Computational Statistics

Luca Bortolussi

Department of Mathematics and Geosciences
University of Trieste

Office 303, third floor, E1 3
luca@dmi.units.it

Trieste, Winter Semester 2016/2017

SOME FACTS WORTH CONSIDERING

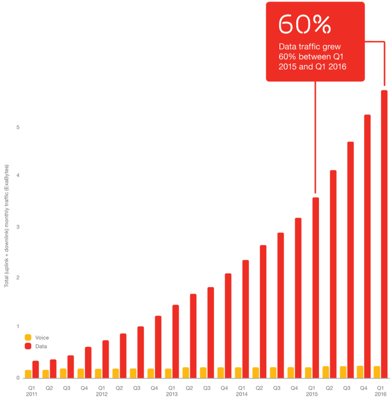
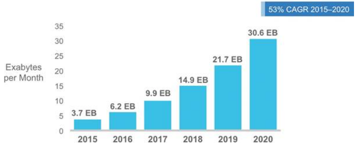


Table 1. The Cisco VNI Forecast—Historical Internet Context

Year	Global Internet Traffic
1992	100 GB per day
1997	100 GB per hour
2002	100 GBps
2007	2000 GBps
2014	16,144 GBps
2019	51,794GBps

Source: Cisco VNI, 2015

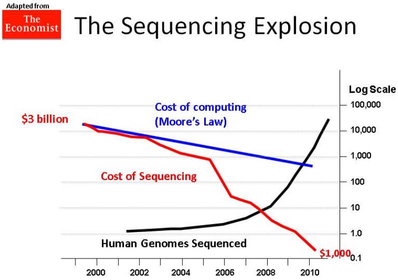


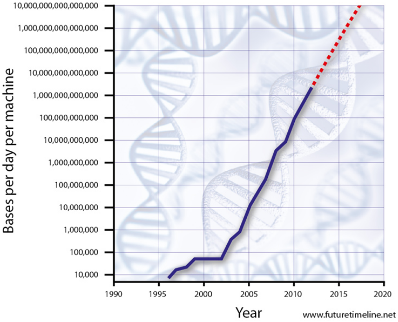
Mobile traffic in 2013 = 18 × total internet traffic in 2000

Mobile traffic in 2021 = 12 × traffic in 2015.

We are living in a world pervaded by data (information?)

SOME FACTS WORTH CONSIDERING

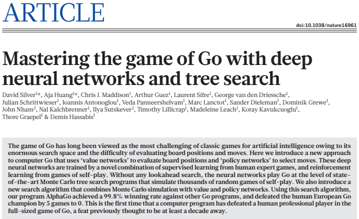




UK National Health Service plans to sequence genome of 750.000 cancer patients in the next ten years

How to make sense of all this data?
How to extract knowledge from it?

SOME FACTS WORTH CONSIDERING



ARTICLE

doi:10.1038/nature16961

Mastering the game of Go with deep neural networks and tree search

David Silver[1*], Aja Huang[1*], Chris J. Maddison[1], Arthur Guez[1], Laurent Sifre[1], George van den Driessche[1], Julian Schrittwieser[1], Ioannis Antonoglou[1], Veda Panneershelvam[1], Marc Lanctot[1], Sander Dieleman[1], Dominik Grewe[1], John Nham[2], Nal Kalchbrenner[1], Ilya Sutskever[2], Timothy Lillicrap[1], Madeleine Leach[1], Koray Kavukcuoglu[1], Thore Graepel[1] & Demis Hassabis[1]

The game of Go has long been viewed as the most challenging of classic games for artificial intelligence owing to its enormous search space and the difficulty of evaluating board positions and moves. Here we introduce a new approach to computer Go that uses 'value networks' to evaluate board positions and 'policy networks' to select moves. These deep neural networks are trained by a novel combination of supervised learning from human expert games, and reinforcement learning from games of self-play. Without any lookahead search, the neural networks play Go at the level of state-of-the-art Monte Carlo tree search programs that simulate thousands of random games of self-play. We also introduce a new search algorithm that combines Monte Carlo simulation with value and policy networks. Using this search algorithm, our program AlphaGo achieved a 99.8% winning rate against other Go programs, and defeated the human European Go champion by 5 games to 0. This is the first time that a computer program has defeated a human professional player in the full-sized game of Go, a feat previously thought to be at least a decade away.

Google purchased DeepMind (after 1 year of operation) for 450M GBP

Surprised? Google's business is based on analysing immense quantities of data...

Some facts worth considering

Data Science, as a term, "was first coined in 2001. Its popularity has exploded since 2010, pushed by the need for teams of people to analyze the big data that corporations and governments are collecting." (Wikibook on data science)

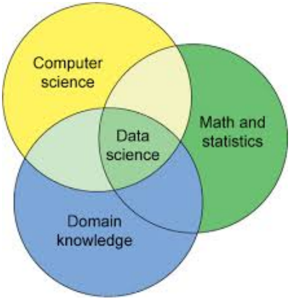
Number of job postings for data scientists increased globally by 20.000% between 2009 and 2015...
73% growth of job offers in data science in Italy, from jan-mar 2015 to jan-mar 2016.

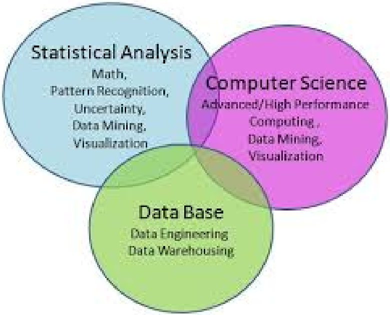
THE PROBLEM (BIG DATA)

- Vast amounts of quantitative data arising from every aspect of life, due to technological advances.
- Advanced informatics tools necessary just to handle the data (data storage, transmission, querying - cloud computing, data centers)
- Widespread belief that data is valuable, yet worthless without analytic tools
- Converting data to knowledge is the challenge. This is where computational statistics comes into play.
- 4 v for big data: volume (of data - Petabytes like nuts), velocity (of data acquisition - Twitter collects 12 Terabytes per day), variety (of data sources), value (in the datasets, but need to extract it).

Data Science

Data Science is an interdisciplinary field about processes and systems to extract knowledge or insights from large volumes of data in various forms, either structured or unstructured... [wikipedia]



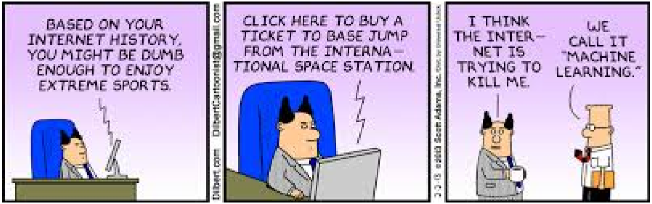


Computational statistics lies in between statistics and computer science. It is more often known as machine learning. Advances in this field are at the core of the successes of data science.

Machine Learning

If you google it…

Machine learning is a subfield of computer science that evolved from the study of pattern recognition and computational learning theory in artificial intelligence. Machine learning explores the study and construction of algorithms that can learn from and make predictions on data. [source: wikipedia]



A rough classification

> Machine learning explores the study and construction of algorithms that can learn from and make predictions on data. [source: wikipedia]

- Supervised learning: learn a model from input-output data. The goal is to predict a the (most-likely) output value for a new, unobserved, input. We distinguish
  - Regression (continuous output)
  - Classification (binary/ discrete output)
- Unsupervised learning: extract information/ learn a model from input-only data
- Reinforcement Learning: find suitable actions to take in a given situation in order to maximize a reward.

It's all about the models

- Machine Learning is all about learning models...
- But, what is a model? Discuss for 5 minutes and provide 3 examples

My own answer

- A model is a hypothesis that certain features of a system of interest are well replicated in another, simpler system.

My own answer

- A model is a hypothesis that certain features of a system of interest are well replicated in another, simpler system.
- A *mathematical model* is a model where the simpler system consists of a set of mathematical relations between objects (equations, inequalities, etc).

My own answer

- A model is a hypothesis that certain features of a system of interest are well replicated in another, simpler system.
- A *mathematical model* is a model where the simpler system consists of a set of mathematical relations between objects (equations, inequalities, etc).
- A *stochastic model* is a mathematical model where the objects are probability distributions.

My own answer

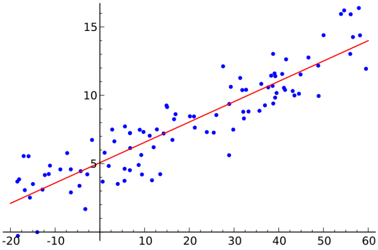
- A model is a hypothesis that certain features of a system of interest are well replicated in another, simpler system.
- A *mathematical model* is a model where the simpler system consists of a set of mathematical relations between objects (equations, inequalities, etc).
- A *stochastic model* is a mathematical model where the objects are probability distributions.
- All modelling usually starts by defining a *family* of models indexed by some parameters, which are tweaked to reflect how well the feature of interest is captured.

My own answer

- A model is a hypothesis that certain features of a system of interest are well replicated in another, simpler system.
- A *mathematical model* is a model where the simpler system consists of a set of mathematical relations between objects (equations, inequalities, etc).
- A *stochastic model* is a mathematical model where the objects are probability distributions.
- All modelling usually starts by defining a *family* of models indexed by some parameters, which are tweaked to reflect how well the feature of interest is captured.
- Machine learning deals with algorithms for automatic selection of a model from observations of the system.

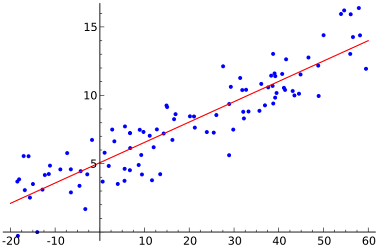
Supervised Learning - regression

Regression: The computer is presented with example inputs and their observed outputs, both continuous, and the goal is to learn a general rule that maps inputs to outputs.



SUPERVISED LEARNING - REGRESSION

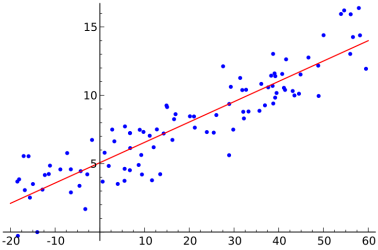
Regression: The computer is presented with example inputs and their observed outputs, both continuous, and the goal is to learn a general rule that maps inputs to outputs.



- we observe input-output data: $\mathbf{x}_1 – y_1, \ldots \mathbf{x}_n – y_n$ ($\mathbf{x}_i \in \mathbb{R}^n$, $y_i \in \mathbb{R}$, input $\mathbf{X}$, output $\mathbf{y}$).

SUPERVISED LEARNING - REGRESSION

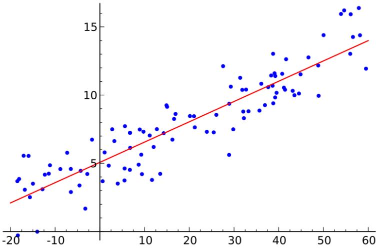
Regression: The computer is presented with example inputs and their observed outputs, both continuous, and the goal is to learn a general rule that maps inputs to outputs.



- we observe input-output data: $\mathbf{x}_1$–$y_1, \ldots \mathbf{x}_n$–y_n ($\mathbf{x}_i \in \mathbb{R}^n$, $y_i \in \mathbb{R}$, input $\mathbf{X}$, output $\mathbf{y}$).
- We assume this data is generated by $y = f(\mathbf{x}) + \epsilon$, ϵ noise.
- We want to learn f (a good approximation of it) from $\mathbf{X}$,$\mathbf{y}$.

Supervised Learning - regression

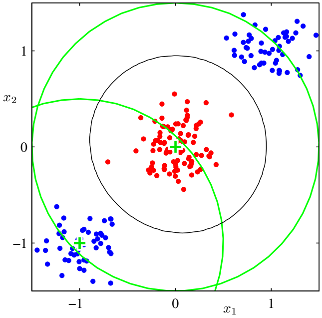
Regression: The computer is presented with example inputs and their observed outputs, both continuous, and the goal is to learn a general rule that maps inputs to outputs.



- we observe input-output data: $\mathbf{x}_1$–$y_1, \ldots \mathbf{x}_n$–y_n ($\mathbf{x}_i \in \mathbb{R}^n$, $y_i \in \mathbb{R}$, input $\mathbf{X}$, output $\mathbf{y}$).
- We assume this data is generated by $y = f(\mathbf{x}) + \epsilon$, ϵ noise.
- We want to learn f (a good approximation of it) from $\mathbf{X}$,$\mathbf{y}$.
- Alternatively, data comes from a probabilistic model $p(y, \mathbf{x})$.
- We want to learn (an approximation of) p.

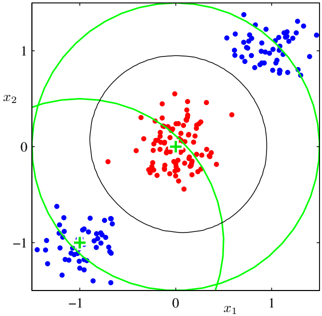
SUPERVISED LEARNING - CLASSIFICATION

Classification: The computer is presented with example inputs from different classes (binary/ discrete), and the goal is to learn a general rule that assigns inputs to a class.



Supervised learning - classification

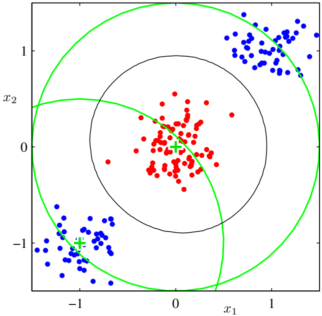
Classification: The computer is presented with example inputs from different classes (binary/ discrete), and the goal is to learn a general rule that assigns inputs to a class.



- we observe input-output data: $\mathbf{x}_1$–y_1, $\ldots$ $\mathbf{x}_n$–y_n ($\mathbf{x}_i \in \mathbb{R}^n$, $y_i \in \{0, \ldots, n\}$, input $\mathbf{X}$, output $\mathbf{y}$).

SUPERVISED LEARNING - CLASSIFICATION

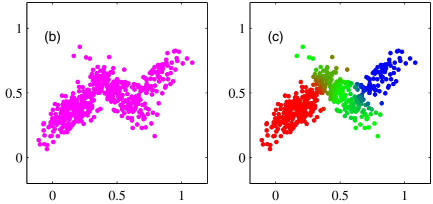
Classification: The computer is presented with example inputs from different classes (binary/ discrete), and the goal is to learn a general rule that assigns inputs to a class.



- we observe input-output data: $\mathbf{x}_1$–y_1, $\ldots \mathbf{x}_n$–y_n ($\mathbf{x}_i \in \mathbb{R}^n$, $y_i \in \{0, \ldots, n\}$, input $\mathbf{X}$, output $\mathbf{y}$).
- We want to learn a rule $y = f(\mathbf{x})$ assigning each $\mathbf{x}$ to a class.
- Alternatively, data comes from a probabilistic model $p(y, \mathbf{x})$, and we want to learn it as accurately as possible.

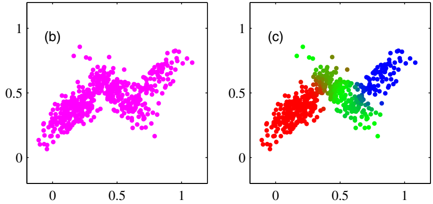
UNSUPERVISED LEARNING

Unsupervised learning: No labels are given to the learning algorithm (input only), leaving it on its own to find structure in its input.



UNSUPERVISED LEARNING

Unsupervised learning: No labels are given to the learning algorithm (input only), leaving it on its own to find structure in its input.



- Clustering: discover groups of similar examples within the data.
- Density estimation: determine the distribution of data within the input space.
- Dimensionality reduction: project the data from a high-dimensional space to a lower dimension space. Often down to two or three dimensions for the purpose of visualization.

Reinforcement learning

- Reinforcement learning is concerned with the problem of finding suitable actions to take in a given situation in order to maximize a reward. Typically there is a sequence of states and actions in which the learning algorithm is interacting with its environment. Here the learning algorithm is not given examples of optimal outputs, in contrast to supervised learning, but must instead discover them by a process of trial and error.

Reinforcement learning

- Reinforcement learning is concerned with the problem of finding suitable actions to take in a given situation in order to maximize a reward. Typically there is a sequence of states and actions in which the learning algorithm is interacting with its environment. Here the learning algorithm is not given examples of optimal outputs, in contrast to supervised learning, but must instead discover them by a process of trial and error.
- An example is an algorithm learning to play the game of backgammon to a high standard. Here the algorithm must learn to take a board position as input, along with the result of a dice throw, and produce a strong move as the output.

Reinforcement learning

- Reinforcement learning is concerned with the problem of finding suitable actions to take in a given situation in order to maximize a reward. Typically there is a sequence of states and actions in which the learning algorithm is interacting with its environment. Here the learning algorithm is not given examples of optimal outputs, in contrast to supervised learning, but must instead discover them by a process of trial and error.
- An example is an algorithm learning to play the game of backgammon to a high standard. Here the algorithm must learn to take a board position as input, along with the result of a dice throw, and produce a strong move as the output.
- A general feature of reinforcement learning is the trade-off between exploration, in which the system tries out new kinds of actions to see how effective they are, and exploitation, in which the system makes use of actions that are known to yield a high reward. Too strong a focus on either exploration or exploitation will yield poor results.

Generative and Discriminative models

- Supervised learning can have two flavours
- Two different types of question can be asked:
  - what is the joint probability of input/ output pairs?
  - given a new input, what will be the output?
- The first question requires a model of the population structure of the inputs, and of the conditional probability of the output given the input → **generative modelling**
- The second question is more parsimonious but less explanatory → **discriminative learning**
- Notice that the difference between generative supervised learning and unsupervised learning is moot

Course plan

- 4 hours to “refresh” some basic notions of probability and statistics (this week, hopefully).
- Six blocks (one per week) of four hours of theory and three hours of hands-on laboratory. Blocks will roughly be on the following topics:

COURSE PLAN

- 4 hours to "refresh" some basic notions of probability and statistics (this week, hopefully).
- Six blocks (one per week) of four hours of theory and three hours of hands-on laboratory. Blocks will roughly be on the following topics:
  1. (Bayesian) Linear Regression
  2. Linear Classification (and maybe some notions of Sparse Vector Machines)
  3. Gaussian Processes for Regression and Classification
  4. Unsupervised learning: clustering, nearest neighbour and kernel density estimation, Principal Component Analysis.
  5. Mixtures of Gaussians and Expectation Maximisation

  ? If we have time: Graphical Models, message passing and inference, Neural Networks, Hidden Markov Models; Active Learning and Bayesian optimisation.

Lab+Exam

Laboratory

The course will have theoretical lessons and Laboratory ones, in which we will implement and test different methods on sample data.

Bring your own laptop... (???)

Exam

- Final team project, with presentation.
- A report on lab work (conditional on you not showing me the results during the lab).
- In case, some questions about what we learned (to check you knew what you did in the project and lab!)

Coordinates

Moodle

There is a moodle page of the course. Register, it is where you will get all the material.

Coordinates

Moodle

There is a moodle page of the course. Register, it is where you will get all the material.

Where can you find me?

Coordinates

Moodle

There is a moodle page of the course. Register, it is where you will get all the material.

Where can you find me?

- Around the World.

Coordinates

Moodle

There is a moodle page of the course. Register, it is where you will get all the material.

Where can you find me?

- Around the World.
- Room 328, 3rd floor - email me first at `lbortolussi@units.it`.

Coordinates

Moodle

There is a moodle page of the course. Register, it is where you will get all the material.

Where can you find me?

- Around the World.
- Room 328, 3rd floor - email me first at `lbortolussi@units.it`.

Other stuff

- question time at the end of each lecture
- Requests?

Timetable

Forget the current one. We need to allocate preferrably 7/8 hours per week (2 theoretical lessons x 2 hours, 1 lab x 3 hours or 2 labs x 4 hours).