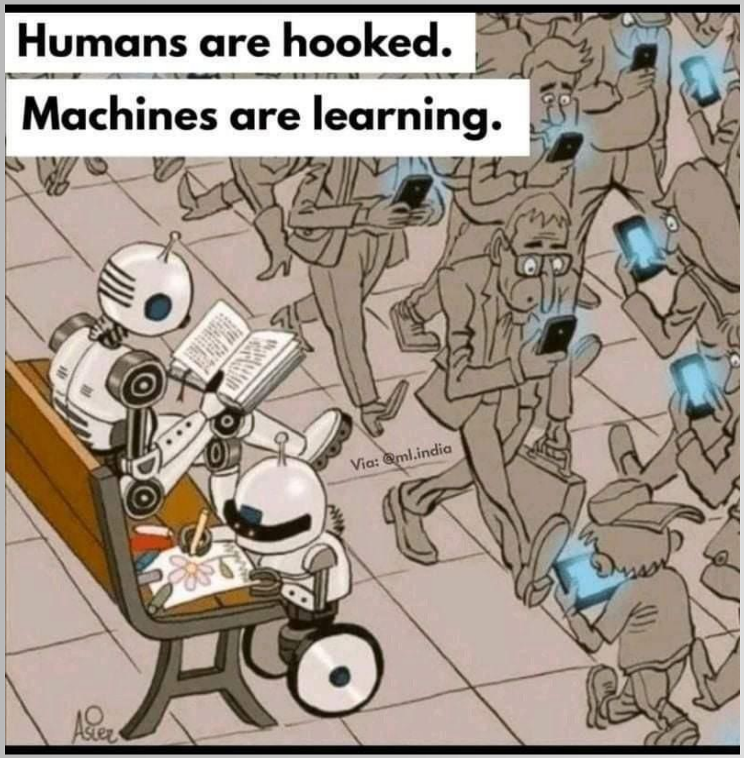


Machine Learning (Intro)

A.Y. 2022/23

Prof. Fabio Aiolli
www.math.unipd.it/~aiolli

Class Schedule

- ✓40 hours in classroom (5cfu)
- ✓8 hours in lab (1cfu)

- ✓Classroom: Monday from 2:30 pm to 4:30 pm and Wednesday from 2:30 pm to 4:30 pm, in P150 (Complesso Paolotti)
- ✓LABS: Yet to be defined (one at the beginning of the course and the others later towards the end)

Material and Exam

- T. Mitchell, "Machine Learning", McGraw Hill, 1998
- E. Alpaydin, "Introduction to Machine Learning", Cambridge University Press, 2010
- C.M. Bishop, "Pattern Recognition e Machine Learning" Springer, 2006
- Ian Goodfellow, Yoshua Bengio, Aaron Courville, “Deep Learning” MIT Press, 2016

- Slides and other material presented during the course

- Exam: 1/4 project (or activity, EP) and 3/4 written (+ oral optional)

Course Activity (Experience Points)

Not necessarily a real project!

It will consist of the application of machine learning techniques to one or more datasets from UCI (https://archive.ics.uci.edu/ml/datasets.php), kaggle (https://www.kaggle.com), or other repositories. The evaluation will be incremental.

Any collaborative activity posted on the moodle will be awarded:

- description of new tasks or datasets
- reporting practical activities and results
- revision and/or extension of other students' activities

E.g. MNIST e Kannada MNIST

https://www.kaggle.com/c/digit-recognizer/

https://www.kaggle.com/c/Kannada-MNIST

Please, subscribe to the MOODLE a.s.a.p.

Alternatively, it is always possible to do a project after the end of the course. The project has to be submitted (and evaluated) BEFORE the written test.

Where should you start..
to avoid falling behind

Linear Algebra (vectors and matrices):

- Youtube ('linear algebra for machine learning')
- Chapter 2 (Goodfellow et al.)

Probability:

- Appendix A (Alpaydin)
- Chapter 3 (Goodfellow et al.)

Python and scientific libraries:

- http://www.python.it/doc/
- Look at numpy, scipy, matplotlib, sklearn: these modules are already available on Google CoLab (a lab will be devoted to this topic soon)

Deductive Reasoning

- Deductive (Aristotele 384 BC – 322 AD)
  - RULE($C \rightarrow R$): All men are mortal
  - CASE(C_1): Socrate is a man
  - Then..
  - RESULT(R_1): Socrate is mortal

- Deductive reasoning is the foundation of math proofs and theorems. Starting from true assumptions, new theorems are inferred by means of their proofs (logical consequences). Very popular in 'classic' AI as well.

Inductive Reasoning

- Inductive (F. Bacon, philosopher 1561–1626)
  - CASE(C_1): Socrate is a man
  - RESULT(R_1): Socrate is mortal
  - Then..
  - RULE($C \rightarrow R$): All men are mortal

- In inductive reasoning, the premises (particular cases) provide evidence supporting the conclusion (*generalization*) but they do not guarantee its truth (it is possible to have true premises and false conclusions). This is what we use in ML and what scientists in general use.

Abductive Reasoning

- Abductive (C.S. Peirce, 1839–1914)
  - RULE($C \rightarrow R$): All men are mortal
  - RESULT(R_1): Socrate is mortal
  - Then..
  - CASE(C_1): Socrate is a man

- In abductive reasoning (instead of generalizing) we move «laterally», assuming an implication to be valid even on the contrary. This is what Sherlock Holmes and Dr. Gregory House use.

Dr House

- DEDUCTION
  - RULE: All lupus sufferers die within 5 days
  - CASE: The patient has lupus
  - RESULT: The patient will die within 5 days
- INDUCTION
  - CASE: The patient has lupus
  - RESULT: The patient died in 5 days
  - RULE: All lupus sufferers die within 5 days
- ABDUCTION
  - RULE: All lupus sufferers die within 5 days
  - RESULT: The patient died in 5 days
  - CASE: The patient had lupus

What is 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

Key questions..
(which we will attempt to answer)

When and why is a machine learning approach useful?

How is it possible to learn?

But most important of all (which we will answer later in the course):
Can we actually learn?

Algorithm

From WIKIPEDIA:

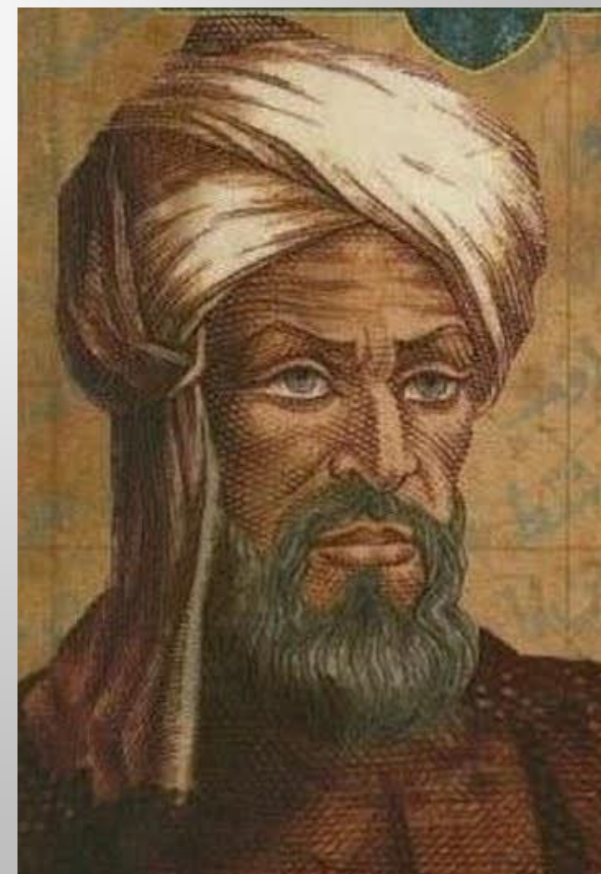
In mathematics and computer science, an **algorithm** (/ˈælɡərɪðəm/ (listen)) is a finite sequence of well-defined, computer-implementable instructions, typically to solve a class of specific problems or to perform a computation.[1][2] Algorithms are always unambiguous and are used as specifications for performing calculations, data processing, automated reasoning, and other tasks. In contrast, a heuristic is a technique used in problem solving that uses practical methods and/or various estimates in order to produce solutions that may not be optimal but are sufficient given the circumstances.[3][4]

As an effective method, an algorithm can be expressed within a finite amount of space and time,[5] and in a well-defined formal language[6] for calculating a function.[7] Starting from an initial state and initial input (perhaps empty),[8] the instructions describe a computation that, when executed, proceeds through a finite[9] number of well-defined successive states, eventually producing "output"[10] and terminating at a final ending state. The transition from one state to the next is not necessarily deterministic; some algorithms, known as randomized algorithms, incorporate random input.[11]

The concept of algorithm has existed since antiquity. Arithmetic algorithms, such as a division algorithm, were used by ancient Babylonian mathematicians c. 2500 BC and Egyptian mathematicians c. 1550 BC.[12] Greek mathematicians later used algorithms in 240 BC in the sieve of Eratosthenes for finding prime numbers,[13] and the Euclidean algorithm for finding the greatest common divisor of two numbers.[14] Arabic mathematicians such as al-Kindi in the 9th century used cryptographic algorithms for code-breaking, based on frequency analysis.[15]

The word *algorithm* itself is derived from the name of the 9th-century mathematician Muḥammad ibn Mūsā al-Khwārizmī, whose nisba (identifying him as from Khwarazm) was Latinized *as Algoritmi*.[16] A partial formalization of the modern concept of algorithm began with attempts to solve the *Entscheidungsproblem* (decision problem) posed by David Hilbert in 1928. Later formalizations were framed as attempts to define "effective calculability"[17] or "effective method".[18] Those formalizations included the Gödel–Herbrand–Kleene recursive functions of 1930, 1934 and 1935, Alonzo Church's lambda calculus of 1936, Emil Post's Formulation 1 of 1936, and Alan Turing's Turing machines of 1936–37 and 1939.

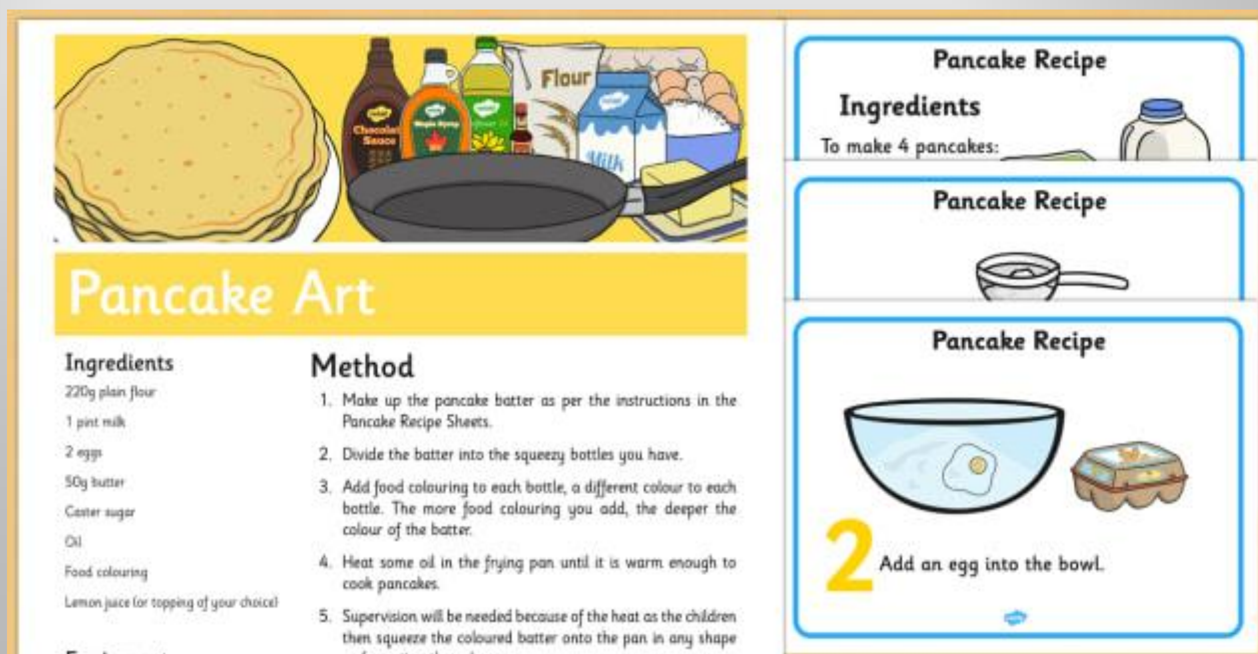
«A finite sequence of well-defined, computer-implementable instructions, typically to solve a class of specific problems or to perform a computation»



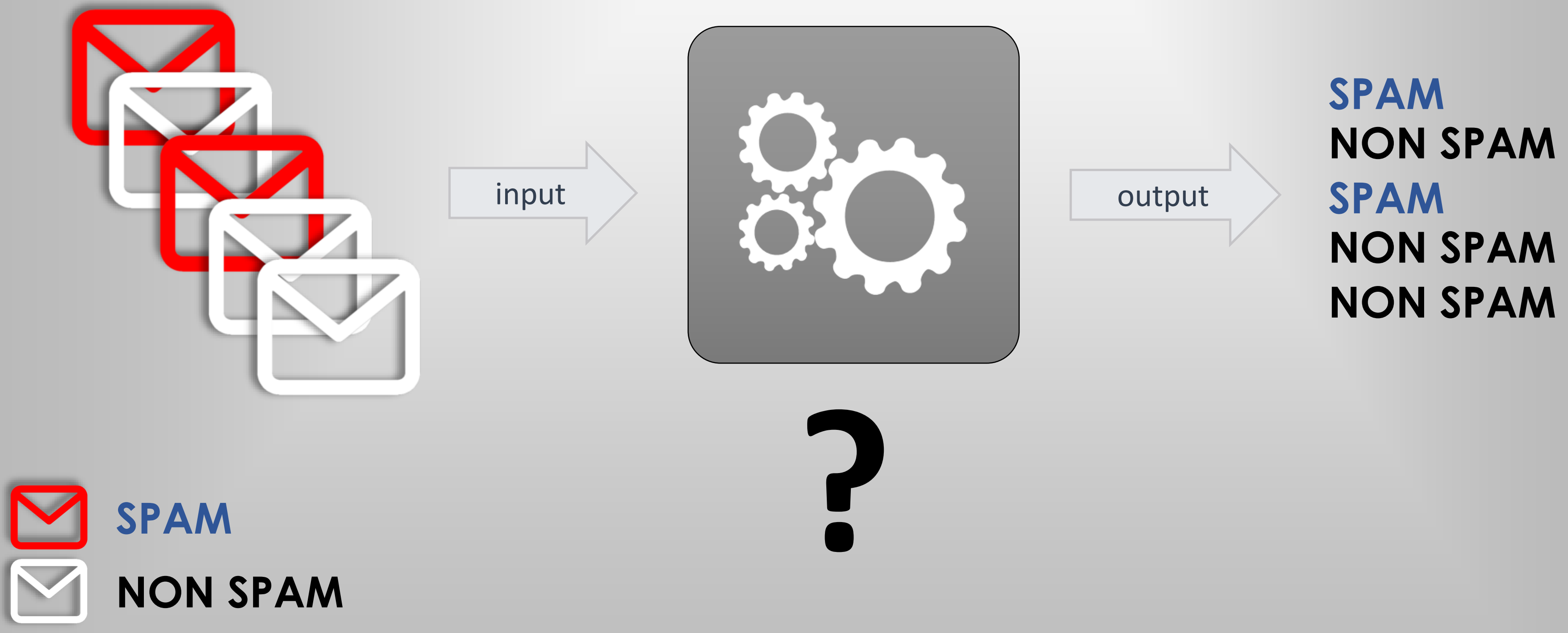
An example of an algorithm: a recipe

« A procedure that solves a given problem through a finite number of elementary step »

Wikipedia



Which algorithm?



Why not an algorithmic approach?

When one or more of the following happens:

- ✓Exact problem characterization/formalization is impossible (input->output)
- ✓Noise and/or uncertainty in input/output
- ✓High complexity of solution formulation
- ✓Inefficiency of the solution
- ✓Lack of 'compiled' knowledge about the problem to solve

When is learning important?

When the system has:

- ✓To adapt to the environment in which it operates (e.g. automatic personalizzation)
- ✓To improve its own performance with respect to a particular task
- ✓To discover regularities and new information (knowledge) from empirical data

Data Vs. Knowledge

In ML, we study methods to transform empirical information, present in the data, into new knowledge

Thanks to the evolution of computers and networks, data is now a day present everywhere, and abundant!

- ✓Receipts from a supermarket chain,
- ✓Web pages content,
- ✓E-commerce
- ✓Bank transactions,
- ✓Social networks, etc.

The Foundamental Assumption

There exists a (stochastic) process that explains the observed data. Maybe we don't know the details, but it exists!

E.g. social behaviors are not purely random

The aim of Machine Learning is to build a good (or better to say, useful) approximation of this process.

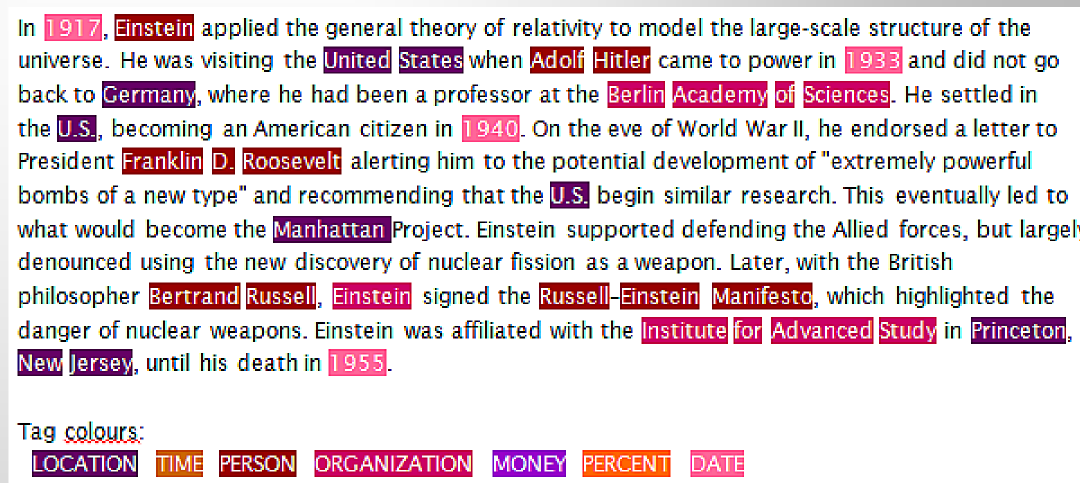
The main goal of ML

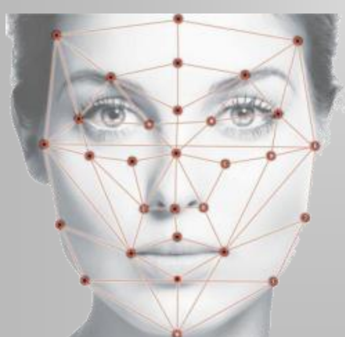
The final goal of ML is to define performance criteria and optimize them using data or previous experience

The parameters of the models will be learned on the basis of what we want to learn optimizing a given criterion. Using data and, possibly, prior knowledge about the domain

- ✓Predictive Models (predictions about the future)
- ✓Descriptive Models (obtaining new knowledge)

Application examples (classification)





Application examples (clustering and RL)

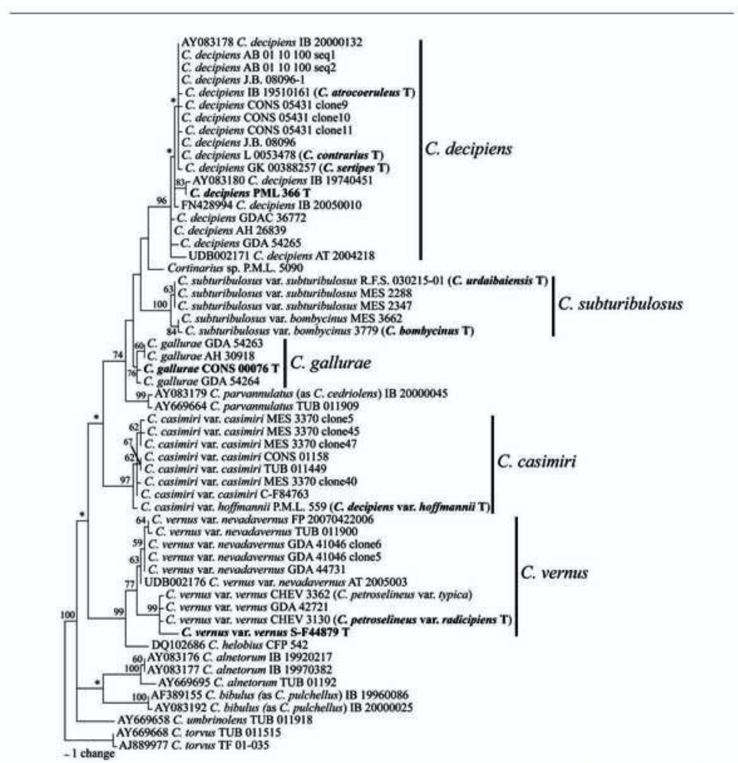
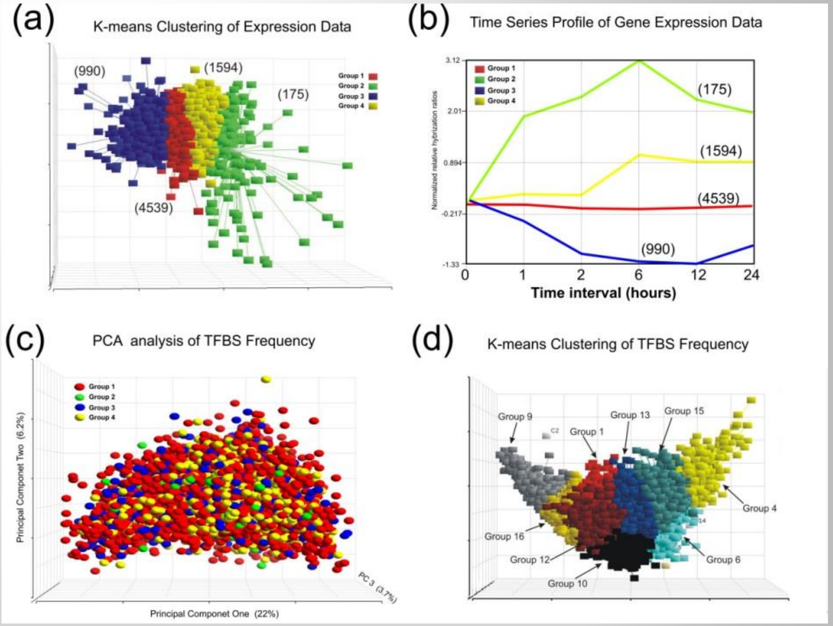
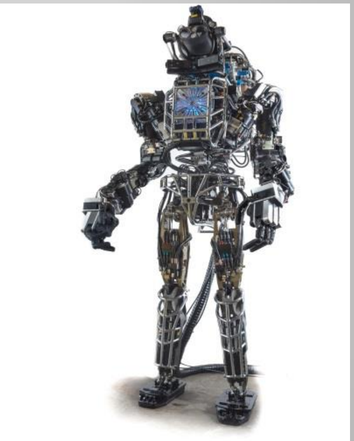
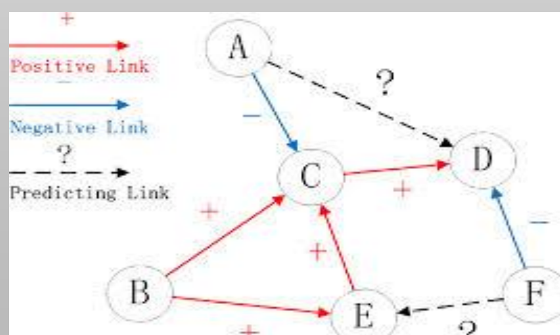


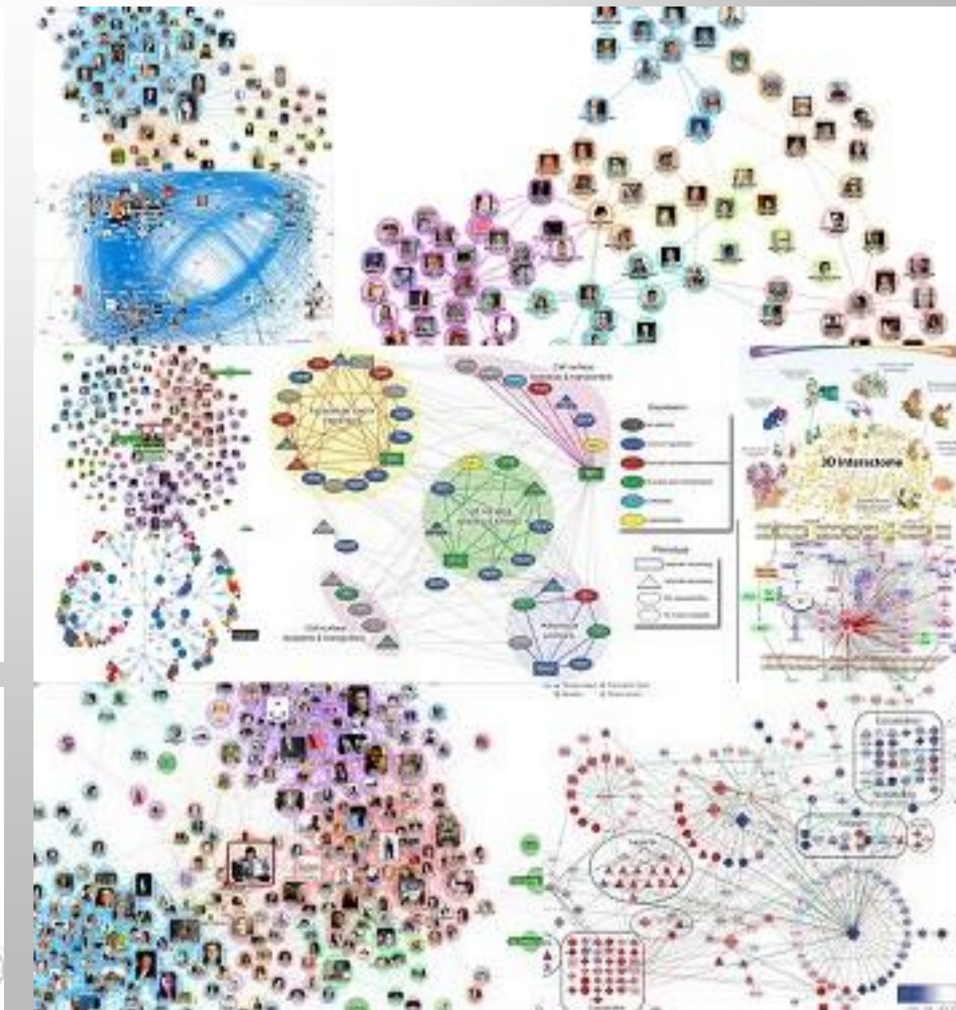
Fig 1 – Phylogenetic relationships of species of *Cortinarius* sect. *Hydrocybe* (subg. *Telamonia*) sensu Brandrud et al. (1992) and Bidaud et al. (1994). Phylogram of one of the most parsimonious trees (length: 173; CI: 0.775; RI: 0.937) obtained from the parsimony analysis. Bootstrap values ≥ 50 % are shown above branches. Branches collapsing in the strict consensus tree are marked with an asterisk. The herbarium references (for all sequences) and accession number (for the sequences taken from GenBank database and which specimens were not included in the morphological analysis) are shown after and before each taxon name, respectively. The bolded names followed by 'T' represent sequences obtained from type specimens. Species studied are indicated at right.





Application examples (RecSys and Link Prediction)





Application examples (1)

- Face Recognition
  - Access control from video recordings and photos. What are the features really important in a face?
- Named Entity Recognition
  - The problem of identifying entities in a sentence: places, titles, names, actions, etc. starting from a set of documents already labeled/tagged
- Document Classification
  - The problem to decide if an email is spam or not, or to assign a classification a document from a set of topics (sport, politics, hobby, art, gossip, etc.), possibly hearchically organized

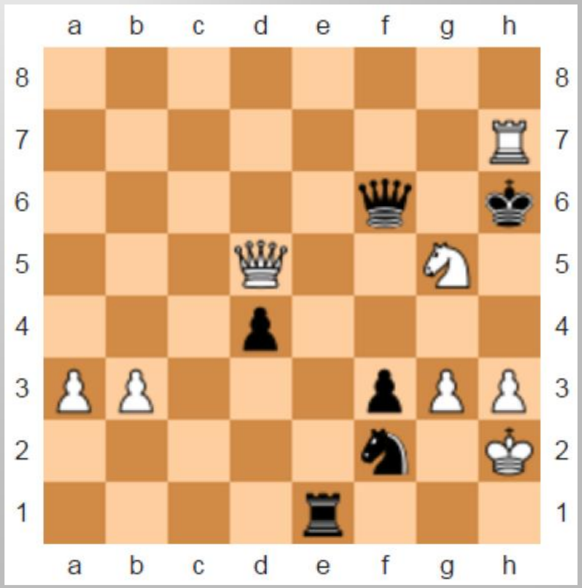
Application examples (2)

- Games and Opponent Profiling
  - For some games with incomplete information(card games, geister, risk, …) we want to predict missing information based on the strategies the opponent used in the past (threats, reactions, etc.).
- Bioinformatics
  - Microarrays are devices able to detect gene expressions from biological tissues. One possible task here is to determine how likely is that a patient react positively to a given terapy
- Recommender Systems
  - Recommender systems are able to suggest items of interest to a user, e.g. products (Amazon), songs (Spotify), new contacts (Facebook), etc.
- Speech Recognition, Handwritten Recognition, Social Network Analysis, and much more.

Deep Blue vs Garry Kasparov

- 1997 Deep Blue (IBM) won a chess match against the world champion Garry Kasparov
- Hardware: able to evaluate 200M board positions per second. Its computational power (11.38 GFLOPs) relevant at the time is less than that of a modern smartphone
- Deep search(6-8 levels on average)
- Score evaluation was very complex and had many parameters. The optimal values for these parameters were learned, analyzing thousands of matches from champions
- A list of openings were provided by chess champions

1996 – First game lost: Kasparov (black) withdraws

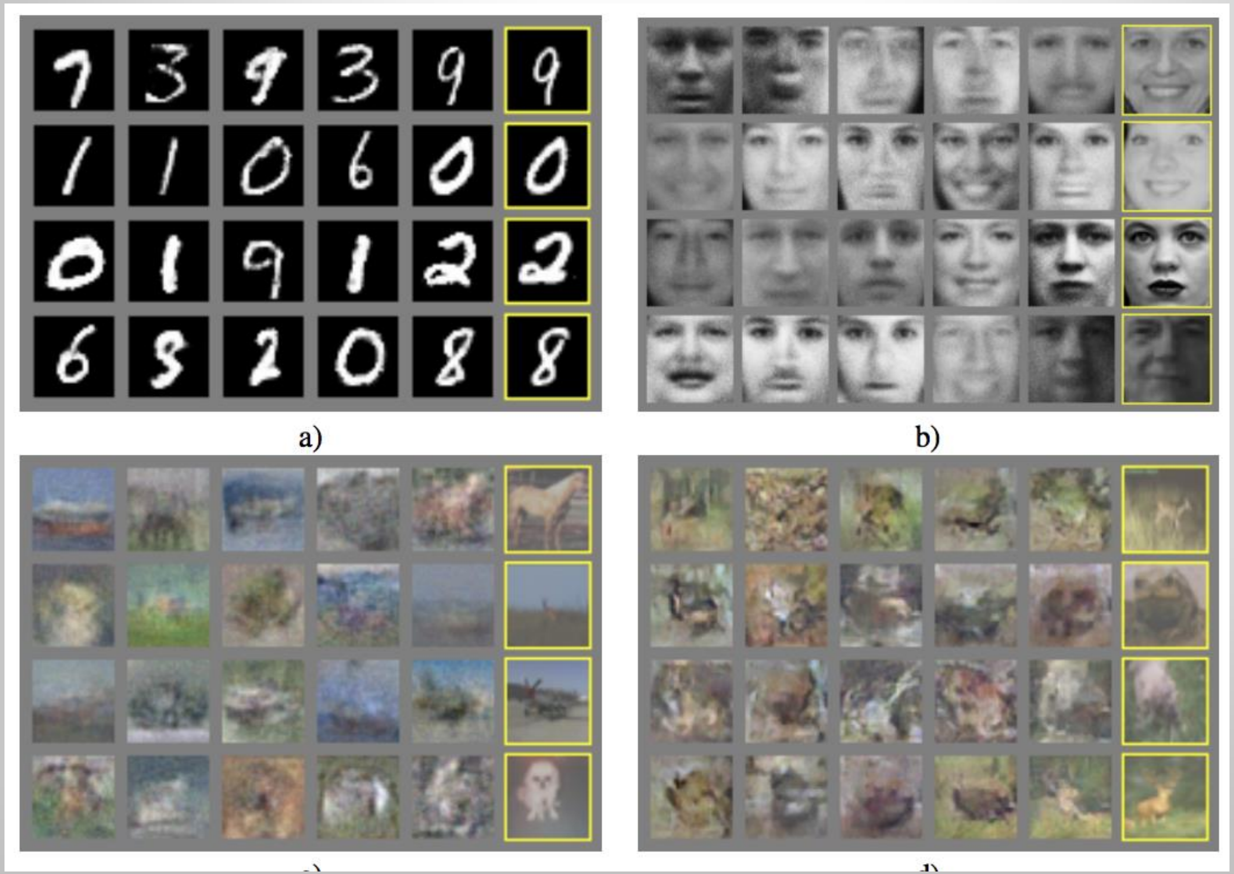


Google DeepMind vs Lee Sedol

- 2016 AlphaGo (Google) won against champion Lee Sedol at GO
- GO ancient chinese game, simple rules but many more moves compared to chess. Even more difficult to apply brute force algorithms.
- AlphaGo didn't use heuristic methods. It was heavily based on ML
- Supervised DeepNN to imitate professional moves starting from matches collected on Go Servers (30M moves)
- + Self training playing millions matches against itself (Reinforcement Learning)

2016 – AlphaGO vs Lee Sedol (4-1)

Generative Adversarial Learning (1)

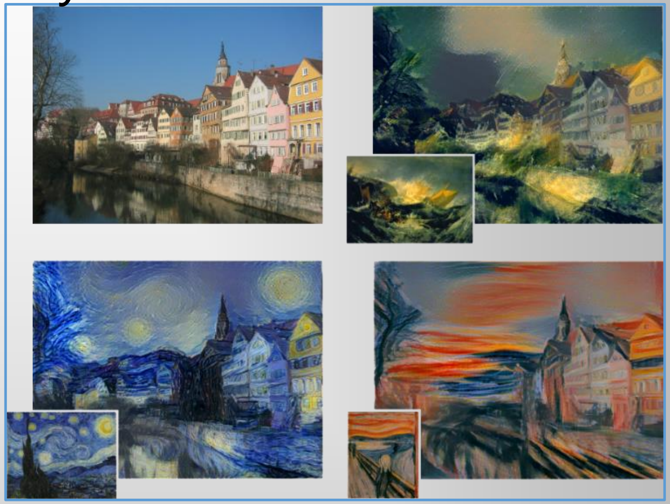


(a) MNIST handwritten digit dataset, (b) Toronto Face Database, (c-d) CIFAR-10 small object photograph dataset [Goodfellow et al. 2014]

Generative Adversarial Learning (2)

Fake video

Style transfer



Text-to-image

Fake faces

Other resources about applications of ML in TED talk…

The wonderful and terryfying implications of computers that can learn

[Jeremy Howard]

How we’re teaching computers to understand pictures

[Fei Fei Li]

Course Content

1) Introduction to SL
2) Theoretical Foundation: PAC and SRM
3) Decision Trees
4) Neural Networks
5) GLM and SVM
6) Preprocessing and Feature Selection
7) Model Selection and Evaluation
8) Representation (kernels, embeddings, CNNs)
9) Bayesian methods
10) Ensembles
11) Clustering
12) Recommender Systems
13) Seminars about application of ML