Introduction to *In-Silico* Learning

Machine Learning, Data Mining, and Search Problems

Giacomo Bergami

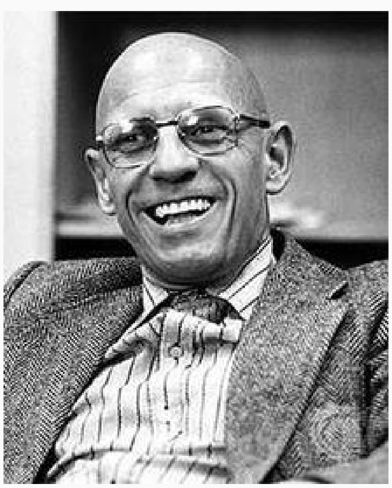
Why "*In-Silico* Learning"?

Mind over Machine (1986)

Hubert Dreyfus (1929-2017)

- Skill acquisition requires both holistic and situational intelligence:
  - *Holistic*: The components as a whole posses more features than each single component by itself.
  - *Situational*: Agent-World interaction
- Machines cannot reason using common sense knowledge: `https://youtu.be/SUZUbYCBtGI`
  - The AI community's response was derisive and personal: his MIT colleagues working in AI "*dared not be seen having lunch with me*".
- Daniel Crevier: "*time has proven the accuracy and perceptiveness of some of Dreyfus's comments.*"

The Chomsky-Foucault Debate: On Human Nature (1971)



Michel Foucault (1926 – 1984)

"Therefore a science, the advancement of science, and the acquisition of science, is not simply the oblivion of old [scientific] prejudices, or the fall of certain obstacles [to understanding], it is a new grid [of concepts] that masks certain things while allowing for the appearance of new knowledge".

`https://youtu.be/7TUD4gfvtDY`

Introduction

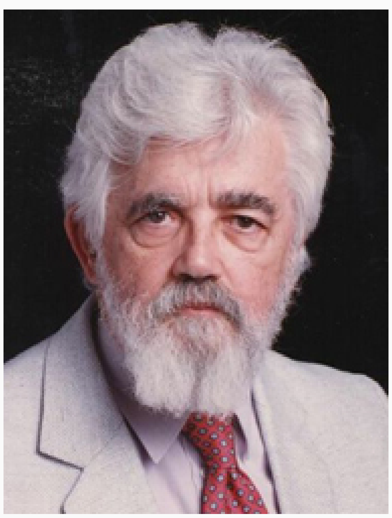
Semantic Information Processing (1968)

Marvin Minsky (1927 – 2016)

- **Artificial Intelligence** is the *"the science of making machines do things that would require intelligence if done by men"*.
- It is considered as the "*quadruple point*" between logic, computer science, neurosciences and philosophy.

Logic

Why Logic is relevant for "*In-Silico* Reasoning"?



John McCarthy (1927 - 2011)

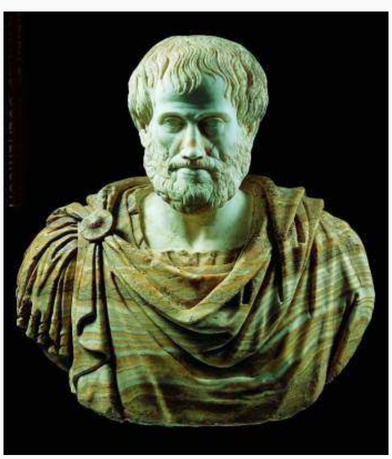
- McCarthy had a deep knowledge of logic in relation to Artificial Intelligence.
- He is one of the inventors of LISP.
- He introduced procedural inclusion of knowledge using abstracted reasoning plans.
- The resulting controversy is still relevant and being investigated.

Summa Logicæ (1323)

William Ockham (1288 - 1347)

"Logic is the most useful tool of all the arts. Without it no science can be fully known. It is not worn out by repeated use, after the manner of material tools, but rather admits of continual growth through the diligent exercise of any other science. For just as a mechanic who lacks a complete knowledge of his tool gains a fuller [knowledge] by using it, so one who is educated in the firm principles of logic, while he painstakingly devotes his labor to the other sciences, acquires at the same time a greater skill at this art.".

Categories and *On Interpretation* (~300 BC)



Aristotélēs (384 - 322 BC)

Within the bounds of Western Philosophy, Aristoteles is considered the father of modern logic, even though with some flaws.

- *Categories*: he introduces ontology reasoning for classifying and describing reality.
- *On Interpretation*: introduces the relation between logic (reasoning) and language.
  - Shows the basic steps of reasoning: given some premises (hypothesis) you can infer some consequence (thesis).
  - Introduces quantifications ($\forall, \exists \ldots$)

Elements ($\Sigma\tau o\iota\xi\varepsilon\iota\alpha$ - ~300 BC)

Eukleídēs of Alexandria

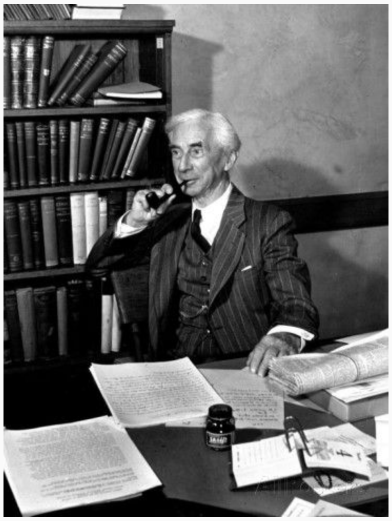
- They contain a first formulation of what is now known today as *Euclidean geometry*.
- *Elements* is composed of different books. Each book contains different definitions and propositions. Propositions are proved via demonstrations.
- Euclid's formulation is still being taught in secondary schools to provide a first example of an axiomatic system of rigorous demonstration.

Ethica ordine geometrico demonstrata (Posth. 1677)

Baruch Spinoza (1632 - 1677)

- The book is perhaps the most ambitious attempt to apply the method of Euclid in philosophy.
- Spinoza puts forward a small number of definitions and axioms from which he attempts to derive hundreds of propositions and corollaries.

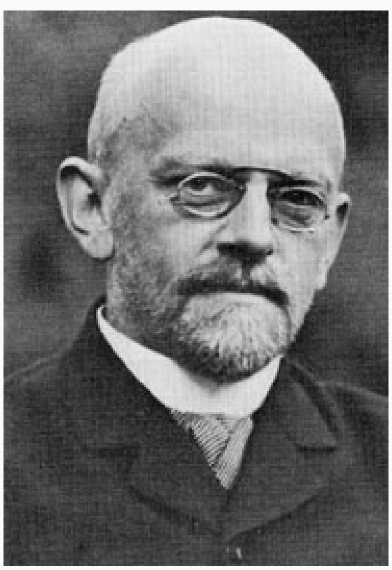
Principia Mathematica (1910-1913)



Bertrand A. W. Russell (1872 - 1970)

- Everything could be decomposed into logical formulæ.
- The *Principia Mathematica* represent an important attempt to build mathematics from a defined set of axioms and logical rules.
- The *Principia*, however, still contained some contradictions.
- This work influenced two German Mathematicians, David Hilbert and Kurt Gödel.

Mathematische Probleme (1902)



David Hilbert (1862 - 1943)

Any logical system at the basis of *Principia Mathematica* must satisfy the following properties:

- Completeness: all true mathematical statements can be proved within the formalism.
- Consistency: no contradiction can be obtained.
- Decidability: it exists algorithm for deciding the truth or falsity of any mathematical statement (*Entscheidungsproblem*).

Über formal unentscheidbare Sätze der Principia Mathematica *und verwandter Systeme* (1931)

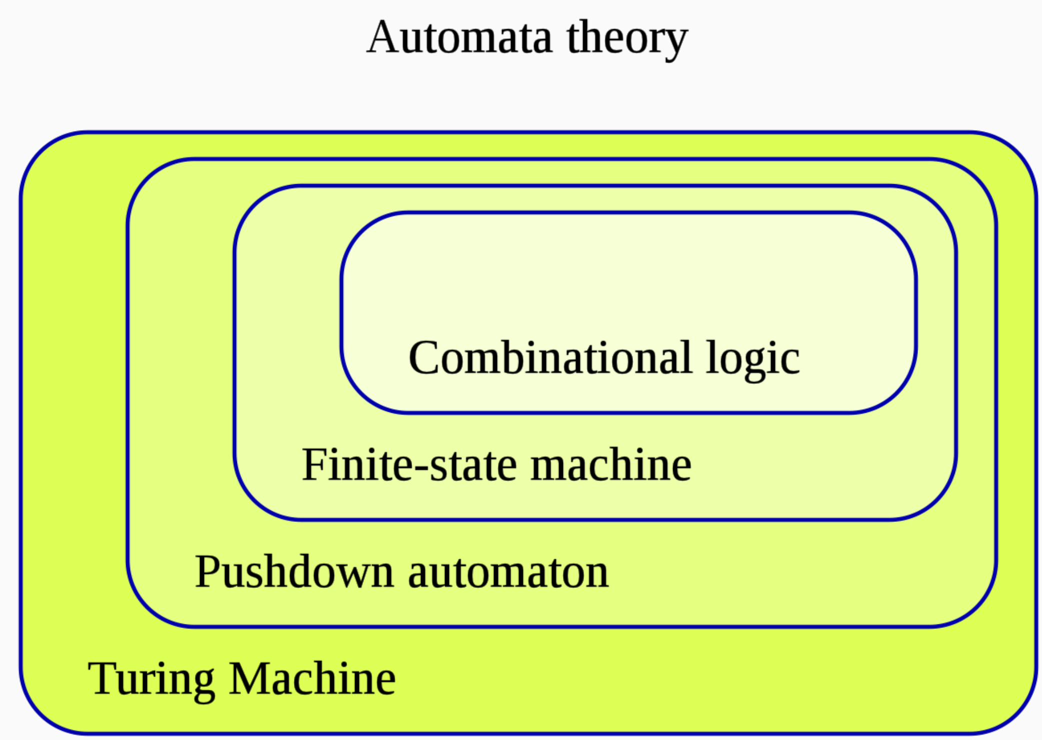
Kurt Gödel (1906 - 1978)

- In Mathematics a *Theory of everything* is not possible.
- Gödel's work is generally taken to show that Hilbert's Program cannot be carried out: no (complex) mathematical system can be both complete and consistent:
  - You cannot prove a system's consistency within the system itself.
  - Some theorems cannot be showed to be true using an algorithm/proof.
- See also "*Gödel's Proof*" by Ernest Nagel & James R. Newman (1957)

"On Computable Numbers, with an Application to the Entscheidungsproblem" (1936)

Alan M. Turing (1912 - 1954)

- It cannot exist a single algorithm that can decide each single theorem: you need to check an infinite number of assumptions:
  - "*what I shall prove is quite different from the well-known results of Gödel … I shall now show that there is no general method which tells whether a given formula U is provable in K [Principia Mathematica]*"
- In order to do so, he defined the *Turing Machine.*



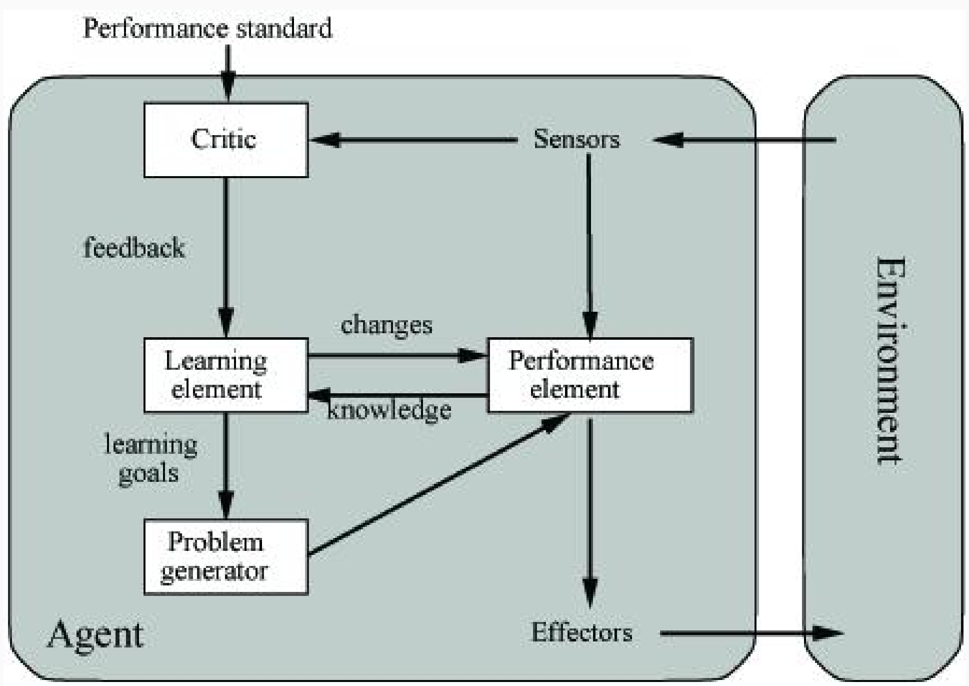
Artificial Intelligence

Sein und Zeit (1927)

Martin Heidegger (1889 - 1976)

- In his work, Heidegger showed that:
  - the human mind is more general than any mechanistic simplification
  - the human knowledge comes from the interaction with the world, from which he gains the common sense knowledge.
- His studies influenced Hubert Dreyfus and his 1972 *"What computers can't do: A critique of artificial reason"*.
  - The challenge to a different approach to symbolic reasoning was taken up recently:
    - Carlos Herrera and Ricardo Sanz: "*Heideggerian AI and the Being of Robots.*" . . . 2016.

A general learning agent (1/2)



from "Artificial Intelligence: A Modern Approach", by Russel & Norvig

A general learning agent (2/2)

- This model allows to interact in worlds where the initial state is unknown.
  - The agent shall increase its knowledge with experience (learning element).
  - As a response to the outside knowledge, the agent might decide to perform some actions (performance element).
  - After changing the world, the agent is observing the consequences of his actions, and he's considering whether it should change its plan of action (Critic).
- Some examples of agent/world interaction models:
  - Rule Based: http://jason.sourceforge.net/
  - Ising Model based: https://github.com/jackbergus/socialsim

Computing Machinery and Intelligence (1950)

Alan M. Turing (1912 - 1954)

- *"It is not possible to produce a set of rules purporting to describe what a man should do in every conceivable set of circumstances."*
- This implies the need to define a system that can dynamically learn from (positive) examples.
- It should also be able to find the correct solution and ignore the background noise.
- Nevertheless, he thinks that *weak AI* can be achieved.

Minds, Machines and Gödel (1961)

John R. Lucas (1929)

- *"Gödel's Theorem seems to me to prove that Mechanism is false, that is, that minds cannot be explained as machines"*
- Humans can ascertain propositions that are non-decidable within a mathematical theory because they can be more general than the used/designed system.

Minds, Brains and Programs (1980)

John Searle (1932)

We can highlight two cornerstones:

- Machines do inference via *symbol pushing*: this mechanism does not necessairly imply that the machine knows what those symbols mean (confutation to the Chinese room argument). Therefore, inelligence implies intentionality.
- (As also recently showed) it is the programmer coding the machine and not the machine itself that can be considered *intelligent*.

Le champ du signe ou la faillite du réductionnisme (1989)

Jean-Yves Girard (1947)

- *"It is not possible to build a machine that can solve any possible problem. [Albeit the human brain has some limitations], the human brain seem to posses a more complex set of operational rules"*
- *"In the real world, the logical principles are no longer about the truth but about the action: with 10 francs I can buy a pack of Camels, with 10 francs I can buy a package of Malboro, but not both."*
  - Do we have a logic which is different from Classical Logic?

A general Theory of In-Silico Reasoning

Why it is important to know your data before applying a ML model?

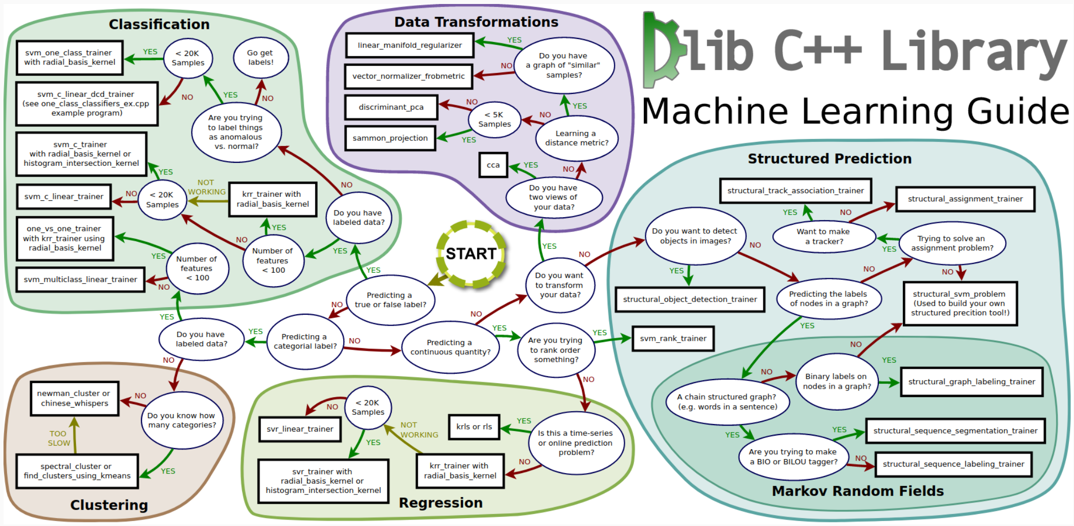
- Getting the best model for solving the best task.
- Knowing the expressive power of the model of interest.

Getting the best model for solving the best task (1/2)

The type of machine learning approach that you need to use depends on the answer to the following questions:

- Is the data structured or does it just represent data points?
- For some statistical techniques, which is the approximate data distribution?
- What is the operation that you need to perform?
- Does the data work with categorical/numerical data?
- How many categories/dimensions are there?
- Which is the size of the dataset?

Getting the best model for solving the best task (2/2)



Knowing the expressive power of the model of interest

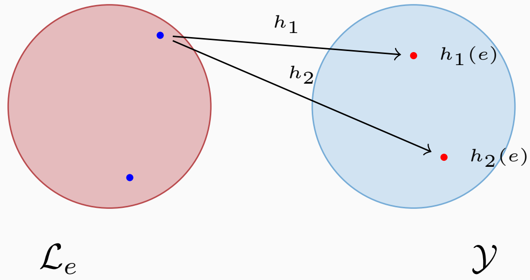
You still need to understand the data distribution that you're using for setting up the training model's configurations. With respect to a binary classification problem (**XOR**):

- Neural Network: you need to use multi-layered networks to learn complicated problems.
- One single SVM can learn more problematic functions via a specific *kernel trick*.

Nevertheless Neural Networks are more expressive than SVM:

- (Graph) Neural Networks can learn NP-Complete decision problems, thus providing a binary classification.
- In fact, we can prove that Neural Networks are Turing Complete.
- Still, they can learn only one function at the time, and cannot learn from biased data.

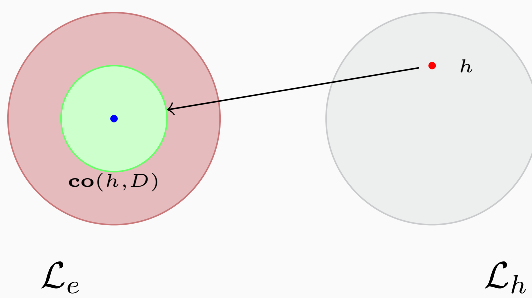
Modelling Learning



Hypotheses viewed as functions

- $\mathcal{Y}$: target classification classes.
  - $\mathcal{Y} = \{0, 1\}$: binary classification
  - $|\mathcal{Y}| > 2$: muticlass classification
- $\mathcal{L}_e$: training data.
- $h_1, h_2 \in \mathcal{L}_h$: learned hypotheses.
  - We want to learn a target function $f \in \mathcal{L}_h$ which is implicit from the data.

Binary Classification



The covers relation

- $\mathcal{L}_h$: hypotheses correlating the data to the classes.
- **co**: returns a subset of D for which $h(d) = 1$ with $d \in D$.

We can also define an approximated hypothesis $\tilde{h}$ such that $\tilde{h}(d) =\approx \mathbb{P}[f(d) = h(d)|h]$

Multiclass Classification

We can get multiclass classifications from binary classification algorithms:

- one-versus-all: for each class k, create k binary classifiers h_y that are deciding whether the object x belongs to the class y or not. Then, return the class minimizing the loss function:

$$h(x) = \arg\min_{y \in \mathcal{Y}} \tilde{h}_y(x)$$

- one-versus-one: for each class k, create $\frac{k(k-1)}{2}$ binary classifiers in C that classify x as belonging to one of two classes in $\mathcal{Y}$. Then, return the class winning the max-wins voting strategy:

$$h(x) = \arg\max_{y \in \mathcal{Y}} |\{ h_i \in C \mid h_i(x) = y \}|$$

Confusion Matrix for Binary Classification

Prediction outcome

actual value		**p**	**n**	**total**
	p′	True Positive	False Negative	P′
	n′	False Positive	True Negative	N′
	total	P	N	

Displaying the performance of a learning algorithm

- Precision or Positive Predicted Value: $P = \frac{|p \cap p'|}{|P|}$
- Recall or True Positive Rate: $R = \frac{|p \cap p'|}{|P'|}$

This result can be easily generalized for any *multiclass l. algorithm.*

Three Main Learning Models (Towards Explainability, 1/2)

- **Machine Learning**
  - Learn an unknown target function f from a set of examples $E = (e_i, f(e_i))_{i \leq n \in \mathbb{N}} \subseteq \mathcal{L}_e \times \mathcal{Y}$, so that we generate an hypothesis h minimizing the loss function with E (e.g., *distance between prediction and* $f(e_i)$).
  - E.g., Neural Networks, Support Vector Machines.
- **Data Mining**
  - Extends the previous approach by finding multiple possible hypotheses h satisfying a quality criterion $\mathcal{Q}$ (e.g., *precision*).
  - The poset $(\mathcal{L}_h, \preceq)$ represents all the possible solutions to the problem.
  - E.g., Rule Mining, Markov Logic Networks.
- **Search Problem**
  - Given an initial configuration and some quality criterion $\mathcal{Q}$ (e.g, heuristics), find the local maximum.
  - The set E represents all the possible reachable configurations from a given initial point.

Three Main Learning Models (Towards Explainability, 2/2)

All these three models can be summarized by this high level functions:

- **Machine Learning**:

$$ML(loss, E, \mathcal{L}_h) := \arg\min_{h' \in \mathcal{L}_h} loss(h', E)$$

- **Data Mining**:

$$DM(\mathcal{Q}, E, \mathcal{L}_h) := \{\, h \in \mathcal{L}_h \mid \mathcal{Q}(h, E) \text{ holds} \,\}$$

- **Search Problem** (already covered in the Game Tech module):

$$SP(\mathcal{Q}, E, s) = s \overset{h}{\rightsquigarrow} (\arg\min_{t \in E} \bigwedge_{(s' \to t') \in s \rightsquigarrow t} \mathcal{Q}(s' \to t', t))$$

Three Main Learning Models

Analysis \ Examples	**Neural Networks**	**Rel. Learning**	**Search Problem**
Input	Training Data (P/N)	Poset, (P/N)	Initial Conf.
Quality Requirements	Training Data	Quality Function	Heuristics, Expansion Rules
Output	Best fitting function	Multiple hypotheses	One solution
Features	No domain expert required	Prone to generalizations (Posets) and requires an econder	Requires a domain expert
Problems	Can handle only the forseen negative cases	Combinatorial solutions, but hypotheses can be extracted	Combinatorial solution
Explainable	**No** (black-box)	**Yes** (hypothesis)	The algorithm.

Coming up next!

How In-Silico Learning can help Videogames?

- Machine Learning techniques are able to:
  - classify objects
  - predict user behaviours
- Data Mining techniques are able to:
  - classify users by their behaviour pattern
  - extract common behaviour patterns from the data
- Search Problems were already covered in the Game Tech module.

References

References

- S. Russel and P. Norvig. 2010. "*Artificial Intelligence: A Modern Approach, 3rd edition*" (3rd ed.). Pearson.
- Luc De Raedt. 2008. "*Logical and Relational Learning*". Springer Verlag.
- J. Pérez, J. Marinković, P. Barceló. 2019. "*On the Turing Completeness of Modern Neural Network Architectures*". In ICLR 2019.
- M. Prates, …, and M. Y. Vardi. 2019. "*Learning to Solve NP-Complete Problems: A Graph Neural Network for Decision TSP*". In AAAI 2019.