Course Introduction

Machine Learning

Daniele Loiacono

POLITECNICO
MILANO 1863

Information

- **Daniele Loiacono** (Instructor)
  - Contact: daniele.loiacono@polimi.it - +39 02 2399 **3615**
  - Office: DEIB, room 150
  - https://beep.metid.polimi.it/web/2019-20-machine-learning-daniele-loiacono-/
- Teaching Assistants
  - Mirco Mutti
  - Lorenzo Bisi


- Assessment
  - Written test (closed-book)
    - Questions, exercises, code
    - *See examples on Beep*
  - *Homework / Challenge*

Information (2)

- Logistics
  - Thu, 12.15 - 14.15, 5.0.3, Group 2 (presence)
  - Fri, 14.15 - 16.15, 3.0.2, Group 1 (presence)
  - Fri, 08.15 - 11.15, ALL (online)
- Students with odd ID (codice persona) will be in Group 1, others in Group 2
- Practical classes
  - will be (mainly) in presence
  - will cover exam-like exercises and practical examples
  - will present practical examples using **Python** (bring your laptop!)
- Interact
  - Feel free to ask questions*
  - Try in practice what you learn

* as much as possible with distance learning

References

- You will have access to all the materials used in classroom but slides are not an alternative to textbooks!
  - Slides are based on the material of prof. Restelli
- Supervised Learning
  - **Bishop, "Pattern Recognition and Machine Learning", Springer, 2006.**
  - Hastie, Tibshirani, Friedman, "The Elements of Statistical Learning: Data Mining, Inference, and Prediction", Springer, 2009.
  - Mitchell, "Machine Learning", McGraw Hill, 1997.
- Reinforcement Learning
  - **Sutton and Barto, "Reinforcement Learning: an Introduction", MIT Press, 1998.** New draft available at: http://www.incompleteideas.net/book/the-book-2nd.html

What will you learn?

Goals

- Learn to correctly **model** machine learning problems
- Learn the **principles** and the **main techniques** of ML
- Learn how to **assess** the performances of ML models
- Learn **limitations** of ML techniques and how to **choose** the most appropriate one for your problem
- Provide the basic background to understand latest developments in this field

Topics

- Linear Regression
- Linear Classification
- Bias-Variance
- Model Selection
- PAC-Learning and VC dimension
- Kernel Methods
- Support Vector Machines
- Markov Decision Processes
- Dynamic Programming
- RL in finite MDPs
- Multi-armed bandit

Reinforcement learning

What you will not learn?

Other courses

- A course of 5 credits is **not enough** to cover Machine Learning
- Fortunately, there are **other courses** that deal with other machine learning topics not covered in this course:
  - Data Mining and Text Mining
  - Soft Computing
  - Artificial Neural Networks and Deep Learning
  - Applied Statistics
  - Model Identification and Data Analysis

What is Machine Learning?

Why and when to apply it?

A method to solve a task without explicitly program a calculator

What is Machine Learning

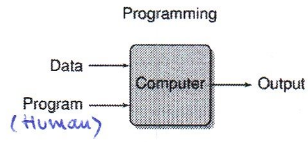
"A computer program is said to learn from **experience E** with respect to some class of **tasks T** and **performance measure P, improves with experience E**" *Mitchell (1997)*

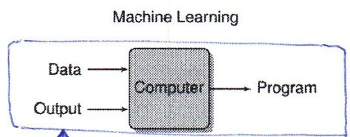
ML is something that uses experience to improve its performance

- ML is the sub-field of AI where the **knowledge** comes from:
  - **Experience**
  - **Induction**
- **However,** Machine learning **is not magic!**
  - You need to know **how it works**
  - You need to know **how to use it**
  - It can **extract** information from data, **not create** information

Why Machine Learning?

- We need computers to **make informed decisions** on new, unseen data
  - Often it is too difficult to design a set of meaningful rules
  - Machine learning allows to **automatically extract relevant information** from previous data and exploit it on new one
- Getting computers to **program themselves (automating automation)**
  - writing software is the **bottleneck**
  - let the **data** do the work instead





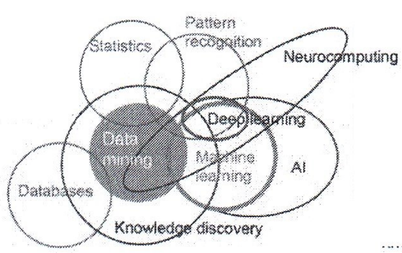
providing some data and some desired outputs we want the computer to learn the program: we want to remove the hand-written program. When is this meaningful? When the program is too difficult (too expensive or too complicated to generalize)

The human factor is in the preparation of the data, the desired output and in the writing of the machine learning algorithm (3 human factor)

Machine Learning applications

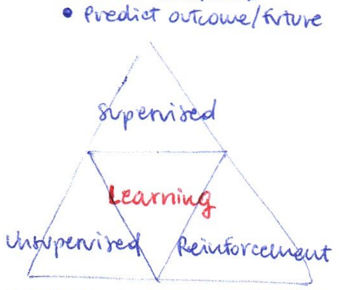
- Machine learning is very popular today and has several applications:
  - Computer vision and robotics
  - Speech recognition
  - Biology and medicine
  - Finance
  - Information retrieval, Web search, ...
  - Entertainment and Videogames
  - Space exploration
  - Education
  - ...

Machine Learning and other fields



Source: SAS, 2014 and PwC, 2016

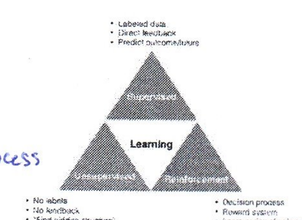
- labeled data
- Direct feedback
- Predict outcome/future



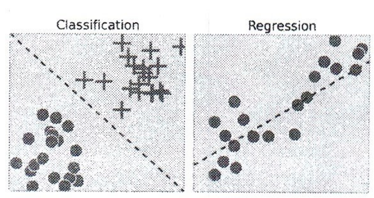
- No labels
- No feedback
- "Find hidden structure"

- Decision process
- Reward system
- Learn series of actions

Learning Paradigms in ML

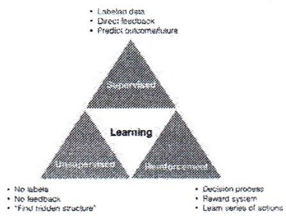


- Supervised Learning : mimic of the learning by examples: we have some labeled data (the label represents the desired (correct) output for a specific datum)

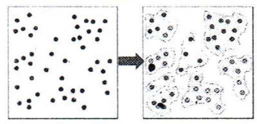


- **classification**: labels represent classes (categories)
- **regression**: labels are numerical values (quantitative values)

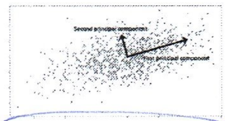
Learning Paradigms in ML



❑ Unsupervised Learning : we have unstructured data and we have to find a structure (we look for patterns, regularities, similarities or structures)



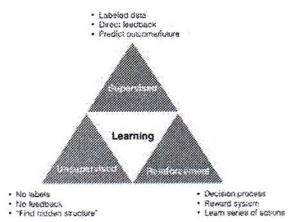
Clustering



Dimensionality Reduction

see if the data can be seen in a more compact way (express the same informations with less variables)

Learning Paradigms in ML



❑ Reinforcement Learning : learning through a reward mechanism: we have a problem/task and there are different available actions. We don't know which ones are "correct", in fact there are no "correct" actions. The ones desired are those which lead to rewards.

we provide the algorithm with actions and feedbacks (which can be rewards or negative feedbacks)

Supervised Learning

- ❑ Goal
  - Learn from **data** a **model** that maps **known inputs** to **known outputs**
  - Training set: $\mathcal{D} = \{\langle x, t\rangle\} \Rightarrow t = f(x)$: input x and target t
- ❑ Tasks
  - Classification : t discrete
  - Regression : t continuous
  - Probability estimation : t is a probability

we want to learn a function that allows to map input x to the target t. Depending on what is t we have different tasks.

- ❑ Techniques
  - Linear Models
  - Artificial Neural Networks
  - Support Vector Machines
  - Decision trees
  - etc.

! The training set needs to be representative for the test sample we want to apply the model to. If we train a model to predict the age of humans, we cannot obtain the age of an animal.

An example of classification

x Training Set t

0
1
1
0

Testing

0
0
1
0

To the same data can correspond different fitting models: for example it can be that they're women and men, long and short hair or else. However the test reveils that it is "has a smile with teeth or not" (different model ⇔ same data)

⇒ we learn something from the training set, however many models can fit a training set.

An example of regression

Training Set

78
34
25

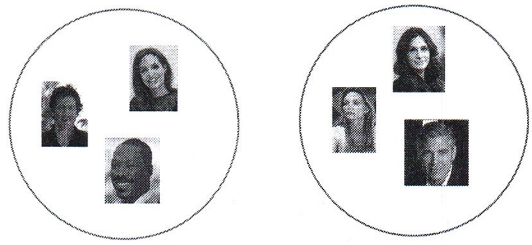
Testing

78
81
41

Unsupervised Learning

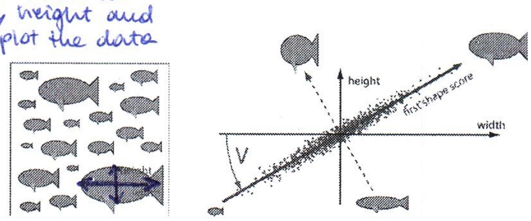
- Goal
  - Learn **previously unknown patterns** and **efficient data representation**
  - Training set: $\mathcal{D} = \{x\} \Rightarrow f(x)$: we still want to learn a function that maps inputs into different clusters (in case of clustering) or a function that maps an input into a lower-dimensional representation (dimensionality reduction case) – features space (space where inputs are more compactly represented)
- Tasks
  - Dimensionality Reduction
  - Clustering
- Techniques
  - K-means
  - Self-organizing maps
  - Principal Component Analysis
  - etc.

An example of clustering



An example of dimensionality reduction

For a group of fishes we consider only height and width. If we plot the data we discover that height and width are not the optimal features to represent the data.



In fact the only feature that matters seems to be the combination of height and width. If we go from 2-dimensional space to 1-dim space we don't loose too much of information but gain with dim-reduction.

Reinforcement Learning

- Goal
  - Learning the **optimal policy** = optimal behaviour to solve a task
  - Training set: $\mathcal{D} = \{\langle x, u, x', r\rangle\} \Rightarrow \pi^*(x) = arg\max_u \{Q^*(x,u)\}$ to estimate
- Problems
  - Markov Decision Process (MDP)
  - Partially Observable MDP (POMDP)
  - Stochastic Games (SG)
- Techniques
  - Q-learning
  - SARSA
  - Fitted Q-iteration
  - etc.

This function maps every situation and action to a possible reward that we can get in a long term

- x = current situation
- u = action
- x' = situation after the action
- r = reaction/reward

$\underbrace{\pi^*(x)}_{\text{what we want to learn}}$ = action to do when the situation is x

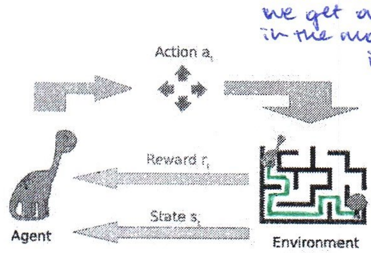
An example of reinforcement learning

: a dinosaur has to reach a tree to eat and it has to go through a maze. At each step we get a feedback: goodness of the position in the maze. The available actions are moving in one of the 4 directions. These actions lead the dinosaur in a new state where the dinosaur may get a reward.

$\pi(s_t)$ = policy function which tells how to behave at the state s_t

a_t = action provided from $\pi(s_t)$

$Q(s_t, a_t)$ = function that given the state and an action will provide the expected reward in the long term



It can be any reward function that who design the task wants to guide the player to the correct solution.

E.g. 0 if the dinosaur didn't reach the tree, 100 if he did. Repeating many times the dinosaur will find the shortest path.