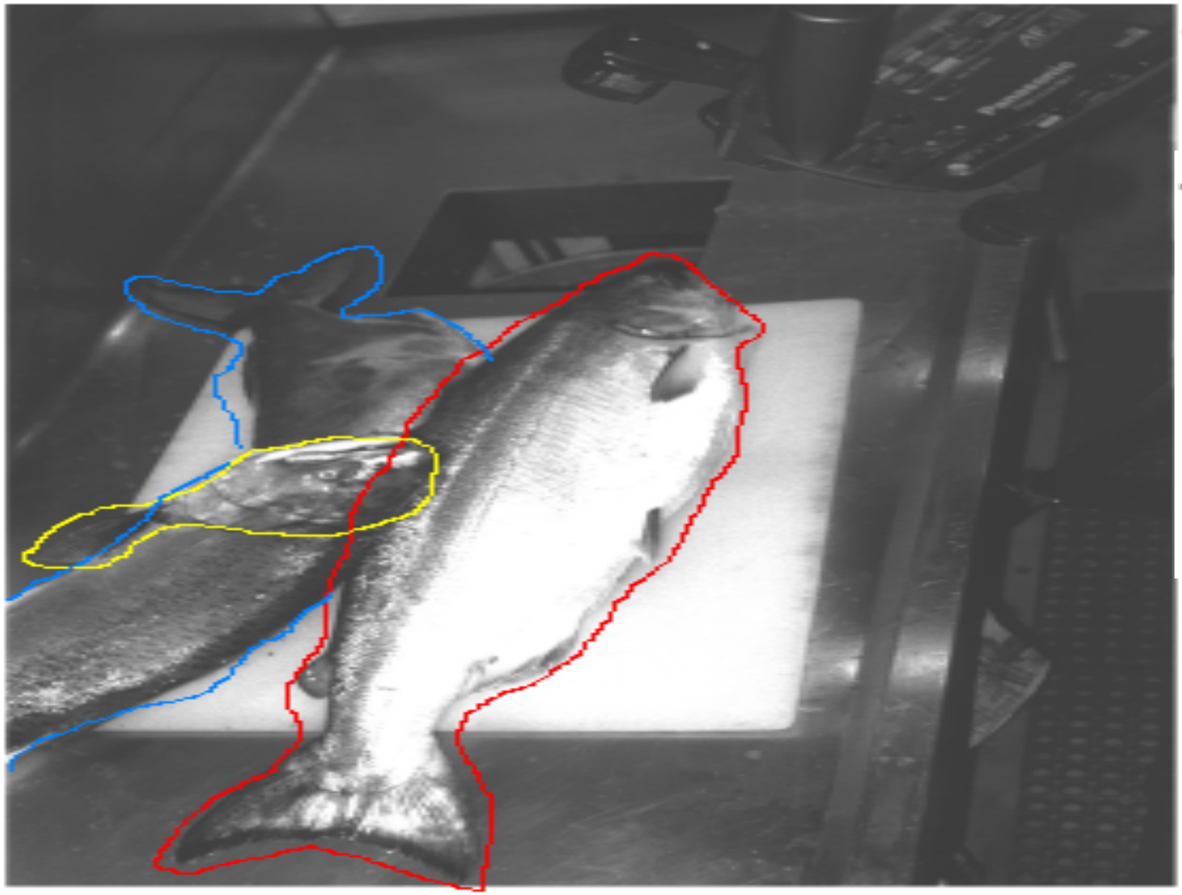


✓This is very common in pattern recognition tasks.

✓We need to identify “regions of interest” (called **ROI**s) which contain the patterns we want to classify, and then to characterize such patterns (regions) with **“features”** which are the most informative measures for the classification task at hand.

Classification of salmon and sea bass….

- We collect images.
- We pre-process images to enhance quality (e.g. to enhance image contrast) and to identify "patterns" of interest. For instance, we "**segment**" images to separate background image from regions containing fish.



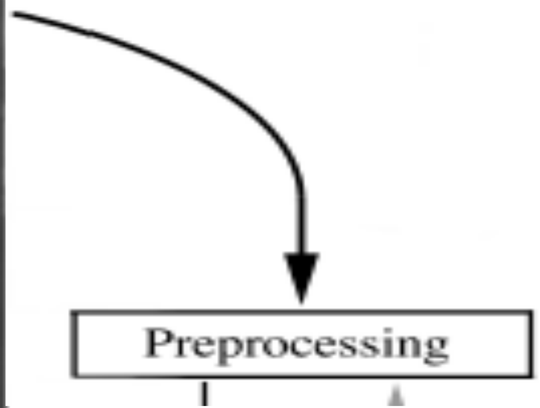
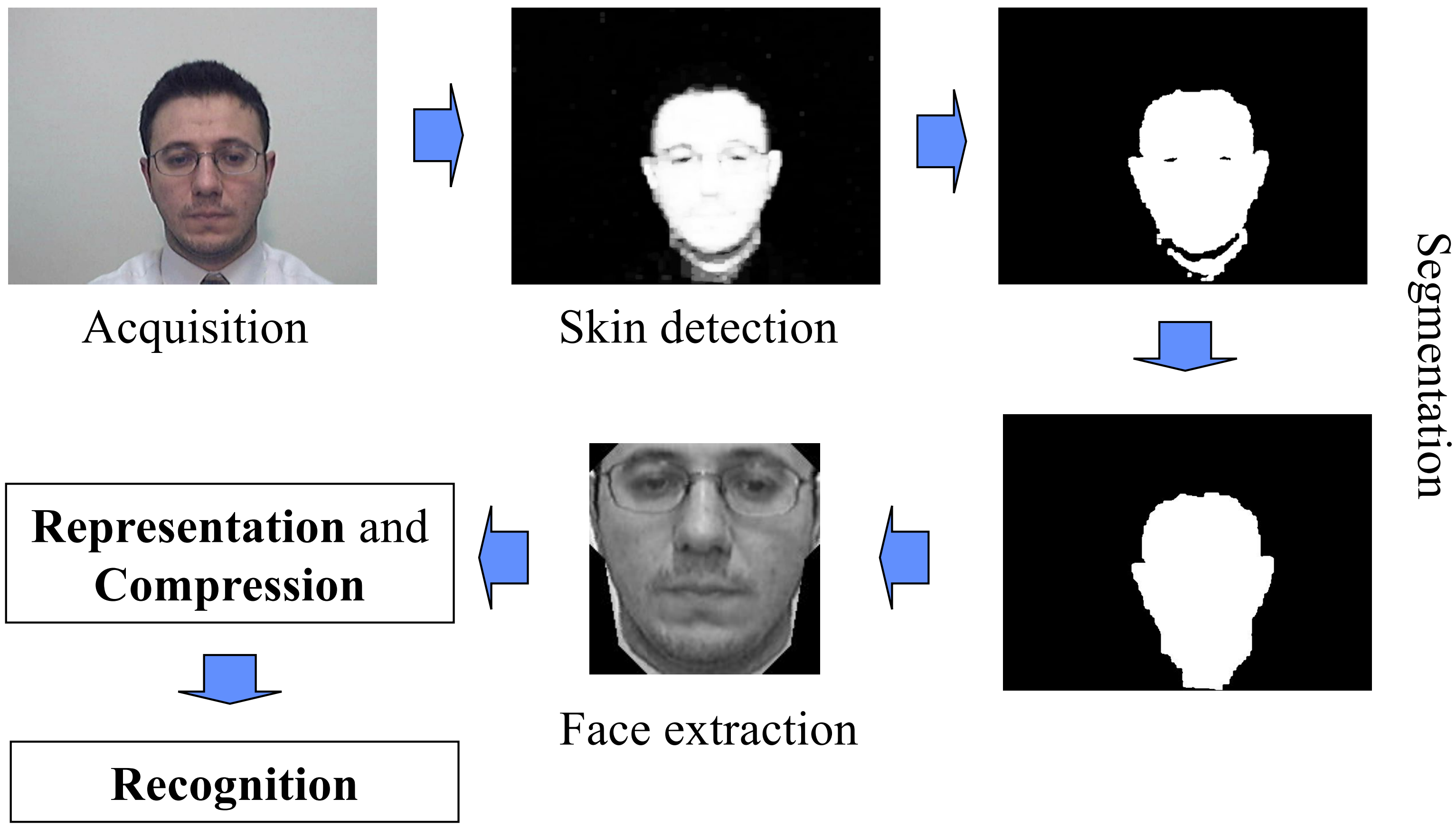


Image segmentation

Face recognition and image segmentation

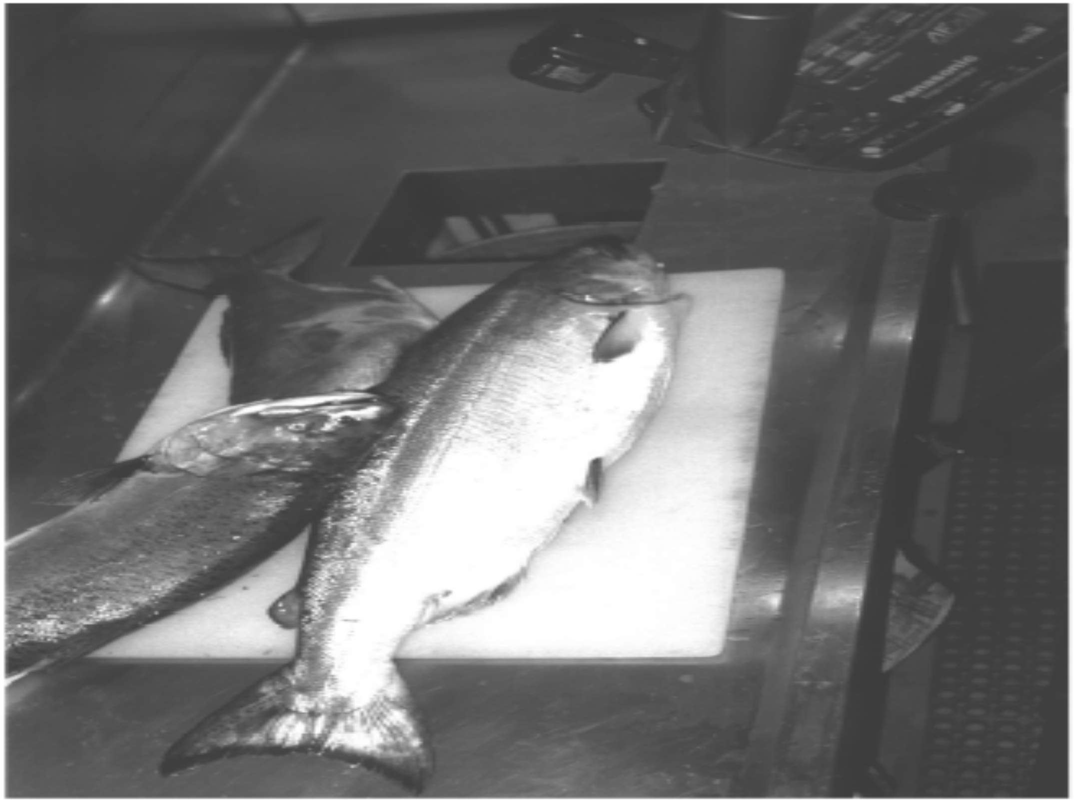
Image segmentation is often used in face recognition, in order to identify the image regions which contain the most relevant information. In this application, segmentation is often based on the detection of "skin" regions.

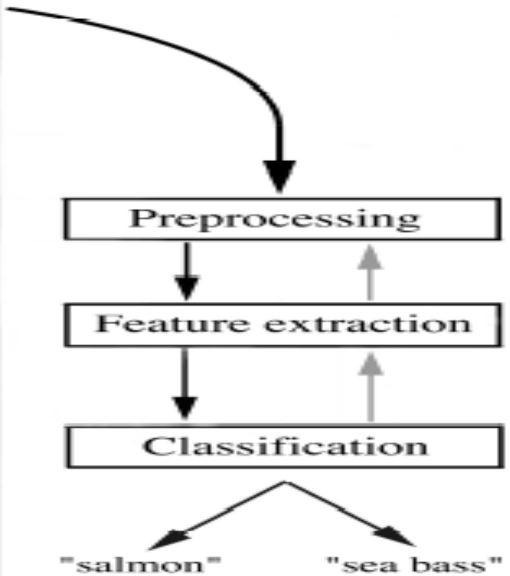


Classification of salmon and sea bass….

•Regions of interest (containing “patterns”) are usually characterized with a set of physical measures, called ***features*** (e.g., length, colour, shape of the fish). Compact representations of patterns facilitate their correct classification → **Feature Extraction**

•**Pre-processing** should be done so that our physical measures (features) are reasonably *invariant* with respect to illumination changes, position of the fish on the conveyor, etc.

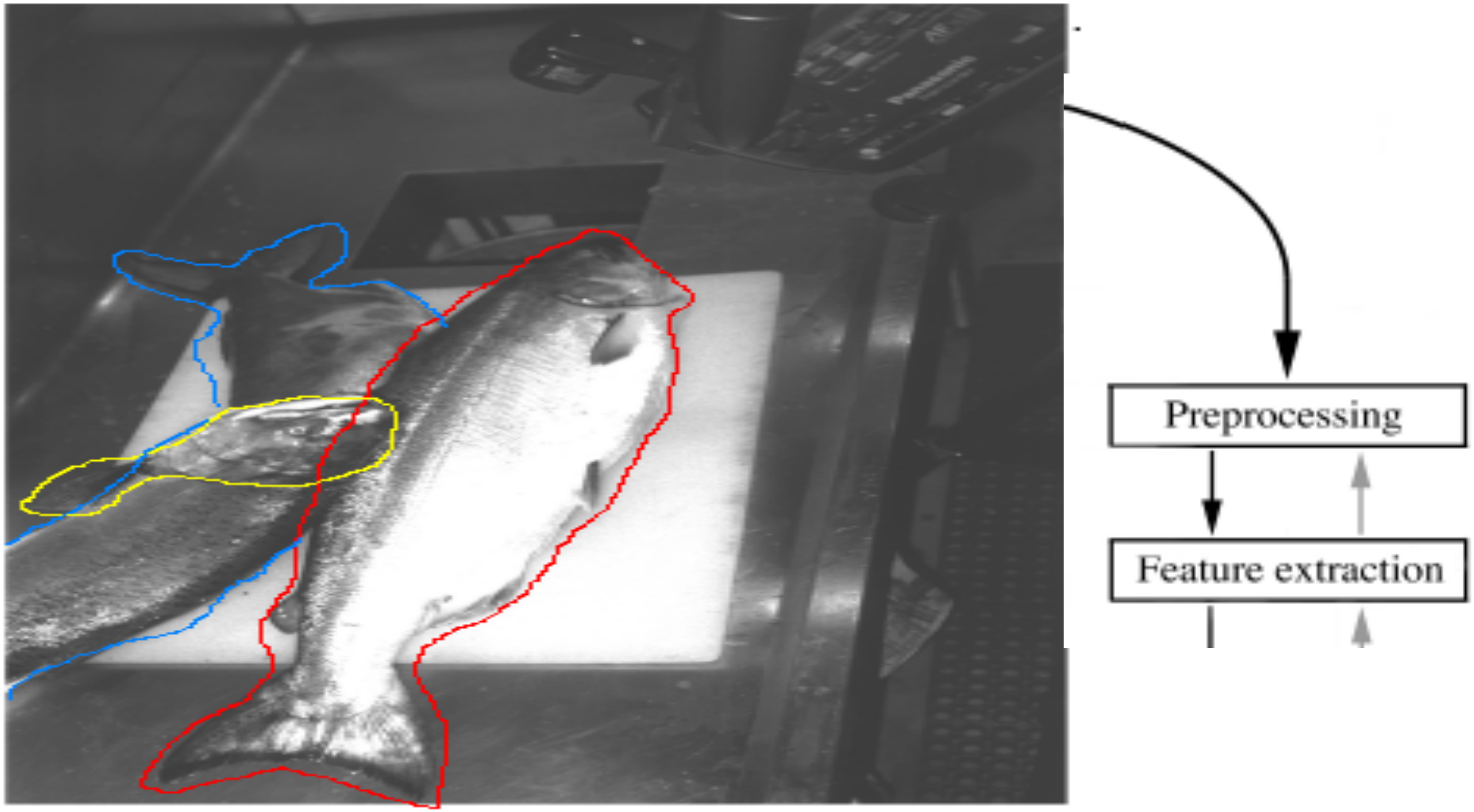




Classification of salmon and sea bass….

•The extraction of physical measures to distinguish salmon from sea bass: **Feature Extraction**.

•For instance, lenght of the fish is an useful feature, if we know that a sea bass is generally longer than a salmon.



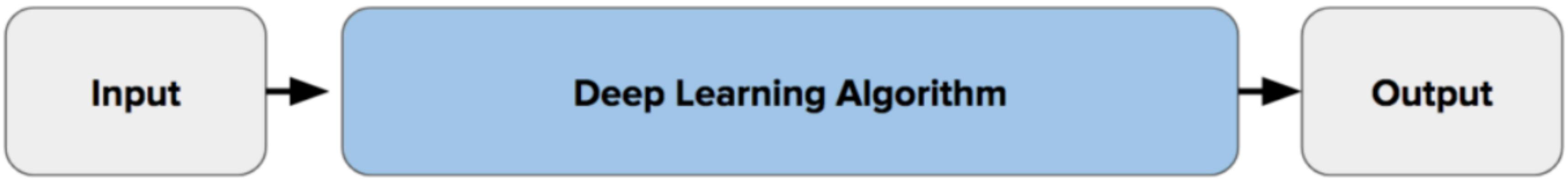
•Patterns to be classified are the three regions highlited in the image

•Each pattern is characterized by a set of features (e.g., length, lightness, area, etc)

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.

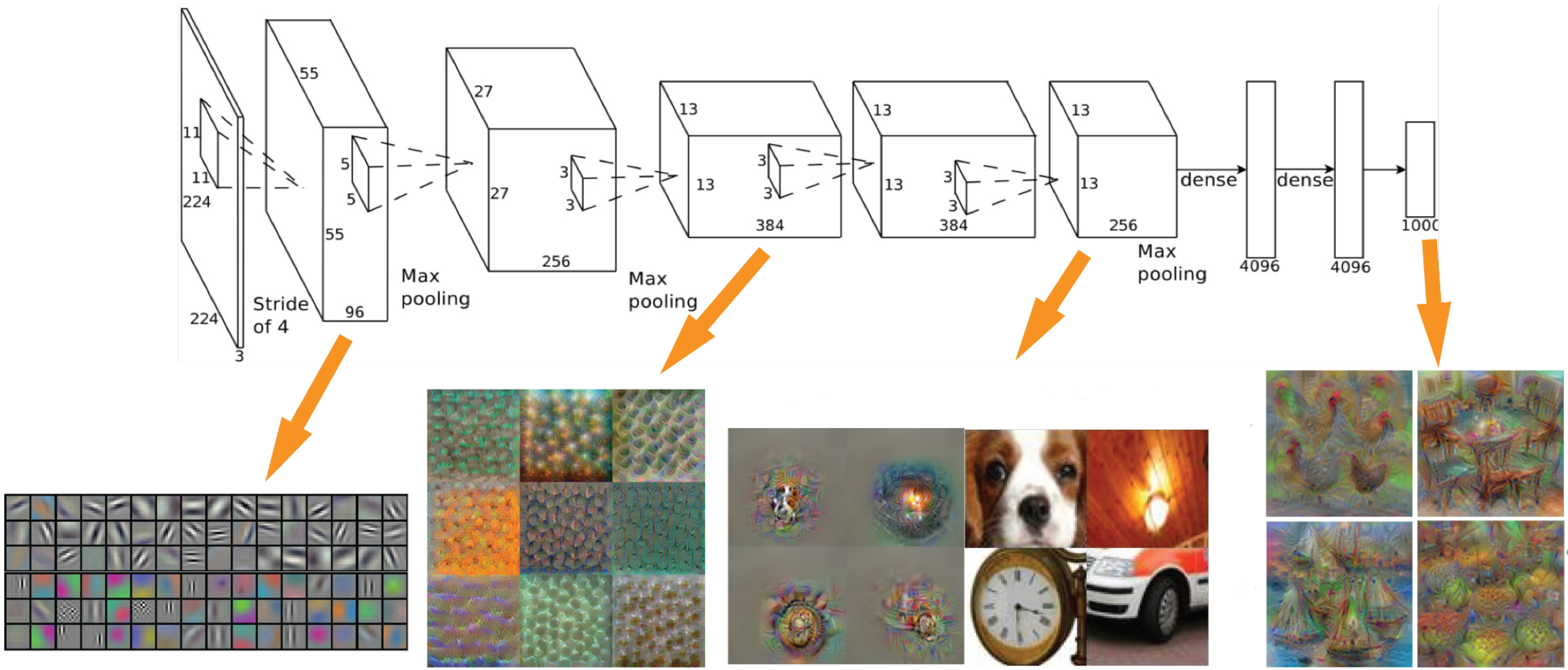
Processing flow for extraction of **handcrafted»** features



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).



Conv 1: **Conv 3:** **Conv 5:** **Fc8:**

Basic concepts: class and feature [L. Kuncheva, 2004]

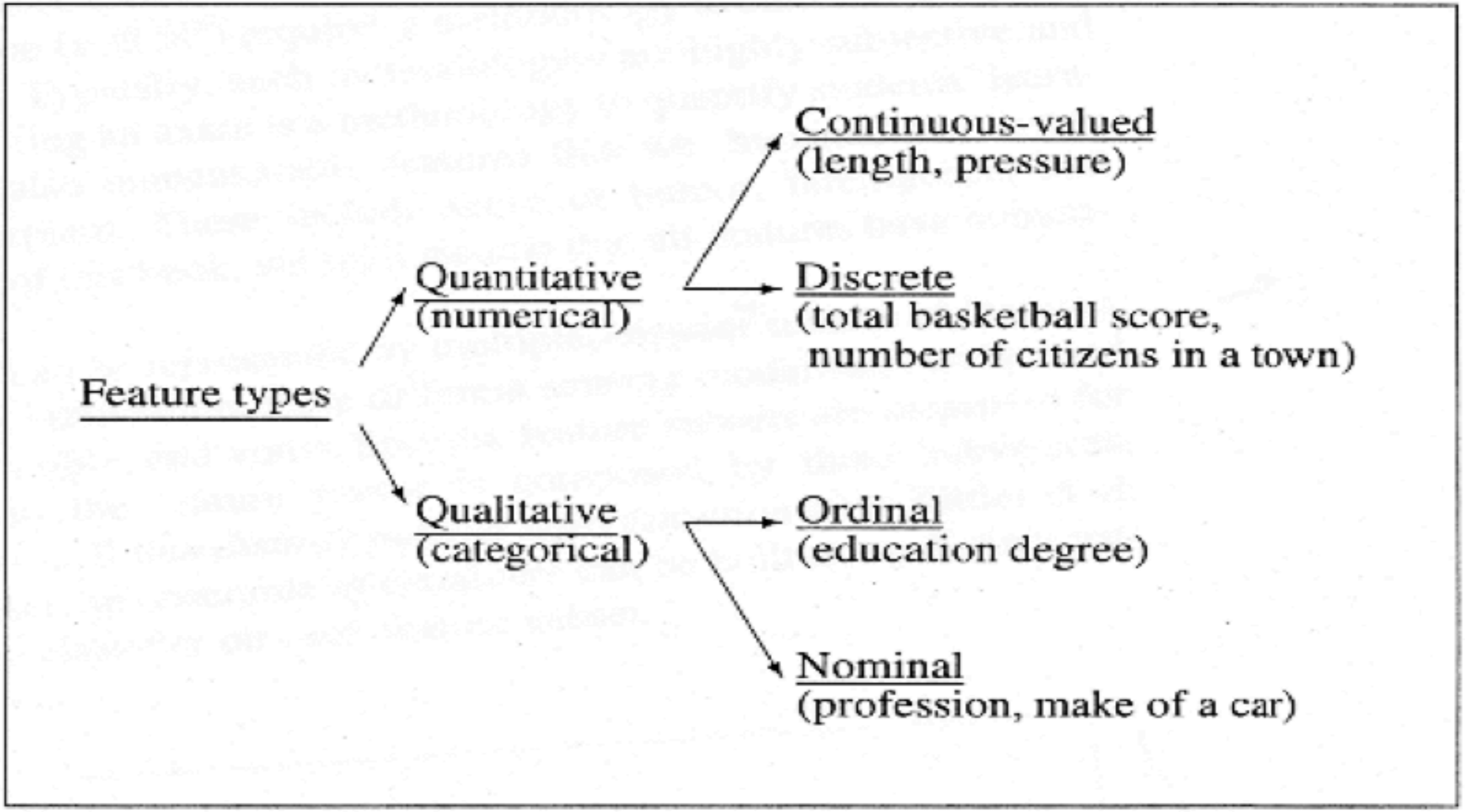
Pattern classification is about assigning class *labels* to patterns ("objects"). Patterns are described by a set of measurements called also **features** (or attributes).

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

Class: intuitively, a class contains similar objects, whereas objects from different classes are dissimilar (salmon and sea bass classes).

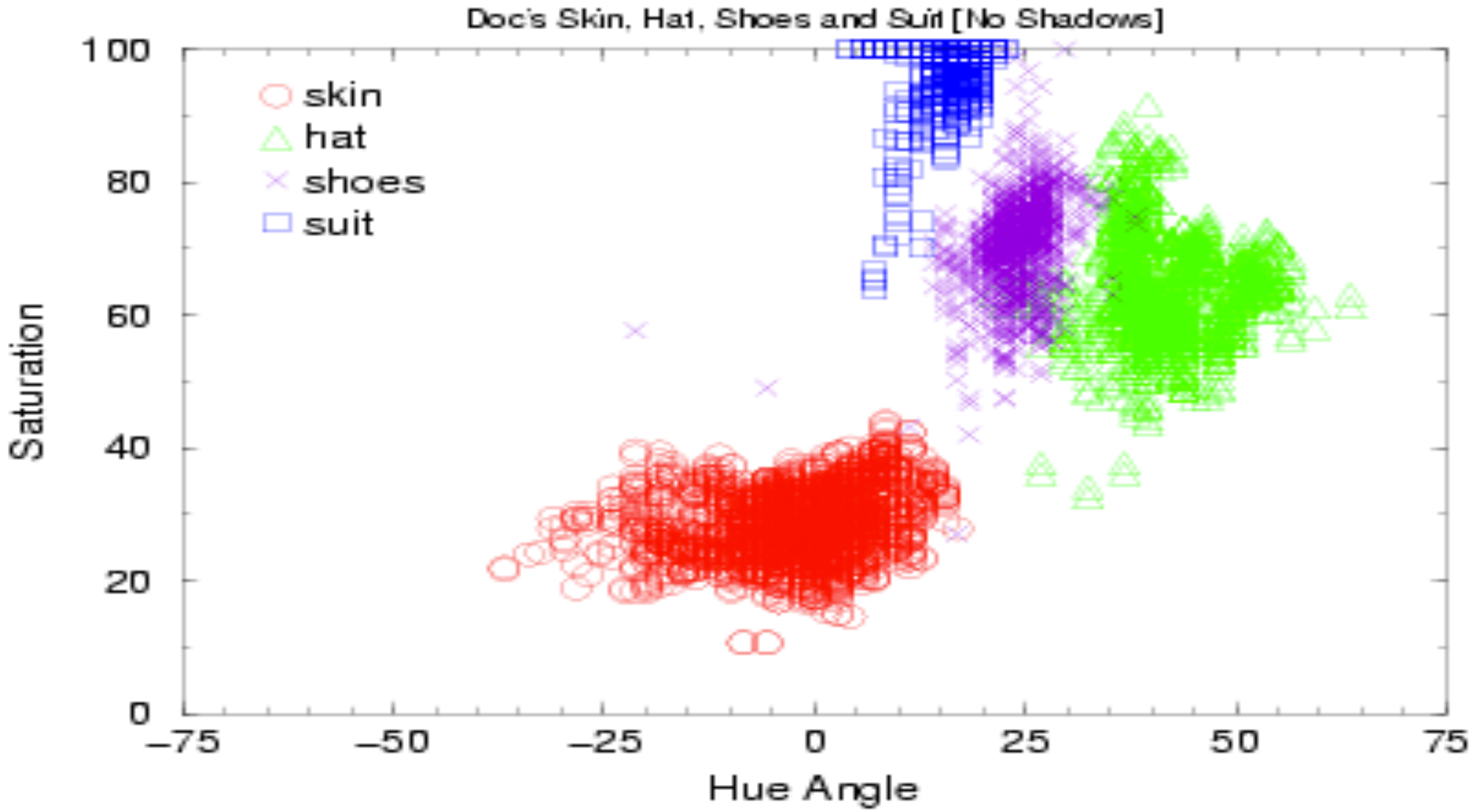
In this course, we shall assume that there are *c* possible classes in the problem, and will denote that as: $\boldsymbol{\Omega = \{\omega_1, \omega_2, \ldots, \omega_c\}}$, each pattern belongs to one of the "c" classes of the set Ω. We will say that each pattern has a class **label**.

Basic concept: feature types [L. Kuncheva, Combining pattern classifiers, Wiley, 2004]



- Statistical pattern classification operates with numerical features.

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.

Classification model

- **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 a very simple classification **model** based on a simple heuristic rule:
  - *A sea bass is generally longer than a salmon*
  - We can rewrite more formally this heuristic rule as follows:

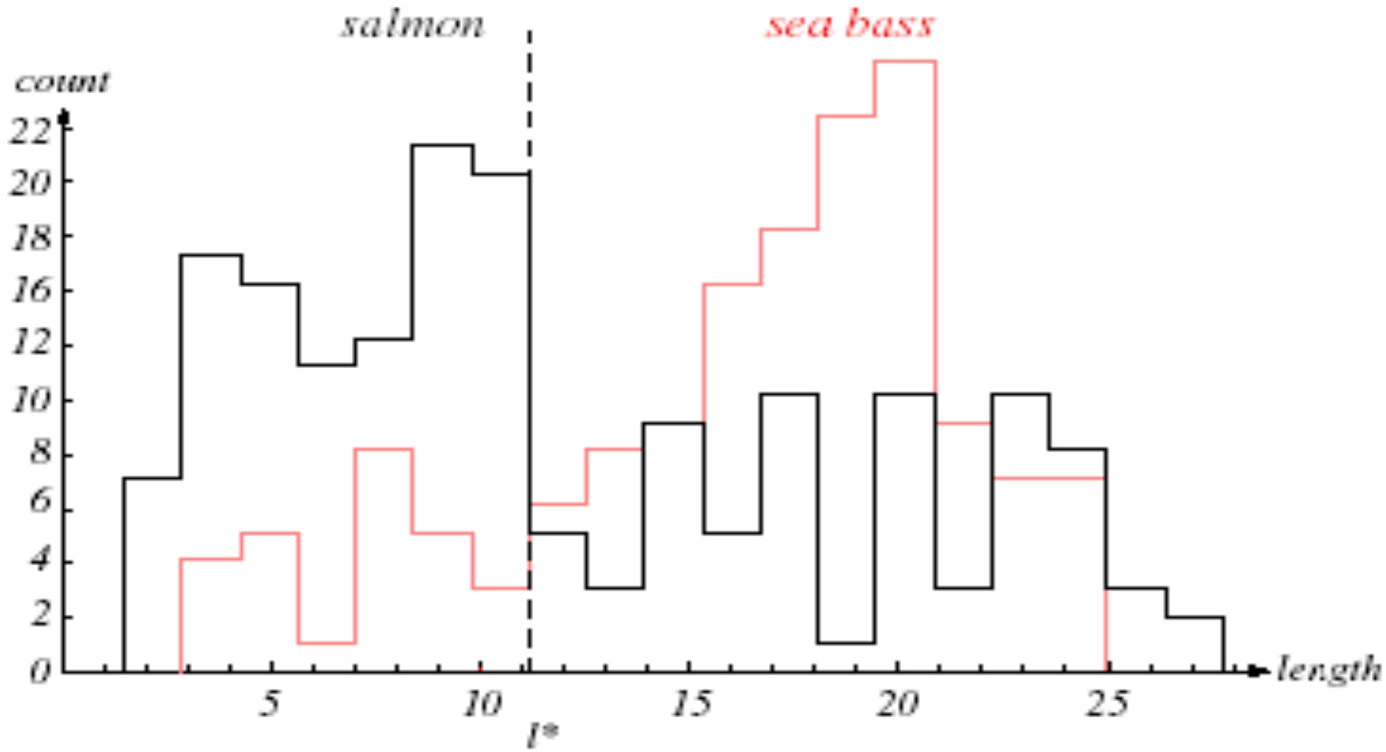
    if $l > l^*$ *then* fish=sea bass , *else* fish=salmon

- The threshold value l^* can be an heuristic value that managers of the fish plant know, otherwise we should estimate it.
- *How can we estimate l^* ?* We need a set of samples/examples of the two fish categories (called **"design o training set"**)

Classification model

Computation of parameters of the classification model by a “training set”

The threshold value l^* can be estimated by the empirical distributions obtained by a “training set” (set of examples of the two fish categories)



In this simple example, we have only one parameter whose value has to be estimated for our classifier ($l=l^*$).

- In general, we will have a set of parameters.

Basic concept: design or training data set

The information to design a classifier is usually in the form of a labeled data set $\boldsymbol{D}$ (called design or training set):

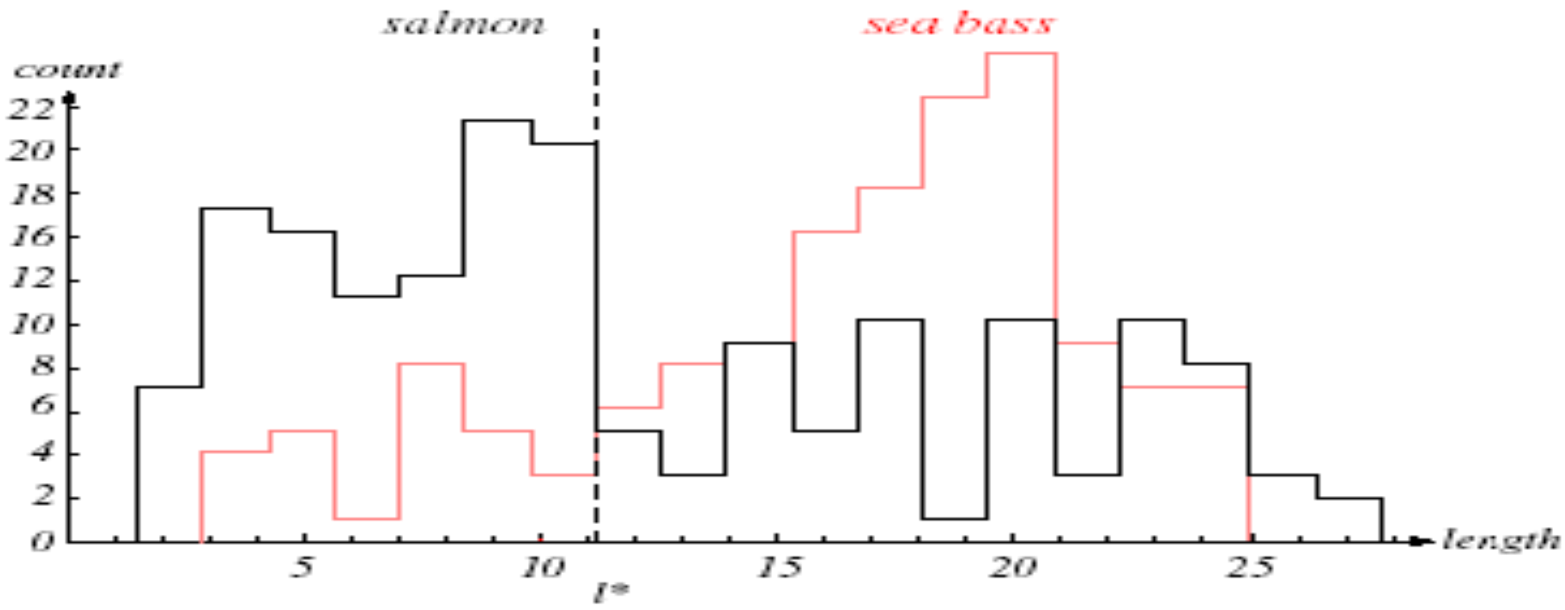
$$\boldsymbol{D} = [\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots, \boldsymbol{x_n}]$$

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

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

In the previous example, $\boldsymbol{D}$ 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:

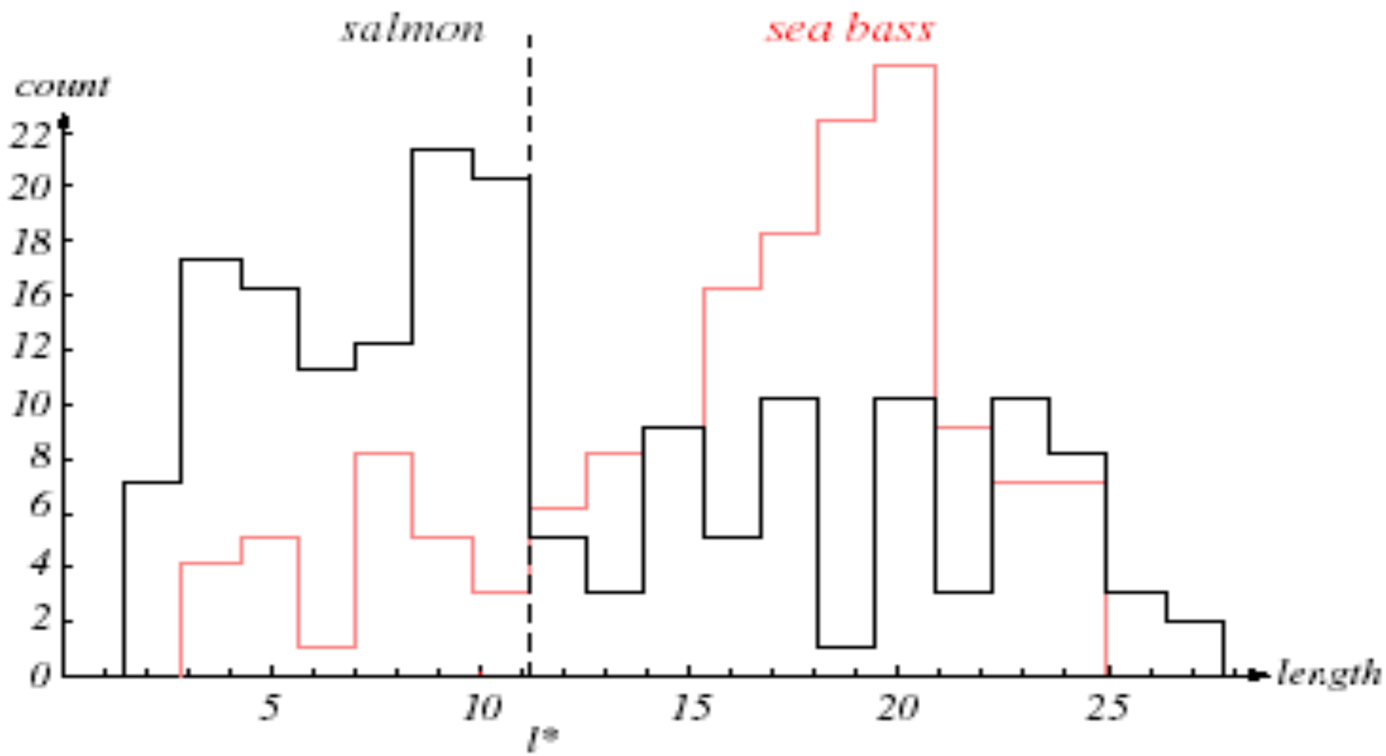
P(lenght / salmon) and *P(lenght / sea bass)*

and then make a probabilistic decision…

➢ We will discuss this in great detail in Part 2 !

Classification of salmon and sea bass….

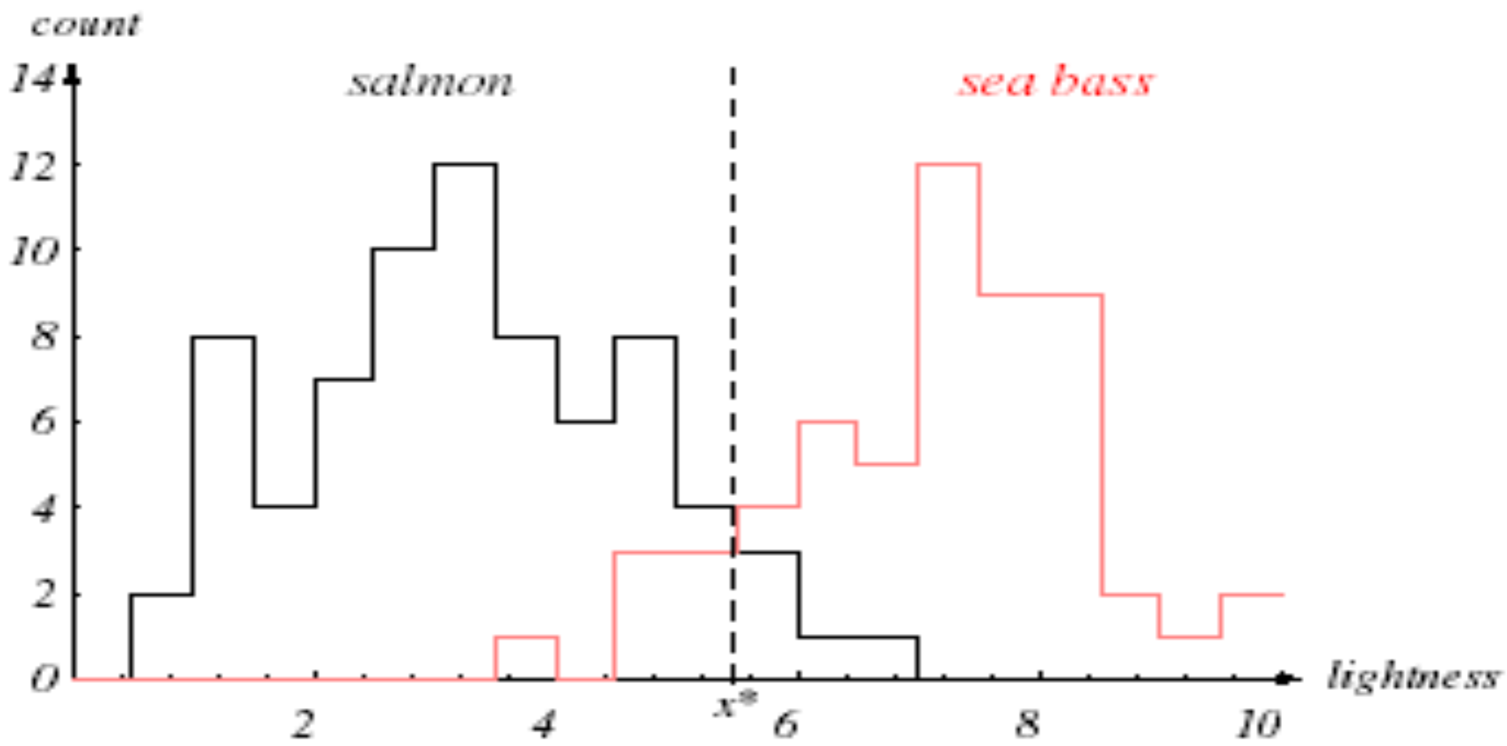
Computation of parameters of the classification model by a "training set"



- If the feature *l* ("lenght" of fish) does not allow a good discrimination between the two classes, we can check if a different feature can do better.

Classification of salmon and sea bass….

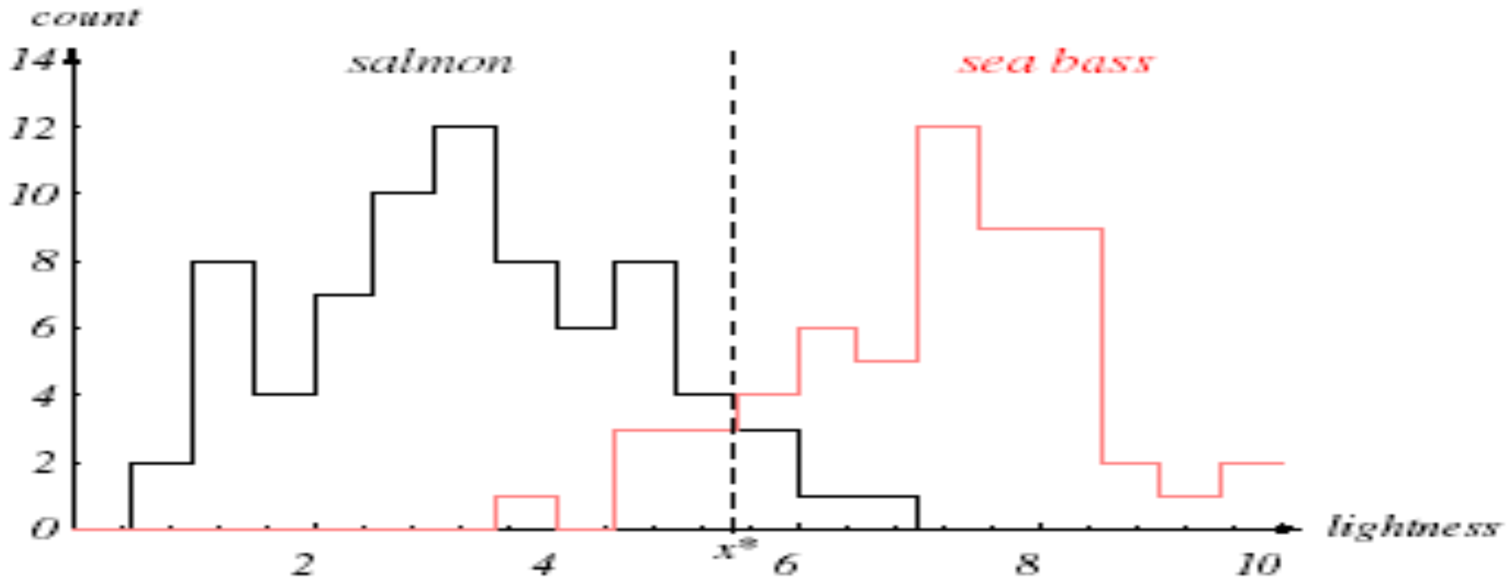
Computation of parameters of the classification model by a "training set"



The "lightness" feature is more effective, it distinguishes much better the two kinds of fish.

Classification of salmon and sea bass….

Classification "cost"



•In some applications, different errors have different "costs"

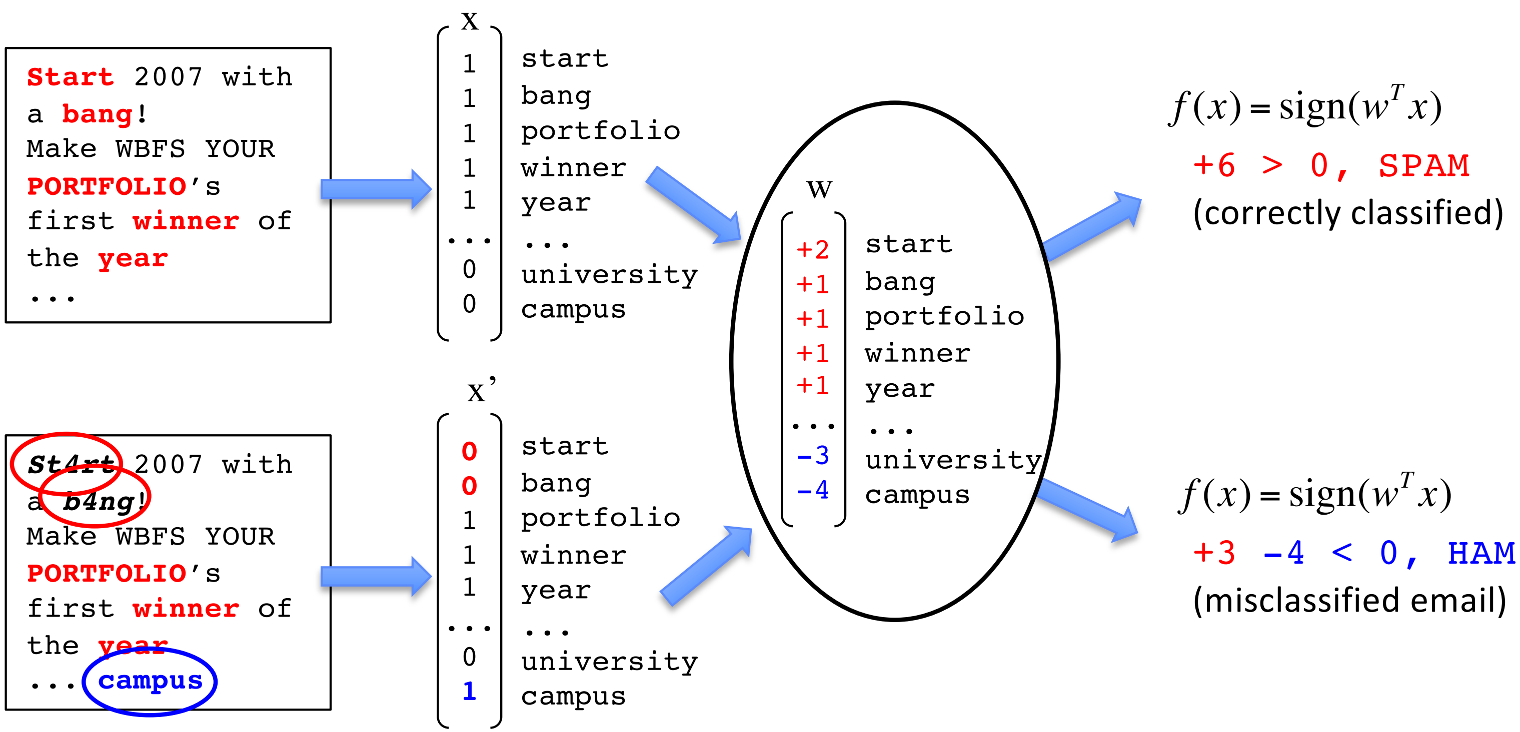
•For istance, misclassifying "salmon" can have a higher cost. Customers might dislike to find sea bass into a box where salmon should be contained.

•Therefore, the threshold value x* should be adjusted in order to take into account "costs"

➢We will see how to handle classification "costs" in Part 2.

Two quick examples of applications of pattern classifiers…

Spam filtering

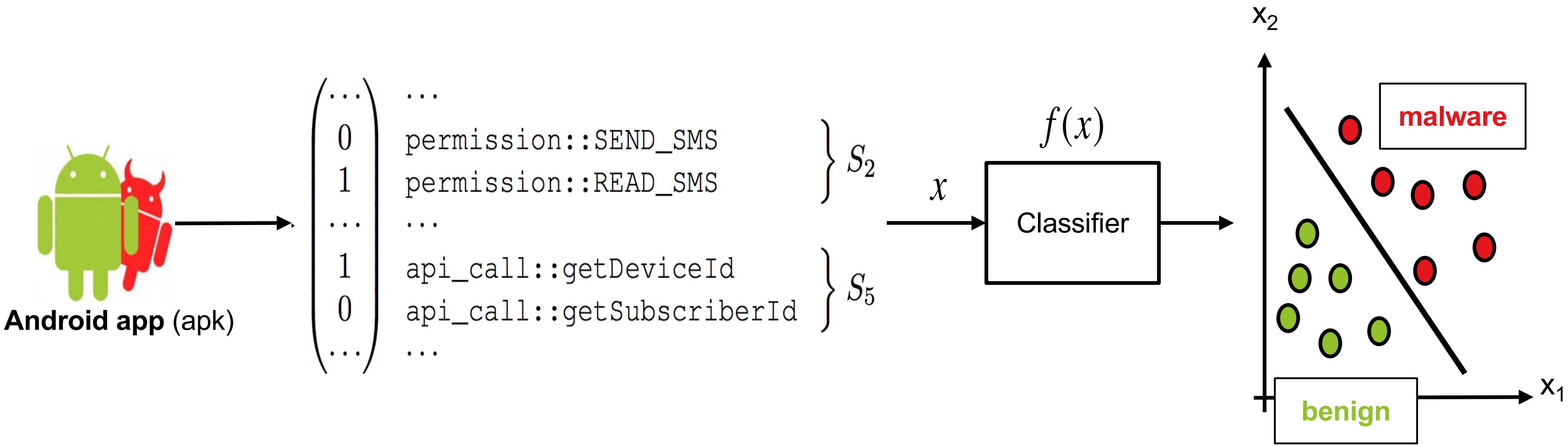


Android Malware Detection

Drebin: Arp et al., NDSS 2014

- Android malware detection directly on the mobile phone
- Linear SVM trained on features extracted from static code analysis

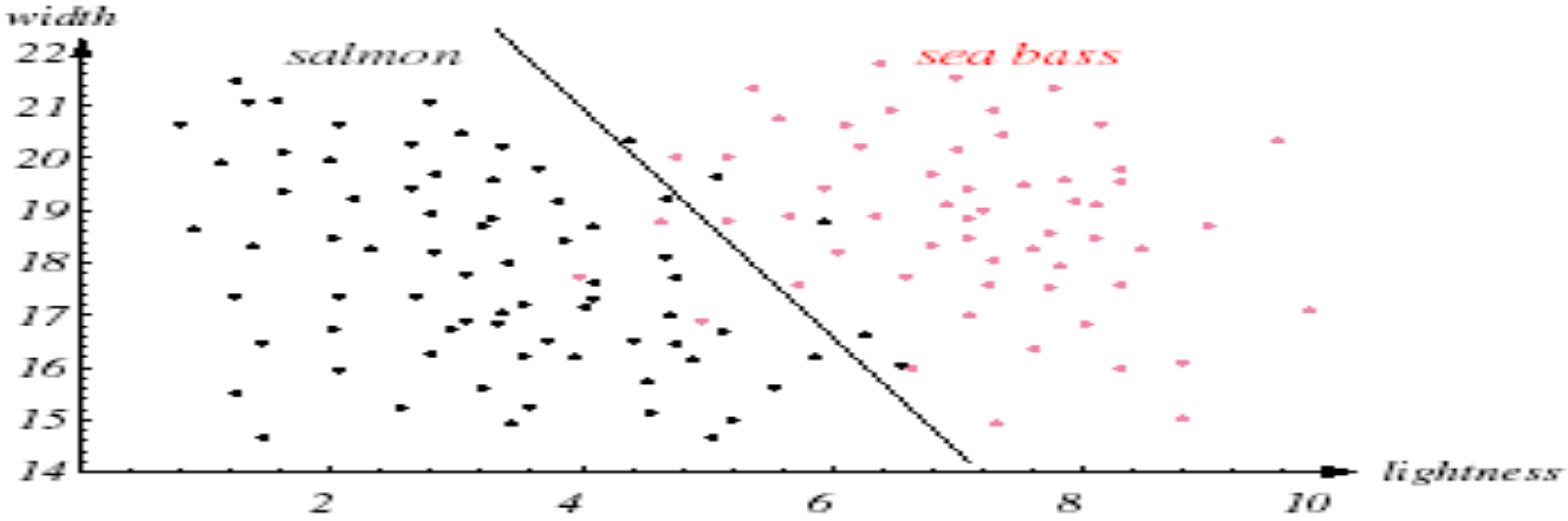
Feature sets		
`manifest`	S_1	Hardware components
	S_2	Requested permissions
	S_3	Application components
	S_4	Filtered intents
`dexcode`	S_5	Restricted API calls
	S_6	Used permission
	S_7	Suspicious API calls
	S_8	Network addresses



Demontis, Biggio et al., IEEE TDSC 2017

Basic concepts

Feature Space and Decision Boundary



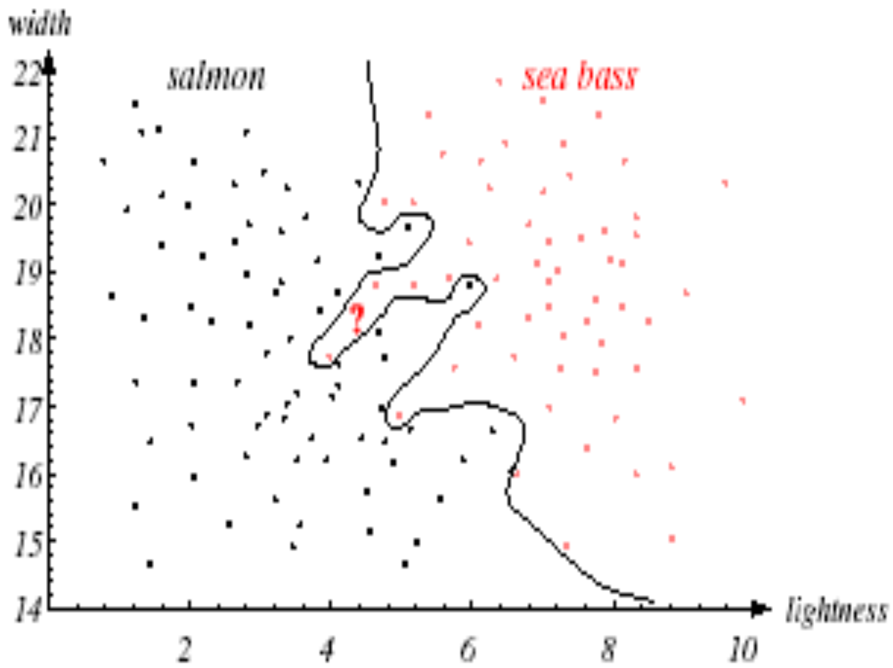
•How to reduce error? We could use two features instead of only one.

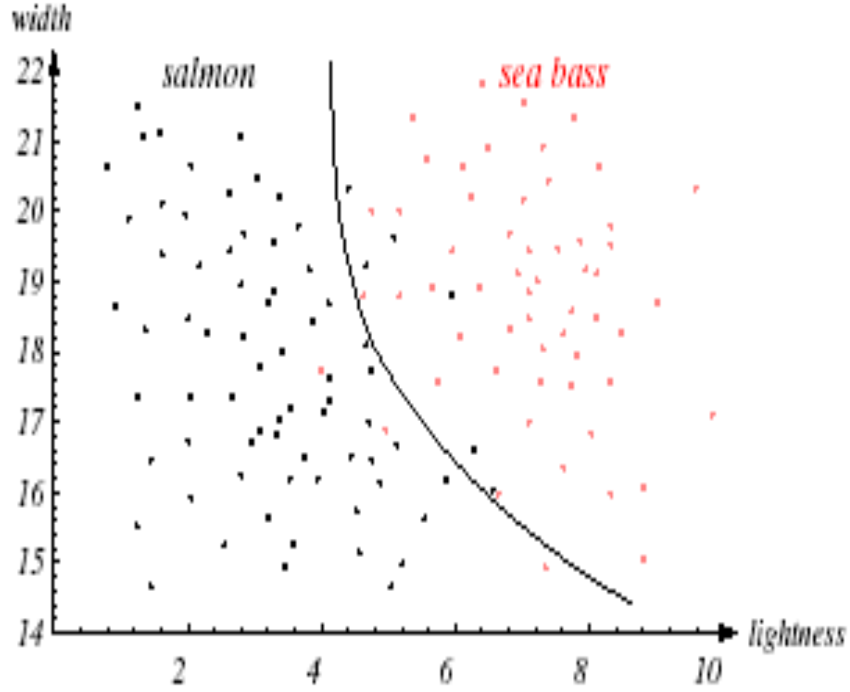
•In a d-dimensional feature space, we will have a **decision boundary** instead of a simple threshold value.

➢ But the extraction of multiple features can be expensive and it can cause a well-known problem called "**curse of dimensionality**": the number of parameters that we have to estimate to design the classifier is too large w.r.t. the data available.

Basic concepts

How much should the model be "complex"?





- What is the best decision boundary (classifier)? Left or right ?
- Key issue: "training set" is often limited (few data)
- The choice of the classification model should take into account the so-called **generalization error**, that is, error on data which is not part of the training set

Basic concepts

Let's us consider this well known boolean function

S	X1	X2	F(S, X1, x2)
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

Are you able to write the boolean expression of the function F(S, X1, X2) ?

Basic concepts

S	X1	X2	F
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

This a digital **multiplexer** !

It is a device that selects one of several digital input signals and forwards the selected input to a single line.
It is also called **data selector** or **controlled switch**.

S \ X1 X2	00	01	11	10
0			1	1
1		1	1	

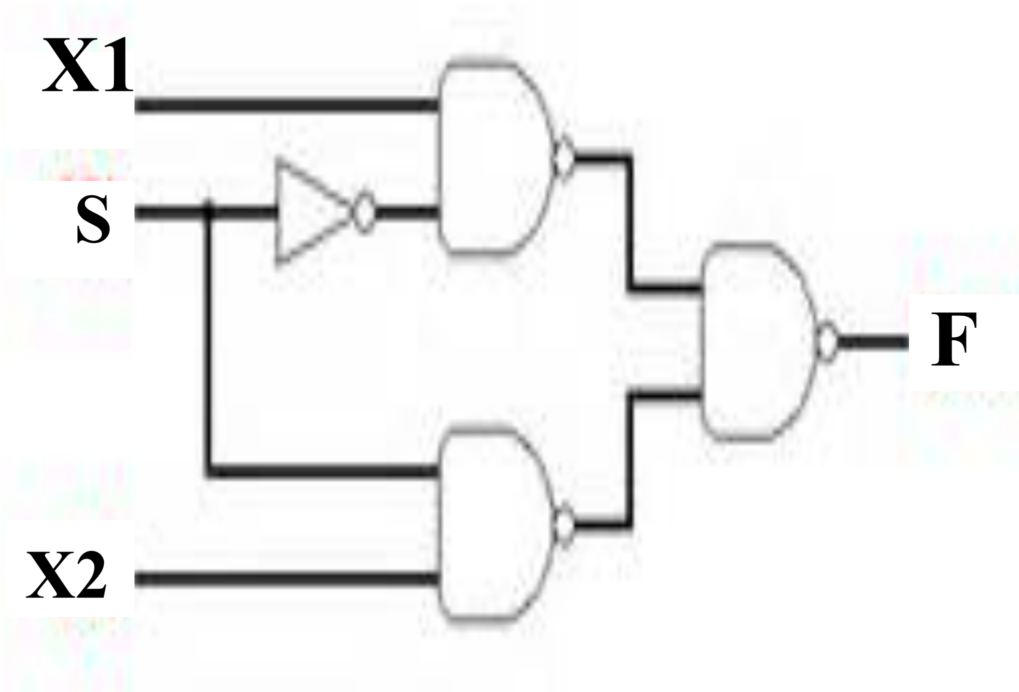
$$F = \bar{S}X1 + SX2$$

Basic concepts

This is a **well-posed** problem !

S	X1	X2	F
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

$$F = \bar{S}X1 + SX2$$



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

Machine learning is an ill-posed problem !

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

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

Machine learning is an ill-posed problem !

- There are $2^{16} = 65536$ possible boolean functions over four input features. We can't figure out which one is correct until we've seen every possible input-output pair. After 7 examples, we still have 2^9 possibilities.

x_1	x_2	x_3	x_4	y
0	0	0	0	?
0	0	0	1	?
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	?
1	0	0	0	?
1	0	0	1	1
1	0	1	0	?
1	0	1	1	?
1	1	0	0	0
1	1	0	1	?
1	1	1	0	?
1	1	1	1	?

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

Machine learning is an ill-posed problem !

- There are $2^{16} = 65536$ possible boolean functions over four input features. We can't figure out which one is correct until we've seen every possible input-output pair. After 7 examples, we still have 2^9 possibilities.

x_1	x_2	x_3	x_4	y
0	0	0	0	?
0	0	0	1	?
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	?
1	0	0	0	?
1	0	0	1	1
1	0	1	0	?
1	0	1	1	?
1	1	0	0	0
1	1	0	1	?
1	1	1	0	?
1	1	1	1	?

How can we learn the right boolean function F(x1, x2, x3, x4) if we have only 7 examples ?

We still have 2^9 possible values for the function F for the remaining 9 examples.

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

We should simplify the problem….

- We should reduce the space of all the possible functions F to be considered
- We should choose a *space of hypotheses* ***H*** that is smaller than the space of all the possible functions F

- For instance, we could choose:
  - simple conjunctive boolean functions
  - m-of-n boolean functions
  - linear functions
  - Gaussian functions
  - Etc.

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

We should simplify the problem….

For instance, we could choose simple **conjunctive** boolean functions

And check if one of the possible conjunctive boolean functions satisfies the 7 examples that we have

If one of the possible conjunctive boolean functions satisfies the 7 examples (**our training set**), we may hypothesise that it is the right function for our problem

And we may use it for future **unknown** examples.

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

Use of simple conjunctive boolean functions

F=x_1 :2 counterexamples

F=x_2 :6 counterexamples

….

F= $x_1 \wedge x_2$:3 counterexamples

No simple conjunctive boolean functions satisfy all the 7 examples of our training set

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

Use of m-of-n boolean functions

- At least *m* of the *n* variables must be true
- There are 32 possible rules
- Only one rule is consistent!

variables	Counterexample 1-of	2-of	3-of	4-of
$\{x_1\}$	3	–	–	–
$\{x_2\}$	2	–	–	–
$\{x_3\}$	1	–	–	–
$\{x_4\}$	7	–	–	–
$\{x_1, x_2\}$	3	3	–	–
$\{x_1, x_3\}$	4	3	–	–
$\{x_1, x_4\}$	6	3	–	–
$\{x_2, x_3\}$	2	3	–	–
$\{x_2, x_4\}$	2	3	–	–
$\{x_3, x_4\}$	4	4	–	–
$\{x_1, x_2, x_3\}$	1	3	3	–
$\{x_1, x_2, x_4\}$	2	3	3	–
$\{x_1, x_3, x_4\}$	1	***	3	–
$\{x_2, x_3, x_4\}$	1	5	3	–
$\{x_1, x_2, x_3, x_4\}$	1	5	3	3

Use of m-of-n boolean functions

–Please, check that the function 2-of-[x1, x3, x4] is the right function that satisfies the 7 examples (**our training set**)

- At least *m* of the *n* variables must be true
- There are 32 possible rules
- Only one rule is consistent!

	Counterexample			
variables	1-of	2-of	3-of	4-of
$\{x_1\}$	3	—	—	—
$\{x_2\}$	2	—	—	—
$\{x_3\}$	1	—	—	—
$\{x_4\}$	7	—	—	—
$\{x_1, x_2\}$	3	3	—	—
$\{x_1, x_3\}$	4	3	—	—
$\{x_1, x_4\}$	6	3	—	—
$\{x_2, x_3\}$	2	3	—	—
$\{x_2, x_4\}$	2	3	—	—
$\{x_3, x_4\}$	4	4	—	—
$\{x_1, x_2, x_3\}$	1	3	3	—
$\{x_1, x_2, x_4\}$	2	3	3	—
$\{x_1, x_3, x_4\}$	1	***	3	—
$\{x_2, x_3, x_4\}$	1	5	3	—
$\{x_1, x_2, x_3, x_4\}$	1	5	3	3

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

Two ways to simplify the problem, namely, to make it (partially) well posed

1)**Using prior knowledge** (if we have it)

If we know that the right model is a simple boolean conjunctive function, we can learn the right function from our training set

2)**Testing different models**

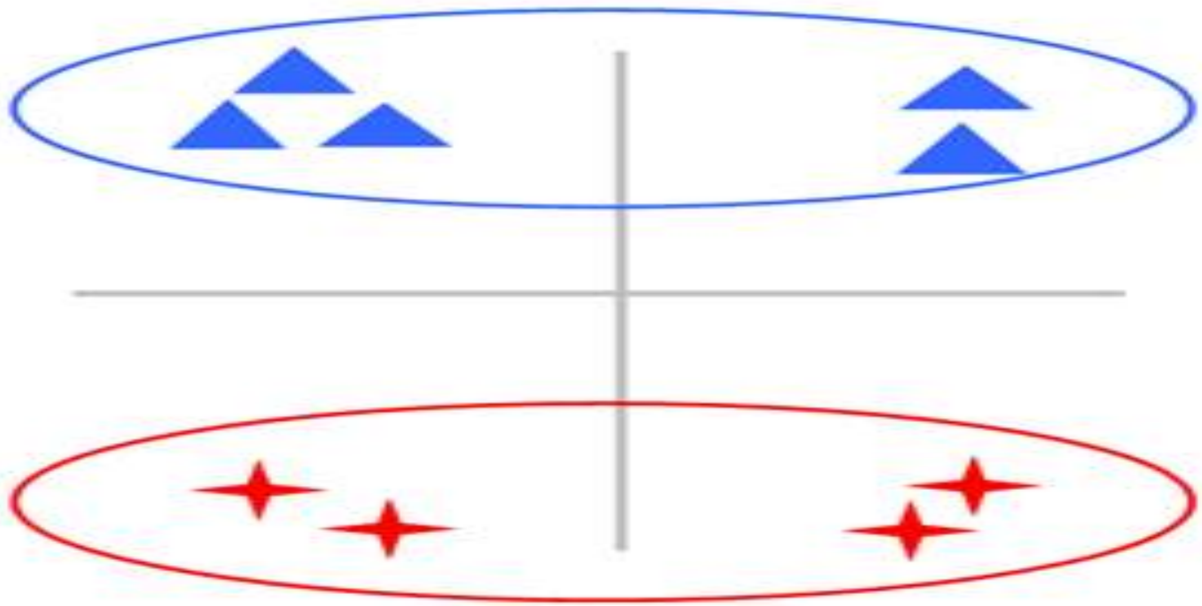
We can start with simple models and then make them more complex until that we find a model that satisfies the examples of our training set.

Basic concepts: feature selection

•It is easy to see that the choice of features is crucial for classification performance.

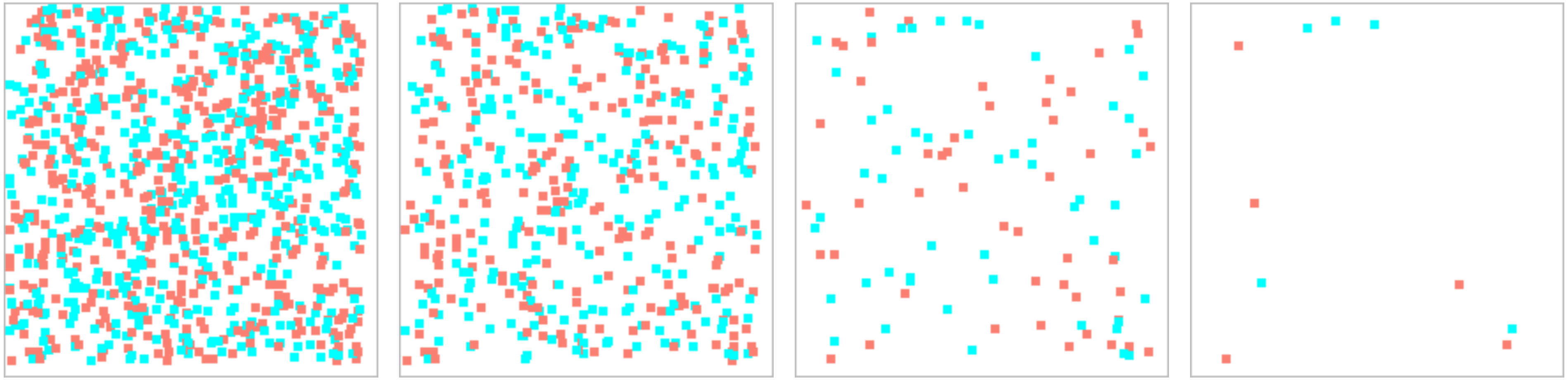
•What is the best feature in the figure below.?

•Feature selection algorithms can help one to select the best (sub)set of features. We will discuss that in further detail later.



Complexity of classification problems

Random Noise



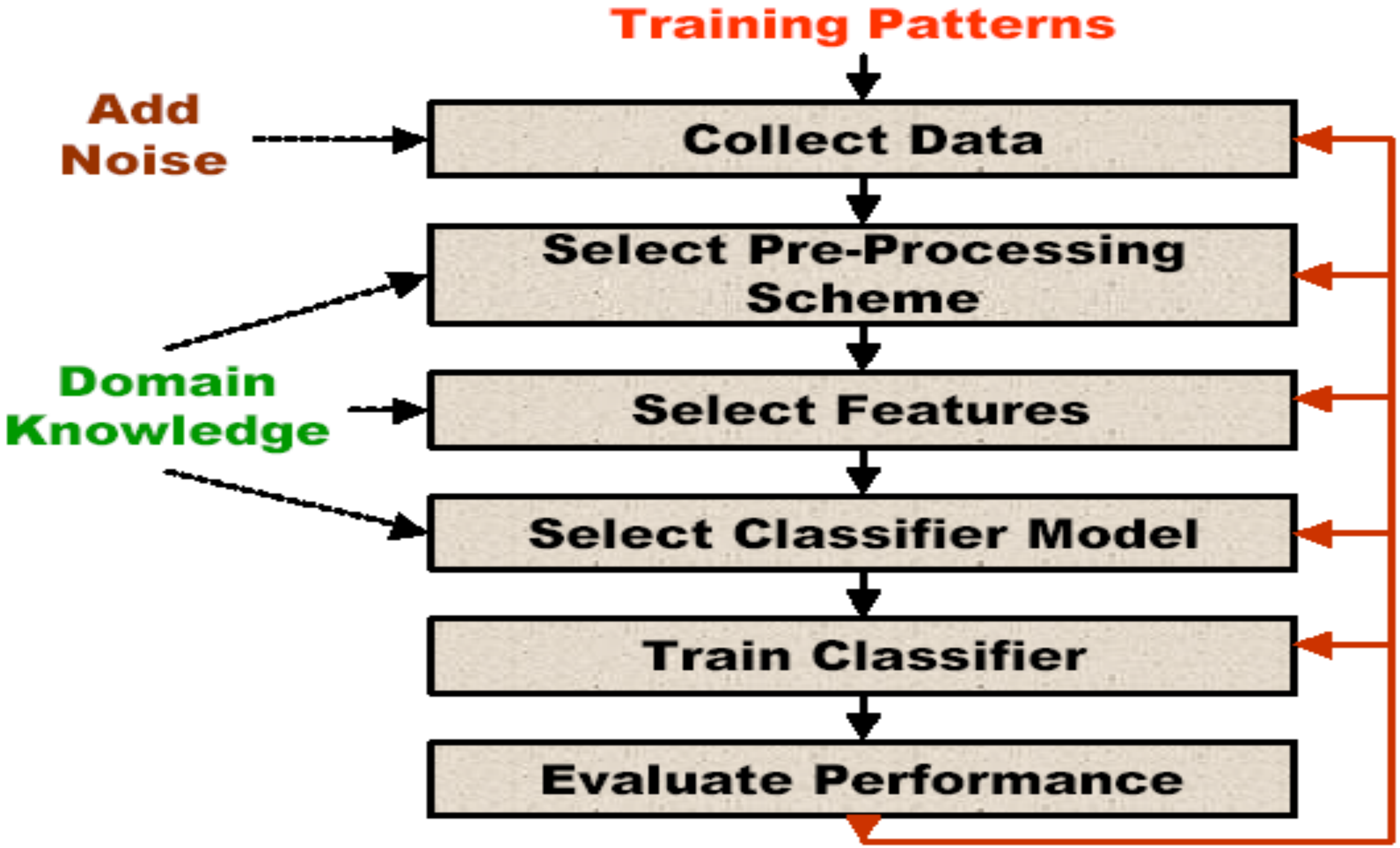
1000 samples 500 samples 100 samples 10 samples

➢We speak of **apparent** complexity as it depends on the size of the data set.

➢The problem looks easy with only 10 points.

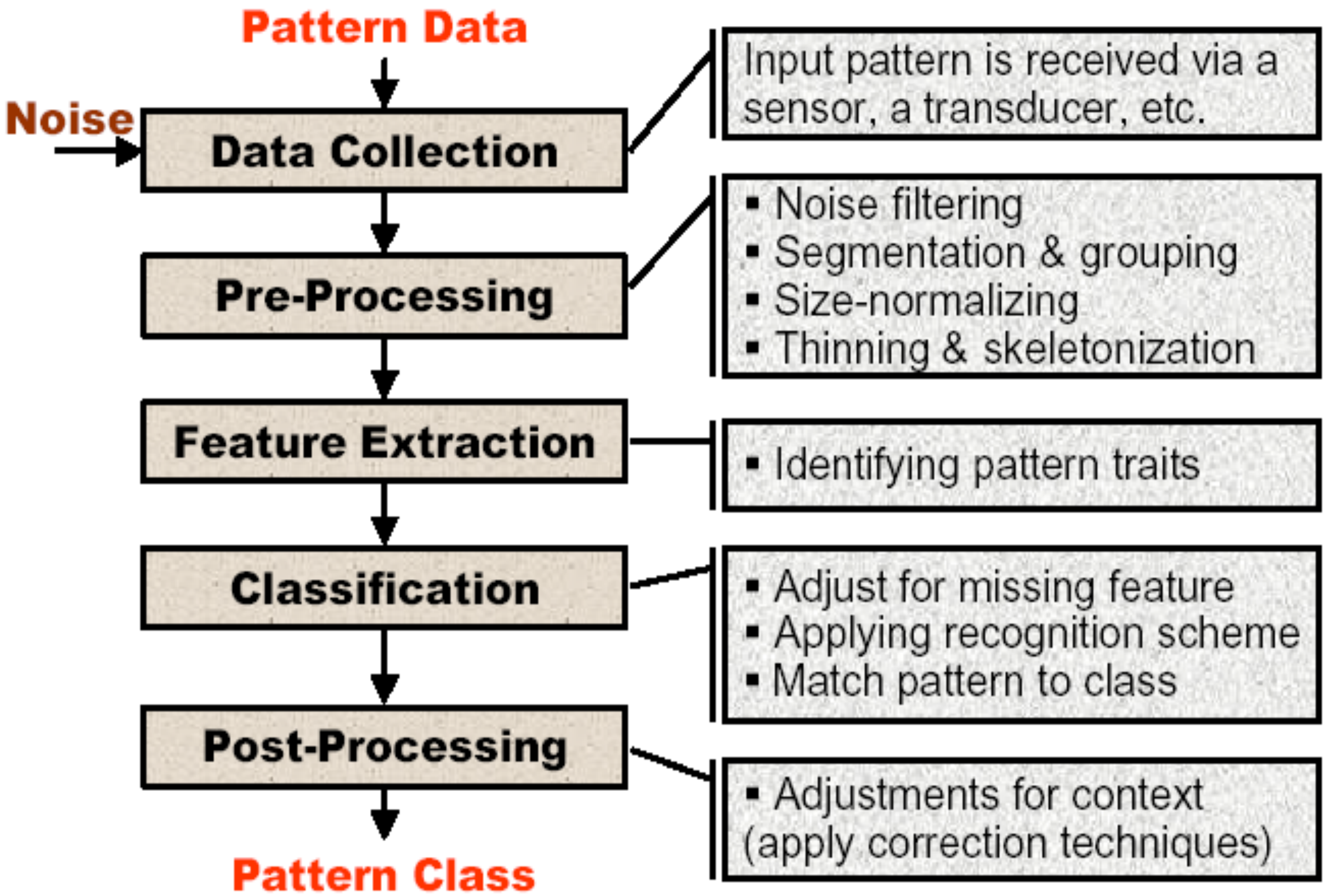
➢It is mandatory to have a data set large enough.

Design cycle of a pattern recognition system



Pattern Recognition course, Concordia University, instructor: Serban Iliescu

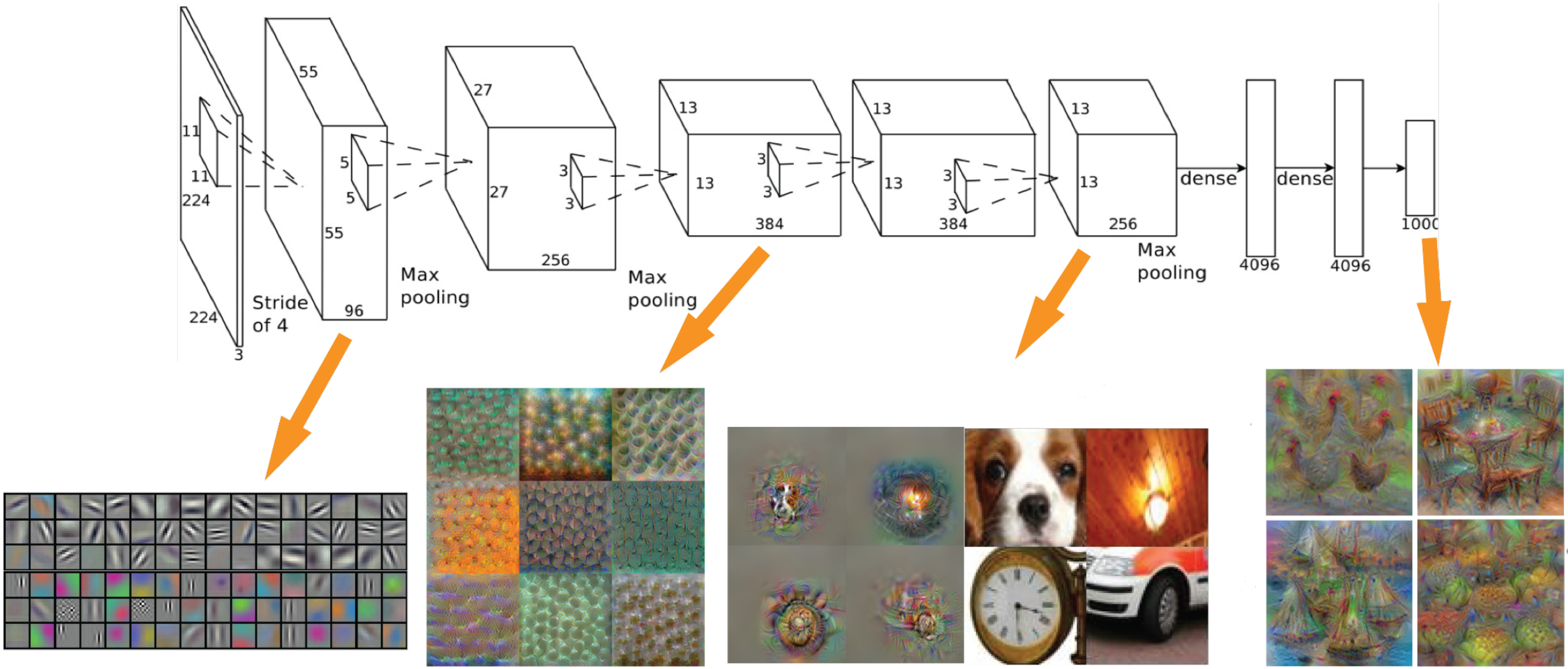
Architecture of a pattern recognition system



Pattern Recognition course, Concordia University, instructor: Serban Iliescu

Learning to do pattern recognition…

- Today, pre-processing, feature selection and classifier-model selection can be **partially** learned with **deep neural networks** (we will see them later).



Conv 1: **Conv 3:** **Conv 5:** **Fc8:**

What is the best representation for «patterns» ?

Different representations of patterns:

✓A vector of numerical features (in this course, we focus on this approach)

✓First-order logical formulas

✓Graphs

✓Semantic networks

✓Etc.

The best representation

Patterns belonging to the same class should be close in the feature space, and should be far away from patterns of different classes

Classification paradigms

Statistical approach (in this course, we focus on this approach)

✓Patterns characterized with a numerical feature vector

✓Classification model is based on probability density functions

Syntactic approach

✓Patterns are described with “rules”

✓Classification model is a “grammar”

Structural approach

✓Patterns are represented with graphs

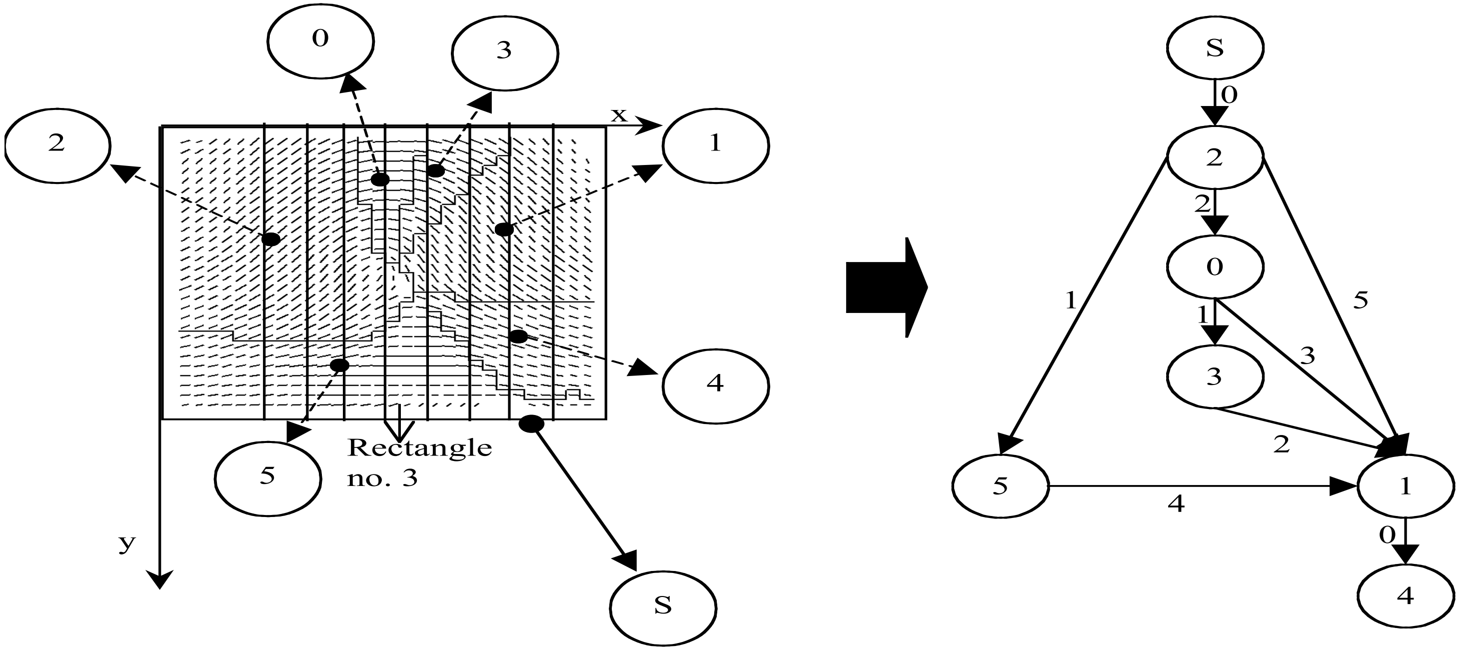
✓Classification model is a matching algorithm between graphs

Symbolic approach

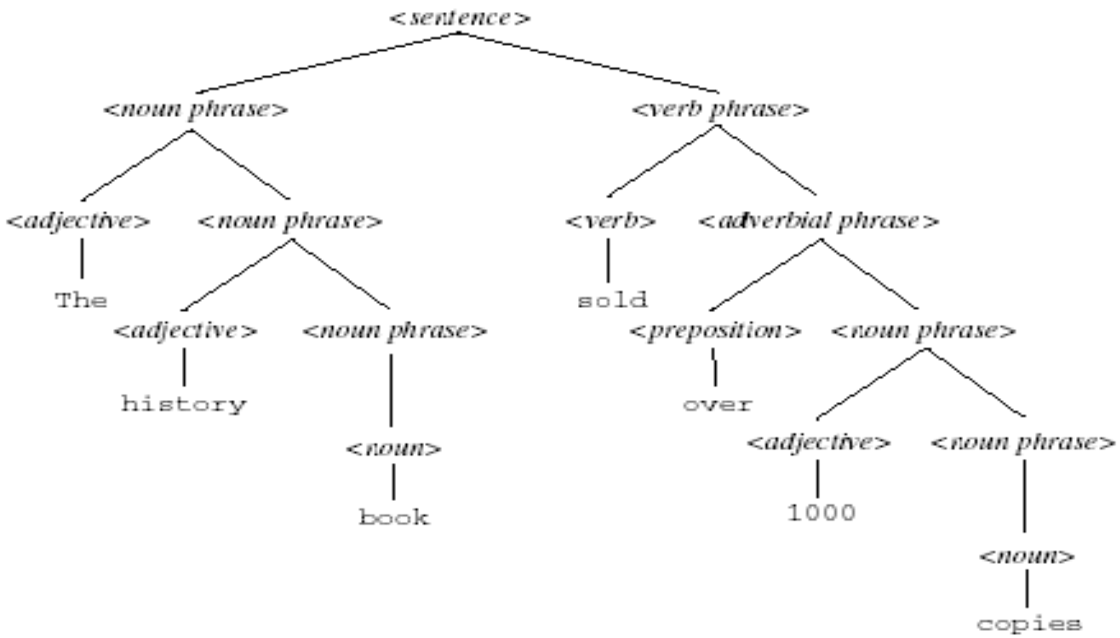
✓Patterns are described with symbolic models

✓Classification models are based on logical inference and symbolic processing

A quick note on structural and syntactic approach



Example of a fingerprint image represented with a graph [Yao, Roli, et al, Pattern Recognition, 2003]



Syntactic approach uses grammars.

An example of a grammar to analyse sentences.

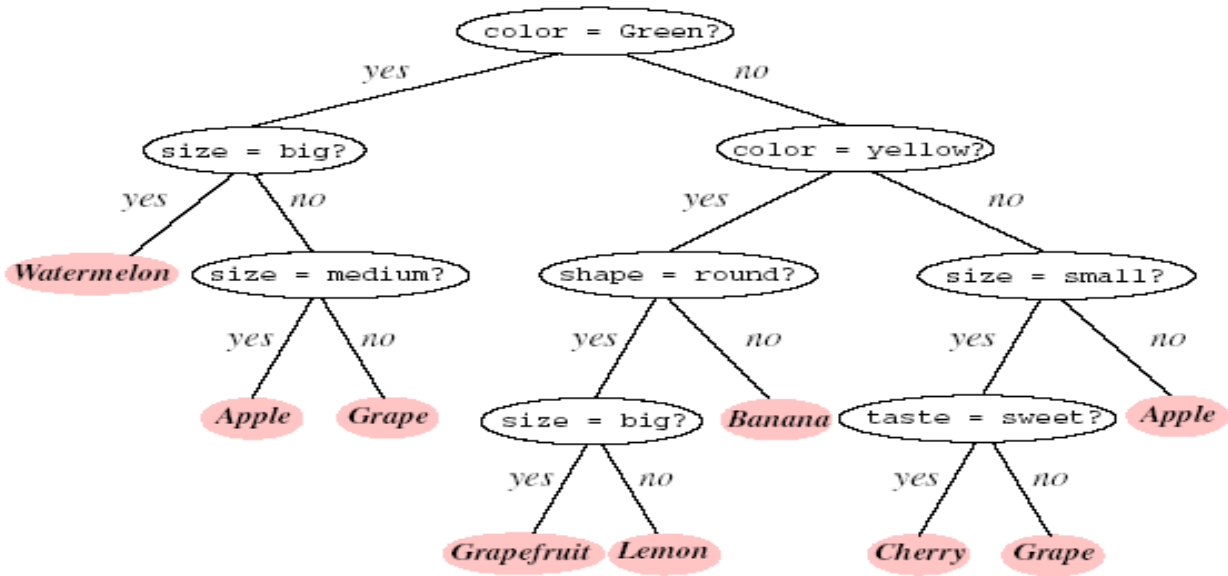
A quick note on symbolic approach

- In some applications, patterns can be characterized well with symbolic descriptions

IF the stain of the organism is gram negative
AND the morphology of the organism is rod
AND the aerobicity of the organism is anaerobic
THEN there is strongly suggestive evidence (0.8) that the class of the organism is Enterobacter iaceae.

A symbolic rule used by the expert system MYCIN designed to identify bacteria causing severe infections.

Medical diagnosis is naturally characterized by symbolic “rules”.



An example of a symbolic “decision tree” with categorical, discrete, features

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 (6 hours)
3. Introduction to pattern classification methods (2 hours)
4. Parametric methods (4 hours)
5. Non parametric methods and decision trees (4 hours)
6. Linear discriminant functions and support vector machines (4 hours)
7. Artificial neural networks (4 hours)
8. Performance evaluation (2 hours)
9. Clustering Methods (2 hours)
10. Adversarial machine learning (2 hours)
11. Exercises (12 hours)
12. 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 intermediate assessments instead of the home computer-exercise assignment
  - 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:** http://pralab.diee.unica.it/MachineLearning