Introduction to machine learning

Juan López González

University of Oviedo

Inverted CERN School of Computing, 24-25 February 2014

General overview

- **Lecture 1**
  - Machine learning
    - Introduction
    - Definition
    - Problems
    - Techniques
- **Lecture 2**
  - ANN
  - SOMs
    - Definition
    - Algorithm
  - Simulation
  - SOM based models

CERN School of Computing

LECTURE 1

Introduction to machine learning and data mining

1. Introduction

1.1. Some definitions
1.2. Machine learning vs Data mining
1.3. Examples
1.4. Essence of machine learning
1.5. A learning puzzle

1.1 Some definitions

- To **learn**
  - To use a set of observations to uncover an underlying process
- To **memorize**
  - To commit to memory
    - It doesn’t mean to understand

1.2 Machine learning vs Data mining

- **Machine learning** (Arthur Samuel)
  - Study, design and development of algorithms that give computers capability to learn without being explicitly programmed.
- **Data mining**
  - Extract knowledge or unknown patterns from data.

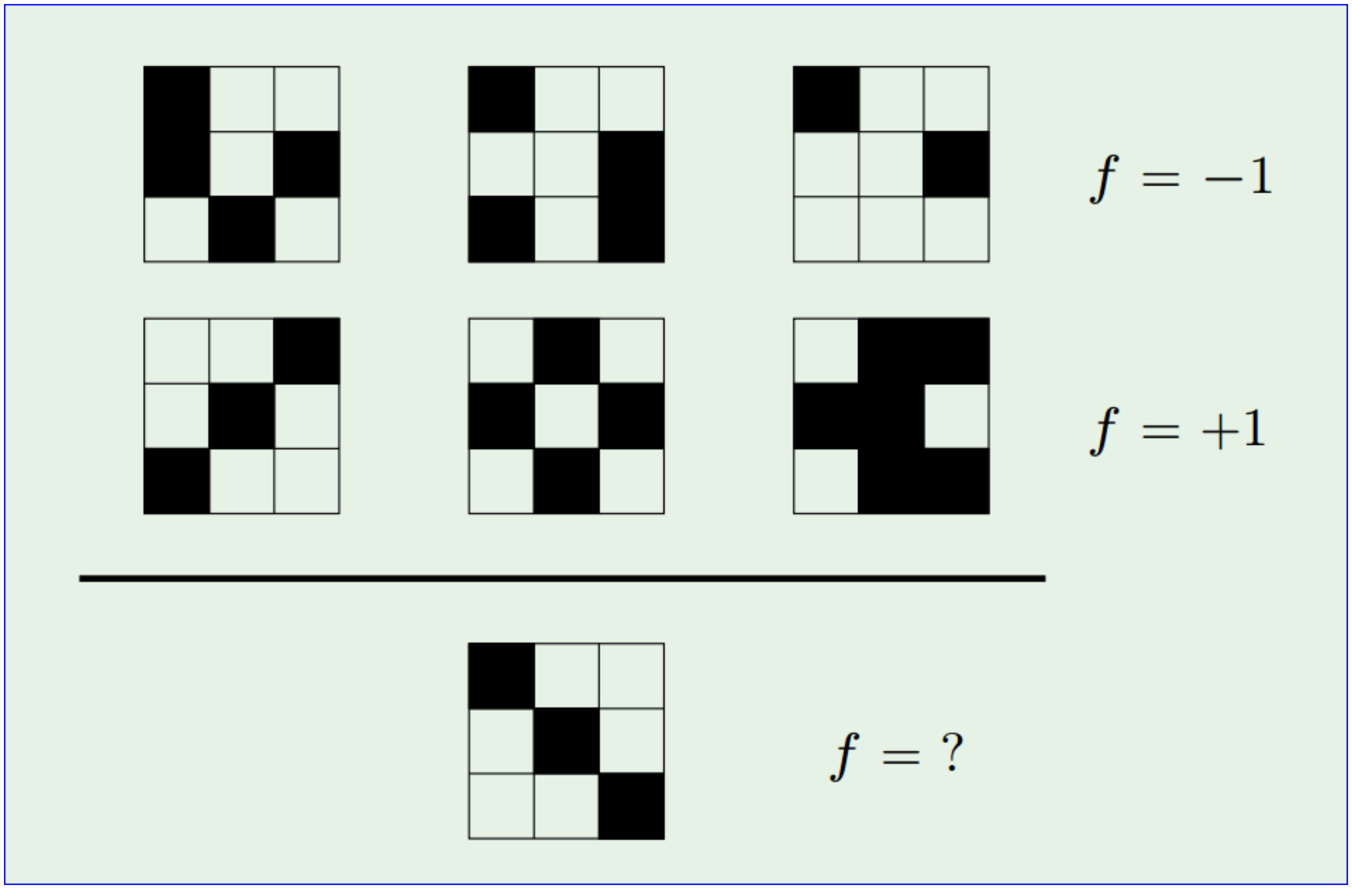
1.3 Examples

- **Credit approval**
  - Gender, age, salary, years in job, current debt…
- **Spam filtering**
  - Subject, From…
- **Topic spotting**
  - Categorize articles
- **Weather prediction**
  - Wind, humidity, temperature…

1.4 Essence of machine learning

- A pattern exists
- We cannot pin it down mathematically
- We have data on it

1.5 A learning puzzle



2. Definition

2.1. Components
2.2. Generalization and representation
2.3. Types of learning

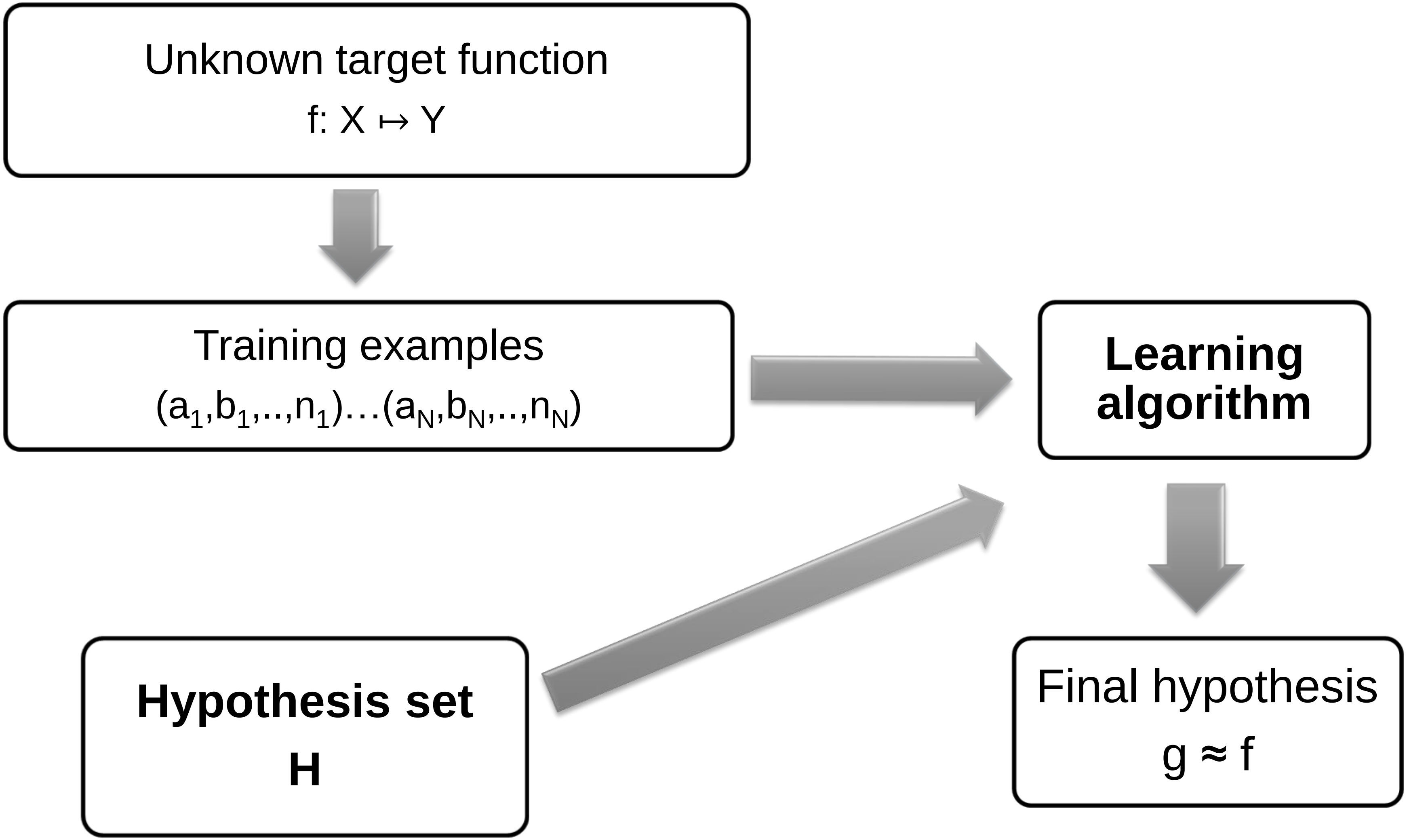
CERN School of Computing

2.1 Components

- Input (customer application)
- Ouput (aprove/reject credit)
- Ideal function ($f: X \mapsto Y$)
  - Data: $(a_1,b_1,..,n_1), (a_2,b_2,..,n_2) \ldots (a_N,b_N,..,n_N)$ (historical records)
  - Result: $(y_1), (y_2) \ldots (y_N)$ (loan paid or not paid)
- Hypothesis ($g: X \mapsto Y$)

CERN School of Computing

2.1 Components



2.2 Generalization and representation

- **Generalization**
  - The algorithm has to build a general model
  - **Objective**
    - Generalize from experience
    - Ability to perform accurately for unseen examples
- **Representation**
  - Results depend on input
    - Input depends on representation
      - Pre-processing?

2.3 Types of learning

- Supervised
  - Input and output
- Unsupervised
  - Only input
- Reinforcement
  - Input, output and grade of output

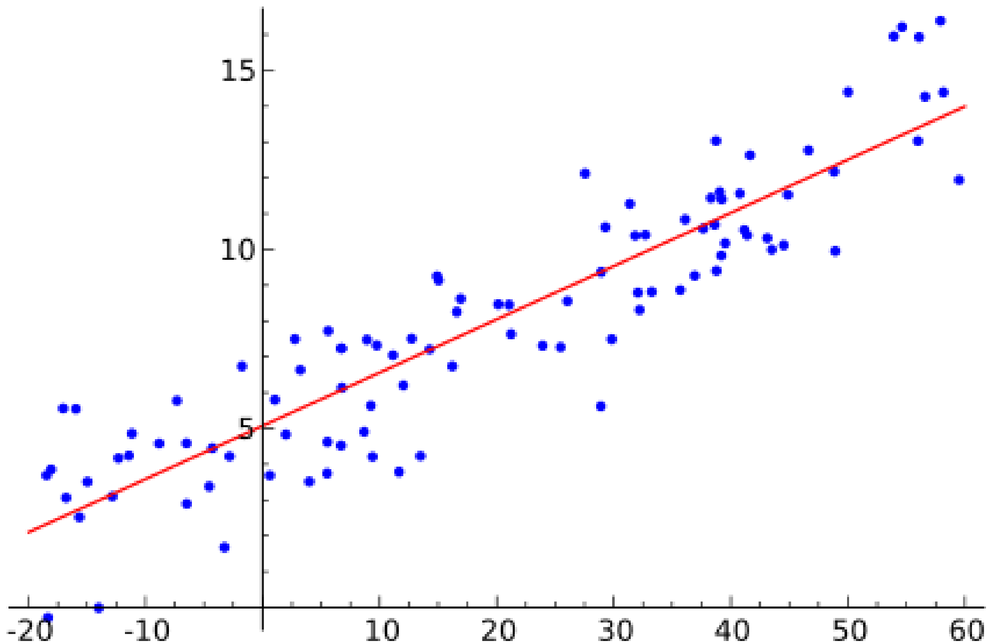
CERN School of Computing

3. Problems

3.1. Regression
3.2. Classification
3.3. Clustering
3.4. Association rules

3.1 Regression

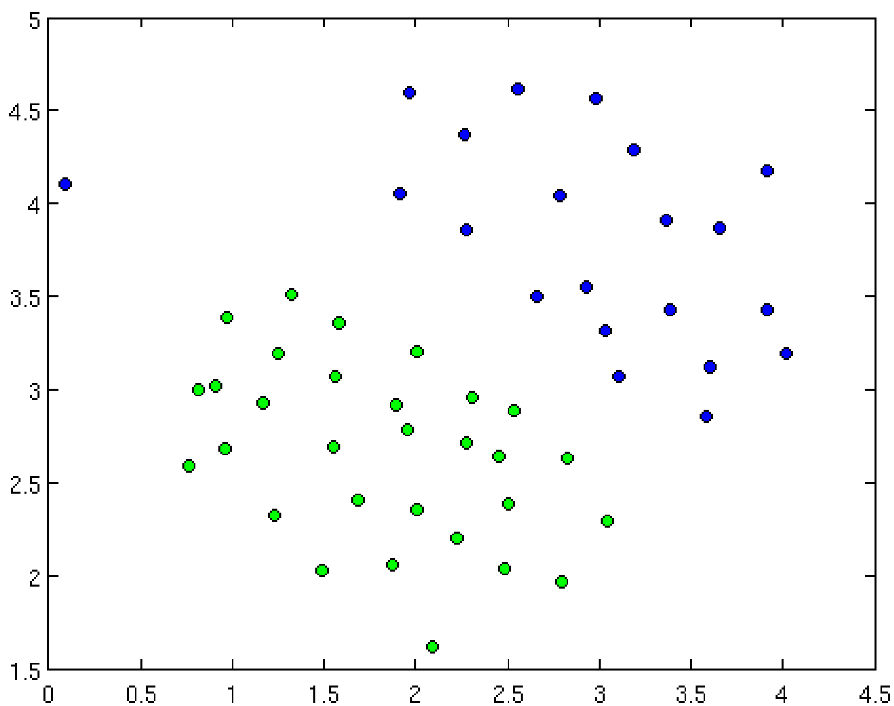
- **Statistical process for estimating the relationships among variables**
  - Could be used for prediction

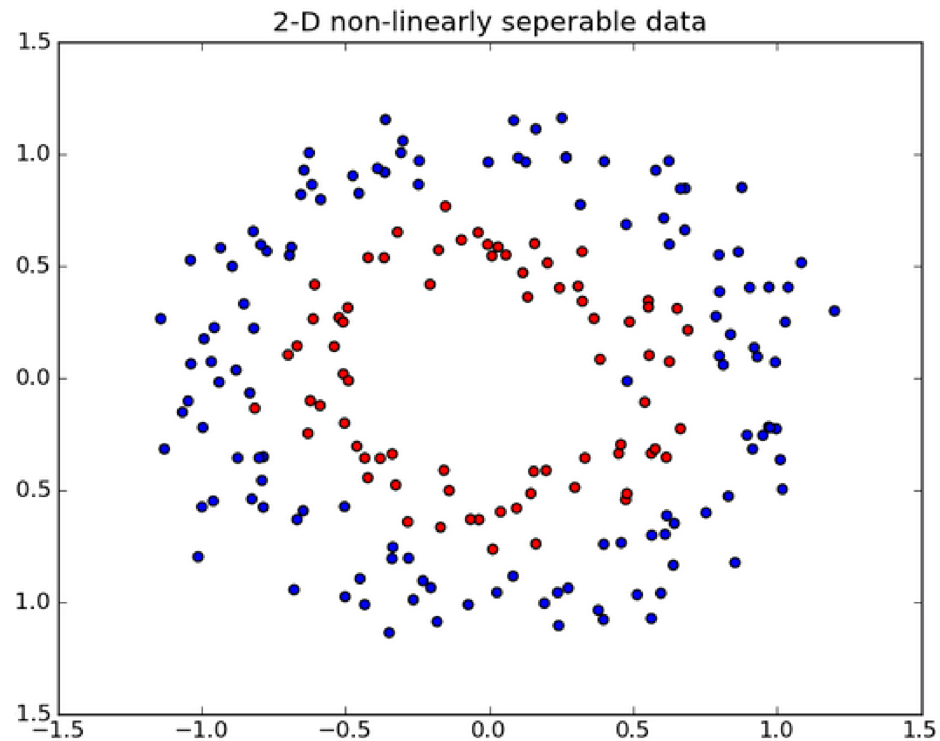


CERN School of Computing

3.2 Classification

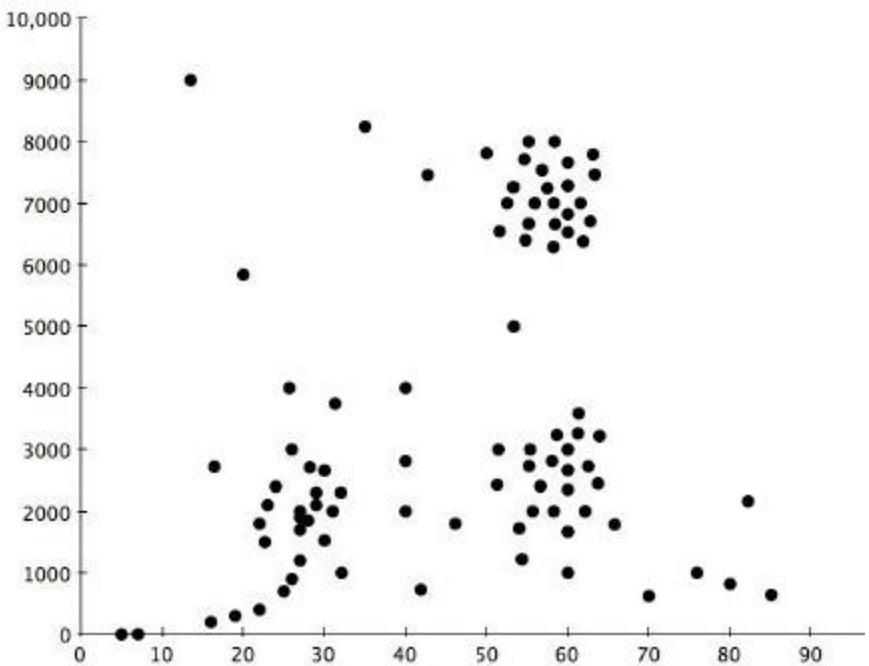
- **Identify to which of a set of categories a new observation belongs**
  - Supervised learning

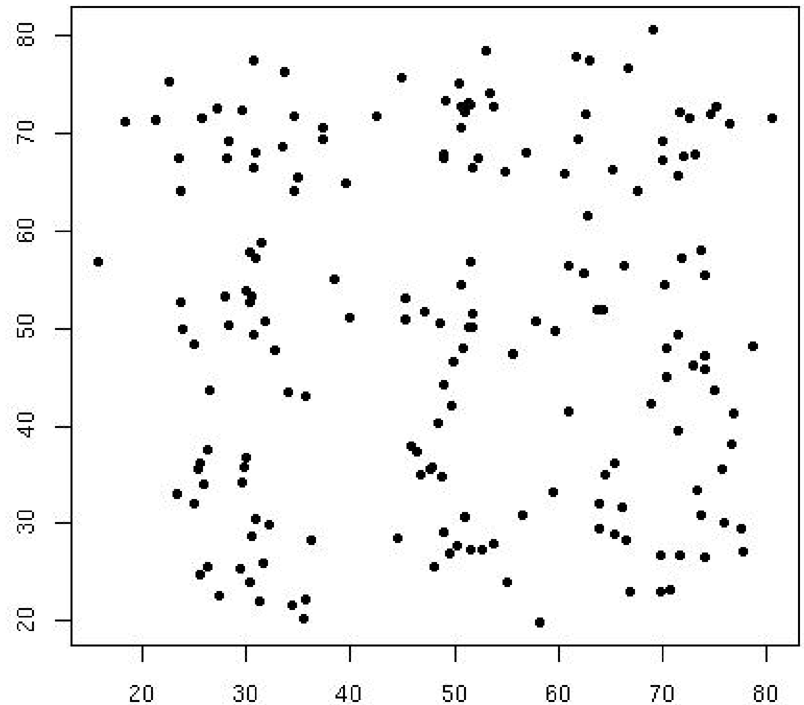




3.3 Clustering

- **Grouping a set of objects in such a way that objects in the same group are more similar**
  - Unsupervised learning

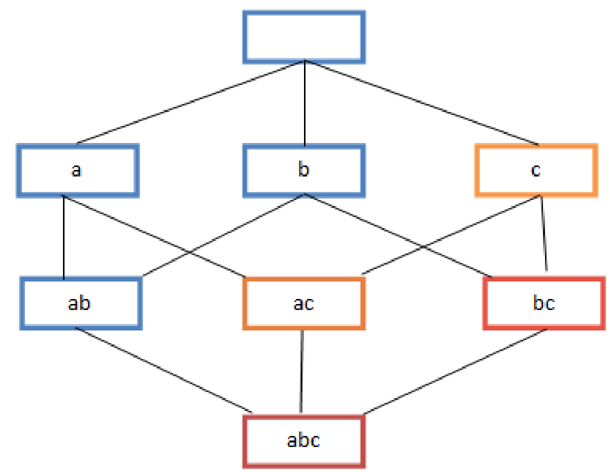




3.4 Association rule

- **Discovering relations between variables in large databases**
  - Based on 'strong rules'
  - If order matters -> Sequential pattern mining

	A	B	C
1	0	0	1
2	1	0	1
3	1	1	0
4	1	0	0
5	0	1	0



frequent *itemset* lattice

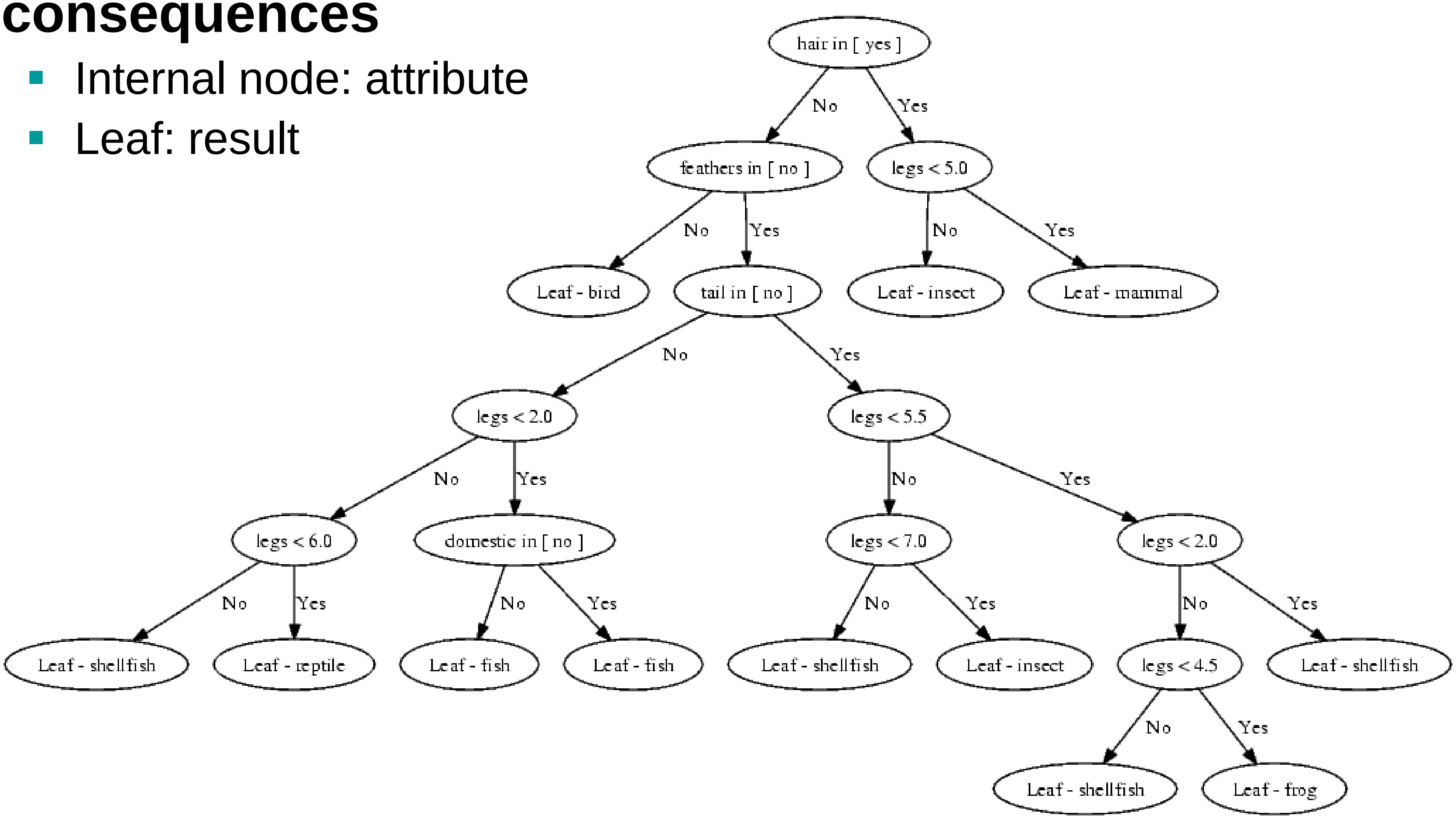
4. Techniques

4.1. Decision trees
4.2. SVM
4.3. Monte Carlo
4.4. K-NN
4.5. ANN

CERN School of Computing

4.1 Decision trees

- **Uses tree-like graph of decisions and possible consequences**
  - Internal node: attribute
  - Leaf: result



CERN School of Computing

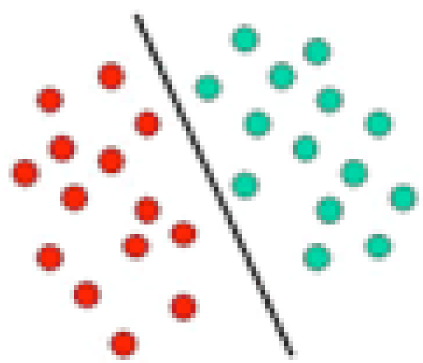
4.1 Decision trees

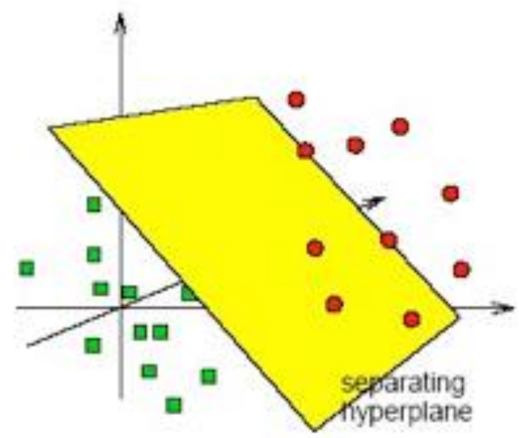
- Results human readable
- Easily combined with other techniques
  - Possible scenarios can be added
- Expensive

- Ex: C4.5
  - Information entropy

4.2 Support Vector Machine (SVM)

- **Separates the graphical representation of the input points**
  - Constructs a hyperplane which can be used for classification
    - Input space transformation helps
    - Non-human readable results

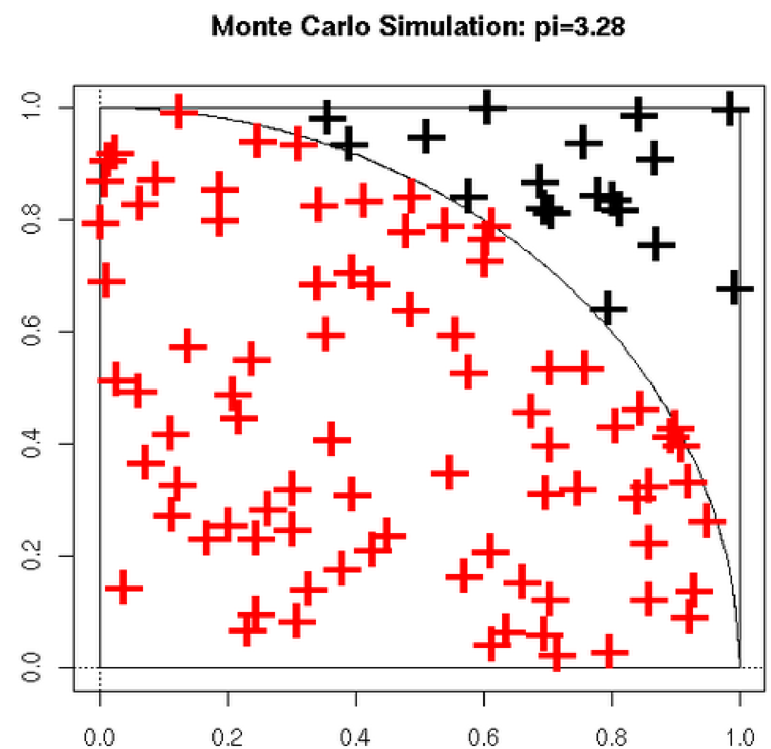




CERN School of Computing

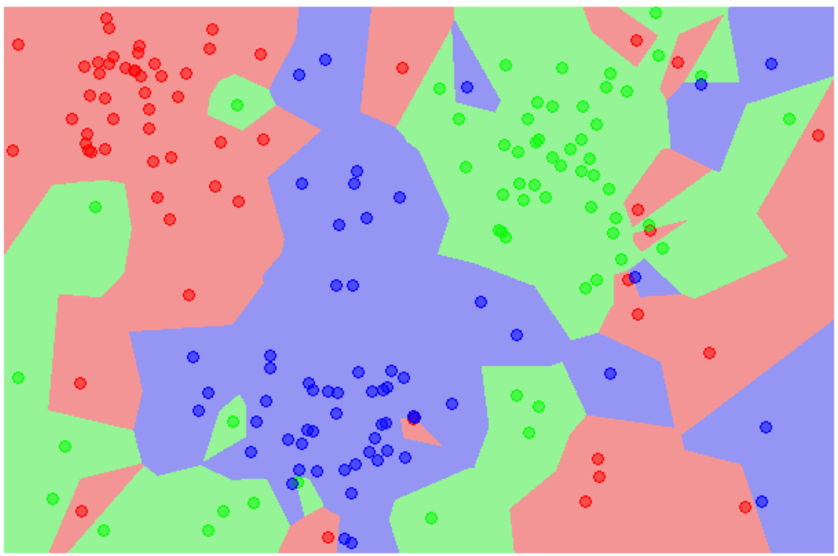
4.3 Monte Carlo

- **Obtain the distribution of an unknown probabilistic entity**
  - Random sampling to obtain numerical results
- **Applications**
  - Physics
  - Microelectronics
  - Geostatistics
  - Computational biology
  - Computer graphics
  - Games
  - …

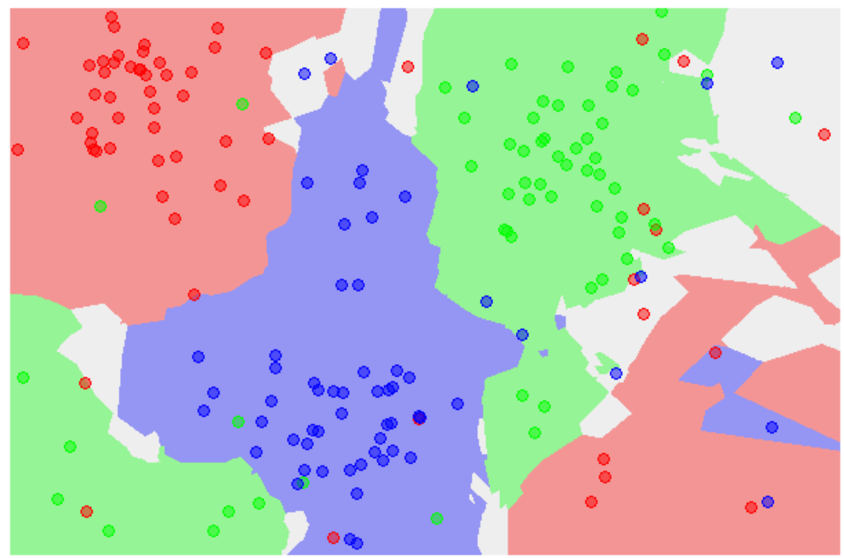


4.4 K-Nearest neighbors (K-NN)

- **Classifies by getting the class of the K closest training examples in the feature space**



K=1



K=5

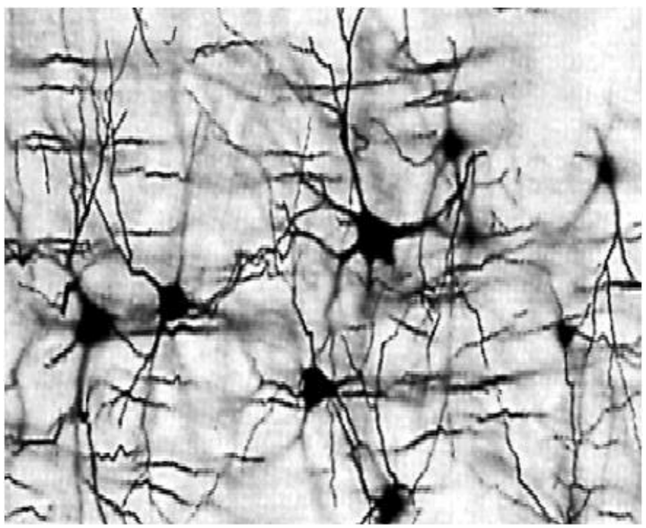
CERN School of Computing

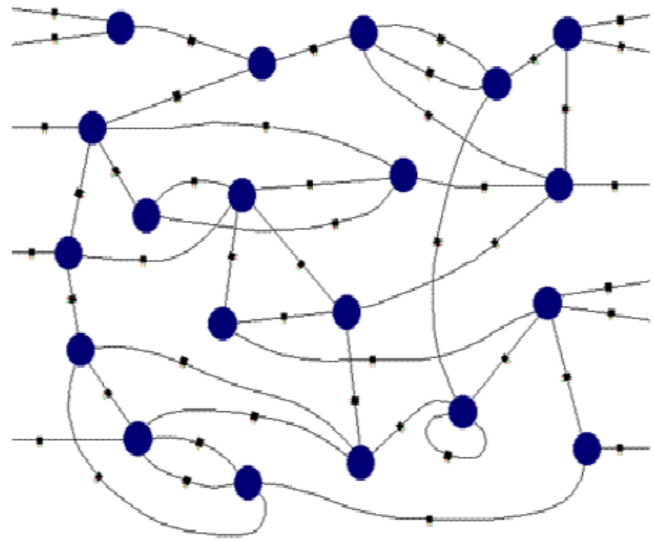
4.4 K-Nearest neighbors (K-NN)

- Easy to implement
  - naive version
- High dimensional data needs dimension reduction
- Large datasets make it computational expensive
- Many k-NN algorithms try to reduce the number of distance evaluations performed

4.5 Artificial neural networks (ANN)

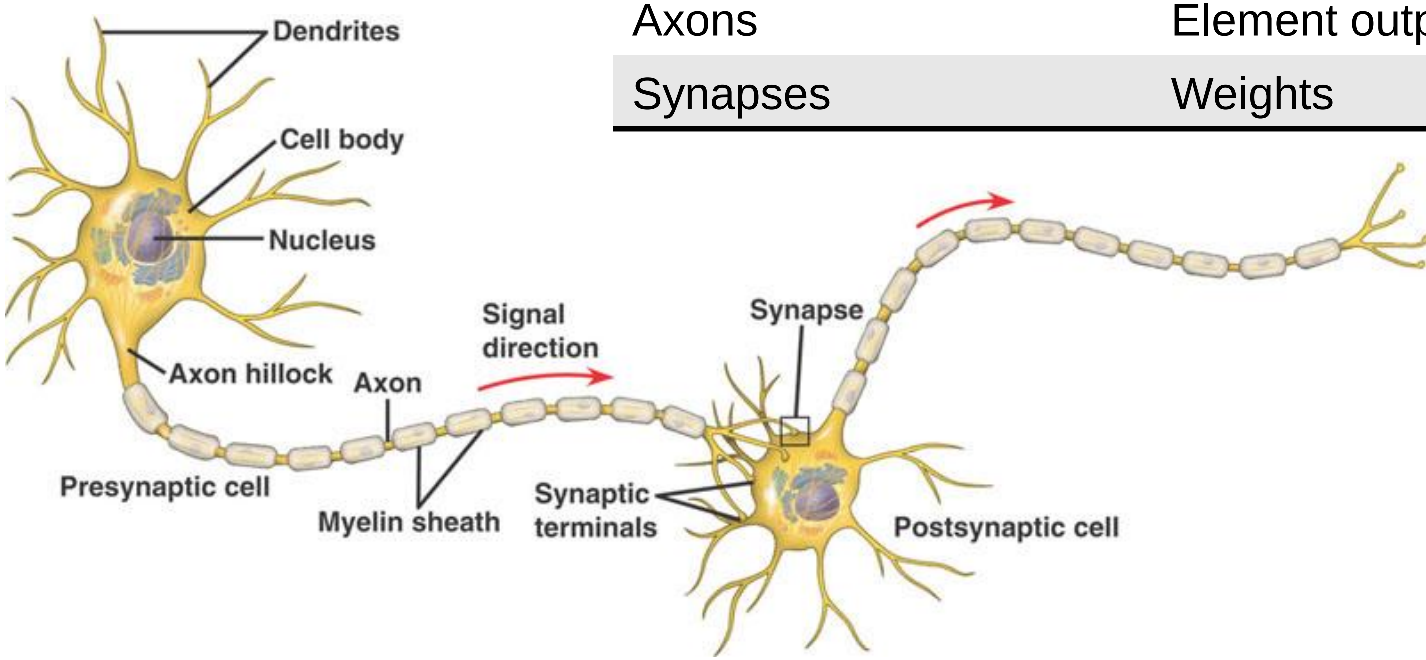
- **Systems of interconnected neurons that compute from inputs**





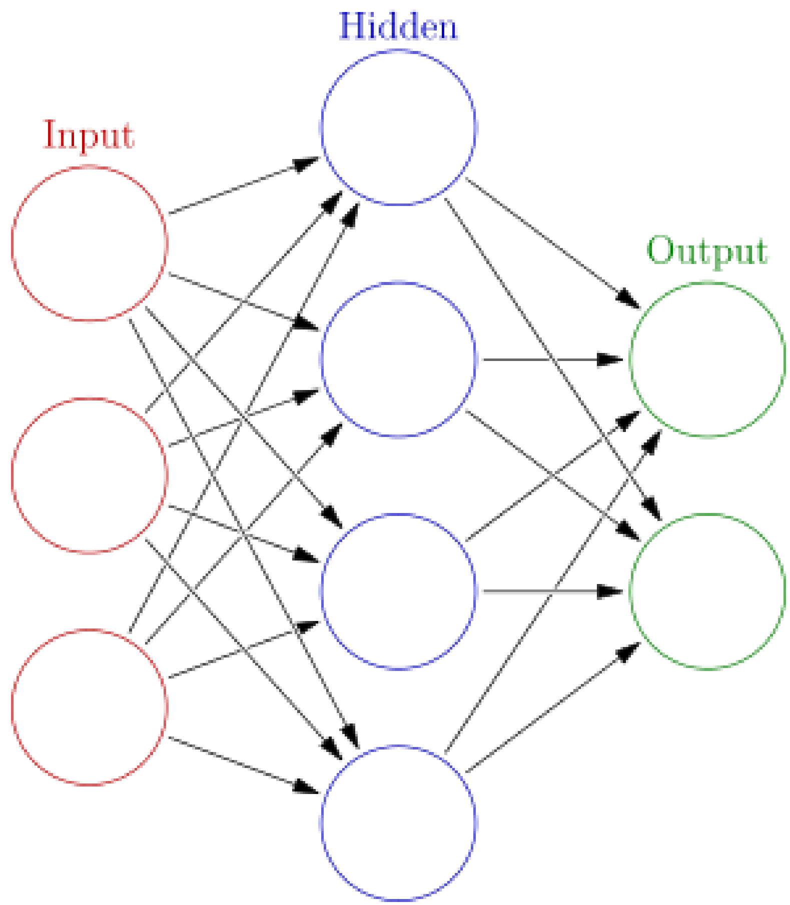
4.5 Artificial neural networks (ANN)

Human	Artificial
Neuron	Processing element
Dendrites	Combining function
Cell body	Transfer function
Axons	Element output
Synapses	Weights



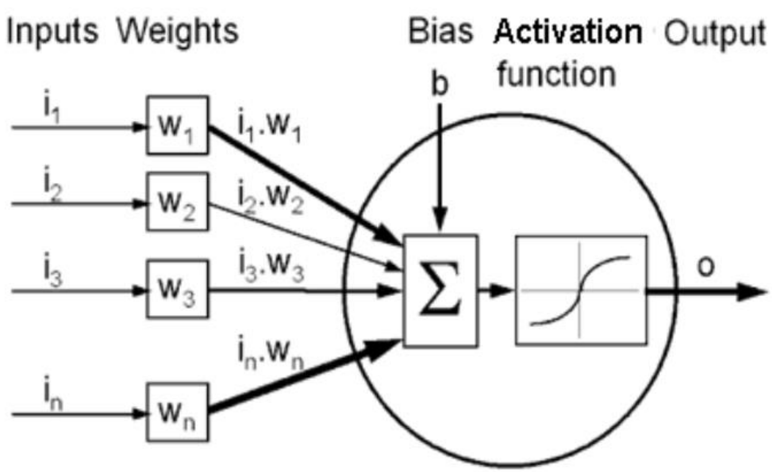
4.5 Artificial neural networks (ANN)

Example:



4.5 Artificial neural networks (ANN)

- **Perceptron**
  - single-layer artificial network with one neuron
  - *calculates the linear combination of its inputs and passes it through a threshold activation function*



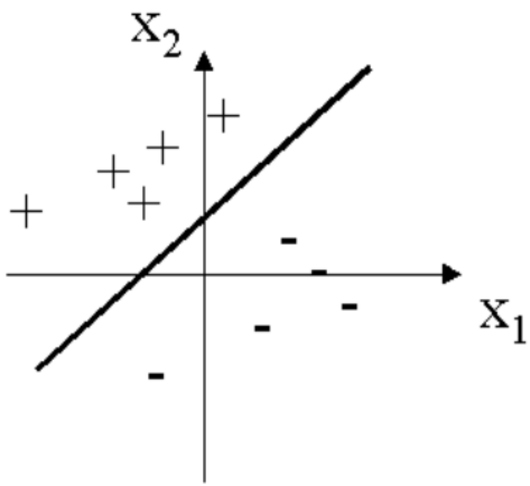
$$y = \operatorname{sgn}\left(\sum_{i=1}^{2} w_i x_i + \theta\right)$$

$$\operatorname{sgn}(s) = \begin{cases} 1 & \text{if } s > 0 \\ -1 & \text{otherwise.} \end{cases}$$

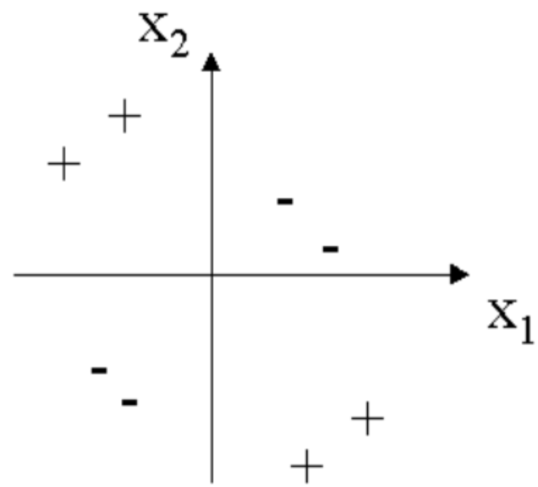
Equivalent to a linear discriminant

4.5 Artificial neural networks (ANN)

- **Perceptron**



Linearly Separable



Not Linearly Separable

$$w_1 x_1 + w_2 x_2 + \theta = 0$$

Equivalent to a linear discriminant

4.5 Artificial neural networks (ANN)

- **Learning**
  - Learn the weights (and threshold)
  - Samples are presented
    - If output is incorrect adjust the weights and threshold towards desired output
    - If the output is correct, do nothing

Q & A