

Machine Learning for researchers (300001030)

Unit 0. Introduction to Machine Learning

Javier Vales Alonso

Doctorate transversal activity

2020

Technical University of Cartagena

Introduction

Machine learning

- Definition and examples

- Data representation

- Types of learning

Related fields

Scientific programming packages

Further reading

How to study this unit?

1. Do a first reading of the unit's slides
2. Next, its recommended to read some of the suggested references (see Syllabus).
3. Do a second reading of the unit's slides and try to identify problems in your field that could be addressed using the learning models discussed.
4. In case of doubts, contact the teacher.

Introduction

Artificial Intelligence

- Modern efforts in the artificial intelligence (AI) field dates back to the second half of the 20th century, though some historical and fictional works predate them.
- A historical summary and a timeline of AI can be read respectively at [history](#) and [timeline of AI](#).
- AI has experienced a series of spring periods intermingled with [AI winters](#). Currently, the paradigm of machine learning has boost expectations, and AI is again in a hype cycle. The previous one was related to the development of [expert systems](#) in the 80s. Like then, now, the AI scientists face ambitious challenges and are, perhaps, over-optimistic.

Artificial Intelligence (II)

Different perspectives and *goals* in AI arise from different fields. Some leading questions, that have paved the way on the development of AI, are:

- Philosophy: Can formal rules be used to draw valid conclusions? How does the mind arise from the physical brain? Where does knowledge come from?
- Biology: How does the brain work? Can we emulate/mimic its behavior?
- Psychology: What learning means? How cognitive processes work?
- Linguistics: How can knowledge be represented? What is the relationship between the mind and the language?

Artificial Intelligence (III)

- Computer science: How agents can be efficiently built?
- Physics: Which are the physics under the processes ruling conscience?
- Mathematics: How can knowledge be formalized? How logic reasoning can be applied to extract new knowledge?¹ What can be computed? Which kind of models can be applied to explain data? How to optimally compute these models? How do we reason with uncertainty?

¹Actually, this approach has intrinsic limitations, see [Gödel's incompleteness theorems](#).

Currently, there are different approaches to what AI encompasses (see Russell ch 1), broadly divided in four categories:

1. Thinking humanly (Cognitive science: how do humans think?)
2. Acting humanly (Turing's approach)
3. Thinking rationally (Logicians' approach)
4. *Acting rationally* (The practical approach: try to achieve the *best* outcome)

Artificial Intelligence (V)

Some fields, where AI has shown a high potential, and where many applications already exists, are:

- Robotics (object manipulation and move about), including, e.g., autonomous driving vehicles.
- Speech recognition.
- Autonomous planning and scheduling, e.g., MAPGEN, including logistics planning.
- Game playing, e.g., starcraft open mind AI.
- Natural language process (NLP) (enable agents to communicate successfully with humans).
- Machine translation (translate between human languages).
- Computer vision (CV) (perceive objects).

Artificial Intelligence (VI)

This course is focused on developing agents which *acts rationally*, aimed at achieving the best outcome or the best *expected* outcome (under uncertainty).

In particular, the course is devoted to *machine learning* (ML), a type of mathematical-centric approach which bases its operation on the availability of *data*, and pursues to find optimal and efficient ways of explaining this data and anticipate outcomes under new, unseen, situations.

ML has achieved great successes in many of the challenges that an agent *acting humanly* requires (NLP, CV, Speech recognition, etc.).

Machine learning

Definition and examples

Some definitions:

- Machine learning is the field of study that gives computers the ability to learn without being explicitly programmed (Arthur Samuel, 1959)
- Machine learning investigates how *to adapt to new circumstances and to detect and extrapolate patterns* (Russell)
- A computer program is said to learn from *experience* E with respect to some *task* T and some performance measurement P, if its performance on T, as measured by P, improves with experience E (Tom Mitchell, 1997)

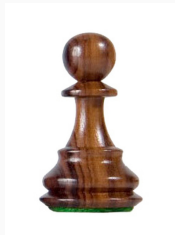
Definition and examples (II)

A typical example (indeed, the first mainstream ML application) is a *spam filter*. The task T is filtering those mails which contain spam. The experience E are the mails flagged by users as Spam/Not spam. The performance measurement P could be the ratio of correctly classified emails (called *accuracy*).

Another example could be an application whose task T is to detect wrong pieces produced by an industrial process using computer vision. The experience E are pictures of regular pieces and the performance measurement P could be the ratio of detected wrong pieces (called *recall*).

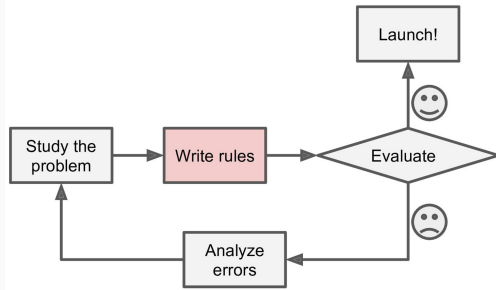
Definition and examples (III)

A more complex example could be a computer learning to play chess from scratch. The task T is selecting piece movements that lead to a victory in the game, with P being the ratio of victories, for example.



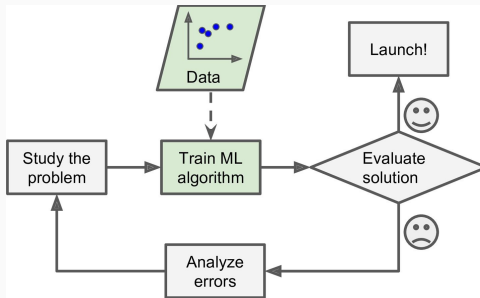
Since learning is from scratch, there are no historical records available; the experience E is limited to the result of new played games. Moreover, a movement may have a long-lasting effect on the outcome of the game, not only at the particular moment when it was played.

Machine learning vs Rule programming



- In the classical approach, the programmer would manually craft rules (heuristics) by inspecting what looks like relevant data (e.g., in the spam filtering application they can be related to the presence of key words like “Lottery”, “Nigerian” or so)
- The program will likely end with a long list of incomplete, inaccurate, and complex rules

Machine learning vs Rule programming (II)



- ML automatically finds patterns in the data related to the task. For example, in the spam filtering example, it should automatically detect unusually frequent words that don't appear in genuine mails.
- Programs are shorter, easier to maintain, and (hopefully) much more accurate.

Data representation

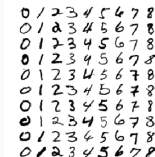
The set of experiences E is called the *data set* or the *training set*. It consists of a set of N points (also called *records* or *instances*) $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ which will be denoted as $\{\mathbf{x}_n\}$. Each instance is D -dimensional, and each of its components is called an *input* (or a *predictor*). The whole data set can be represented as a $N \times D$ matrix \mathbf{X} .

Instead working directly with the input variables, they are usually pre-processed and transformed to *features*. The feature space is created by a set of M singled-valued functions $\{\phi_m\}$. Analogous to \mathbf{X} , a $N \times M$ matrix Φ can be defined in the feature space.

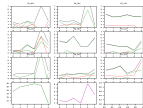
This input form is called a *factored representation*. Other forms, such as **first-order logic** are also possible, but unusual in the ML literature.

Data representation (II)

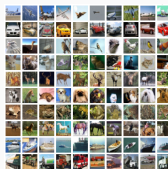
Some examples of data sets:



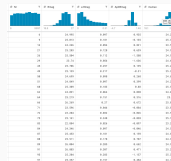
Each “point” in the data set is a 28×28 black and white image with a number handwritten. Thus, instances have 400 binary (0 or 1) input variables. See [MNIST database](#).



In this example, each instance in the data set is a time-series with acceleration measurements (along x , y , and z axes) obtained in 9 sensors (the 9 upper subplots) placed inside a punching bag. For a punch of length L , the data instance has $L \times 9 \times 3$ real-valued input variables.

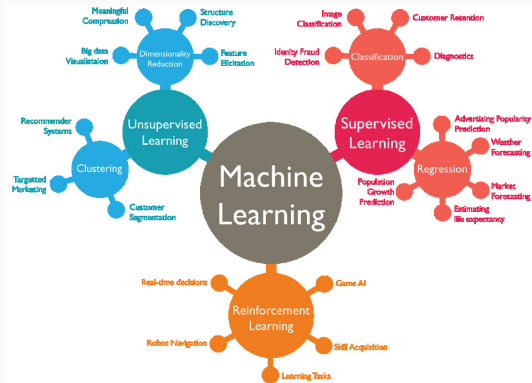


Each point in the data set contains pictures of different objects as 32×32 RGB images. Therefore each instance has 3072 byte-valued input variables. See [CIFAR-10 data set](#).



This data set contains brightness measurements in 17 bands in the visible band for 3462 galaxies, with a total of 65-dimensional real-valued input variables. See [COMBO-17](#)

Types of learning



ML algorithms are commonly classified *according to the feedback available to learn from*, leading to several types: Supervised, unsupervised, semi-supervised, and reinforcement learning.

Types of learning (II)

Another important classification criterion is whether or not the algorithm learns incrementally receiving new feedback on the fly or by processing the data offline (*online vs offline learning*).

The algorithms may be *instance* or *model* based. The former makes decisions by directly comparing the new instance to those in the data set. The latter, on the contrary, proposes a *hypothesis* model (either deterministic or probabilistic), then fits its parameters based on the training data, and later the fitted model is used to make predictions on new data.

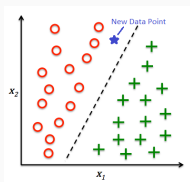
Supervised learning

In supervised learning, the *output* obtained from the process under study when the data \mathbf{x}_n was observed is also available. The output is also called the *label* or the *target* and it is denoted as t_n . The goal of supervised learning is to *predict* the output for previously *unseen* data \mathbf{x}_{new} .

The process of assigning to each data instance in the training set its corresponding output value is called *labeling*, and, often, it has to be performed manually.

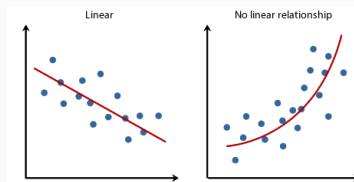
Supervised learning (II)

When the output t is one of a finite set of values (e.g. spam/no spam, joy/sorrow/surprise, etc.) the learning problem is called *classification*. It is called *binary* if there are only two possible classes, or *multi-class* if more. When t takes values on a continuous domain (a real-valued number in most cases), the problem is called *regression* (e.g. tomorrow's Tesla top stock price).



Example of a binary classification.

Predicted output is a class (either a red circle or a green cross).



Example of linear regression.

Predicted output is a real number (the red line or the red curve).

Supervised learning (III)

The foremost supervised learning algorithms are:

- Linear regression. A model-based method for regression problems.
- Logistic regression. A model-based method for linear classification problems.
- *K*-Nearest Neighbors. An instance-based method used for classification (with regression versions too).
- Decision Trees. A model-based method used in classification problems (there is also a version called Regression Trees, for regression problems), based on information theory.
- Random forest. What else than a bunch of decision trees could this be? :) The idea is making independent classifications (each by one tree) and take the majority vote.

Supervised learning (IV)

- Support Vector Machines. A sparse instance-based method based on finding the maximum margin separation boundaries.
- Artificial neural networks (ANN). A non-linear model-based method aimed both at classification and regression problems. Some unsupervised algorithms are also based on ANNs.
- Dense neural networks (DNN). ANNs composed by multiple intermediate layers using a variety of activation functions and connection formats.
- Convolutional neural networks (CNN). DNNs commonly applied to image classification with specific layers and input formats.

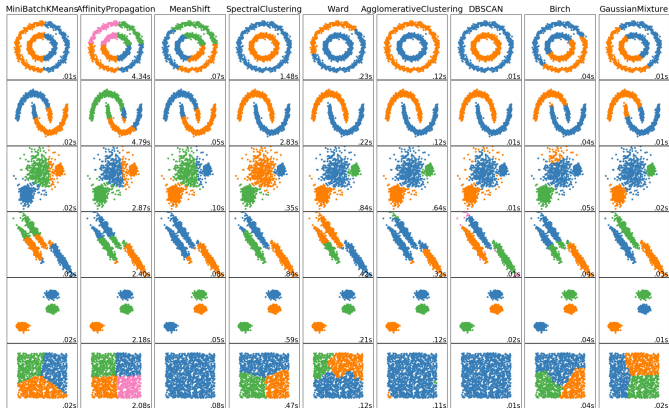
Unsupervised learning

Unsupervised learning is an alternative to previous methods when the data set is not labeled. The main goals of unsupervised learning are discovering similarities (patterns), detecting anomalies and do data preprocessing (coding, auto-labeling, etc.).

The most important unsupervised algorithms are:

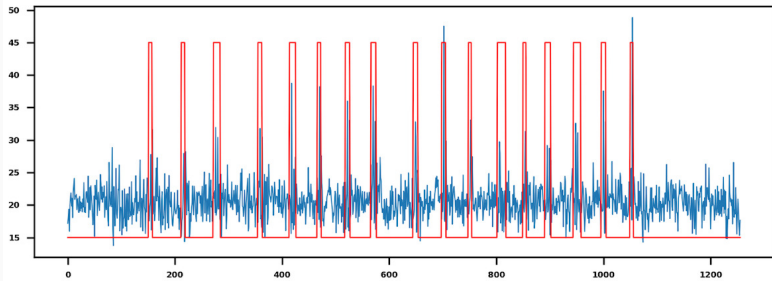
- Clustering: K -means, Hierarchical cluster analysis (HCA), Expectation-Maximization, etc.
- Outlier detection: Distance-based, e.g., Mahalanobi's distance, Autoencoder-based, Local Outlier Factor (LOF), etc.
- Dimensionality reduction: Autoencoder-based, Principal Component Analysis (PCA), Locally-Linear Embeddings (LLE), etc.

Unsupervised learning (II)



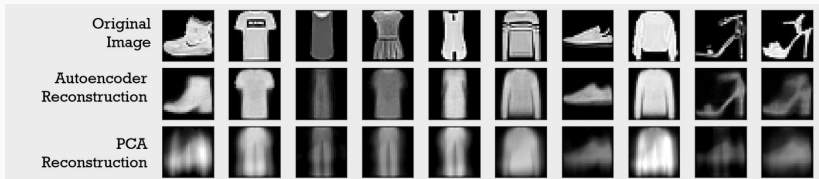
- Data clustering.
- Group points of the data set by their commonalities.
- Essential tool for **data mining** in many scientific fields, e.g. economics and finance, astronomy, biology, etc.

Unsupervised learning (III)



- Outlier detection.
- Find points in the data set (including time series) which seems different than the rest.
- Useful in many fields, e.g., predictive maintenance, forecasting, sensor data pre-analysis, etc.

Unsupervised learning (IV)



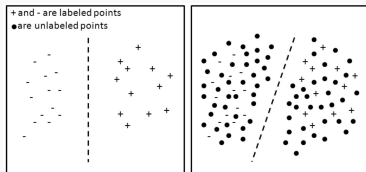
- Dimensionality reduction.
- Find automatically efficient coding of the data
- Useful for feature learning, generative models^a or data visualization among others.

^aSee Further reading section.

Semi-supervised learning

Semi-supervised learning combines previous methods. They are usually employed to do a preliminary labeling by clustering data and then assigning labels (to the whole cluster at a time). Although imprecise, this method can save time if applied properly.

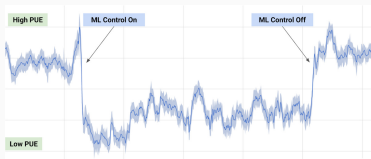
Other uses include working simultaneously with both labeled and unlabeled data to, for example, increase the accuracy of classifiers. This last case is shown in the right figure. See [link](#).



Reinforcement learning

Reinforcement learning (RL) is the most different of the learning types. In RL, an agent can observe the environment to obtain information (called the *state*), perform *actions*, and get *rewards* in return. The goal is to learn strategies (*policies*) that maximize the reward over time. RL has to face environments where the reward is stochastic and depends, not only on the last action-state pair, but on the previous history of states and actions. A common assumption is to model the environment as a [Markov Decision Process](#), where the history is limited to the last stage.

Reinforcement learning (II)



A RL algorithm by DeepMind controlled the cooling system of the Google Data Center achieving savings of 40%.



Also developed by DeepMind, an RL agent for playing starcraft, which beats professional players.

1. Review the machine learning examples provided in the introductory section in slides 11-12 and try to identify which of the previous types of learning fits better to each example.
2. Try to find problems in your field which can be treated using each of the different learning approaches.

Related fields

Big data

- Frequently, the concepts of *big data* and *machine learning* are mistaken.
- Big data refers to environments where the data volume exceeds what a regular computer can process (in a reasonable time) or store.
- For example, storing and checking for viruses in all the apps of a mobile market is a big data task.





- Machine learning can be performed on *small data* (this is the most frequent) or on big data (e.g. by distributing computations among a lot of nodes using Hadoop, Spark or some other platform).
- Nonetheless, it shall be noted that big data does not necessarily imply any “smart” process with the data.
- For example, a system which just stores all the wikipedia is not intelligent *per se*.



- Business intelligence (BI) is concerned with offering clear data representation and data analysis (forecasting, correlation, etc.) to the managers of a company.
- Machine learning usually considers tens, hundreds or thousands of features, which can't be represented directly to humans.
- Although some mechanisms for dimensionality reduction (see Unit 3) can be applied to represent a data set, this is not the primary use of these techniques.

Statistical inference

- **Statistical inference** uses data analysis to deduce properties of an underlying probability process.
- Common tools, such as hypothesis testing, maximum-likelihood estimators, Bayesian inference, multivariate analysis, etc. are on the basis of a comprehensive treatment of machine learning (e.g., the view adopted in Bishop or Vapnik's works).
- A statistical treatment of machine learning is usually referred to as *statistical learning*.
- Although more complex, it allows the development of more advanced techniques (some of them will be discussed in the next Units).

Scientific programming packages

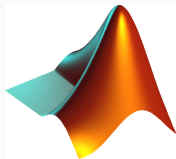
Several frameworks have support for ML. Nowadays, the most used is *Python*, a high-level interpreted programming language with specialized libraries like:



- *scikit-learn*, which implements many ML algorithms.
- *numpy*, which implements a fast matrix algebraic package
- *pandas*, which implements a data manipulation package
- *matplotlib*, which implements a 2D and 3D plotting package

Python also has bindings to *tensorflow* and *keras* for low and high-level (Deep) Neural Network construction, respectively.

Matlab is a scientific programming framework, which offers specialized toolboxes for machine learning (and many other areas).



Contrary to Python, Matlab requires a license but offers a more straightforward and unified environment. For example, in Python, it is common to have issues with data types since many different objects interwork (e.g., lists, ndarrays, dataframes, sets, etc.). On the other hand, Python is easy to deploy on different platforms, whereas Matlab can be only run on x86(-64) platforms.

Other platforms and libraries

Beyond Python and Matlab, other frameworks such as [R](#) exist, but with a shrinking community.

Implementations with high-computing demands (e.g., real-time computer vision processing) should be based on C++ or C# (both are high-level compiled programming languages). Specific libraries supporting different ML algorithms are available for them (e.g., [DLib](#)).

Finally, some related libraries available in most frameworks are [OpenCV](#), which allow still images and video manipulation (including ML algorithms targeted to image processing) or [NLTK](#) for natural language processing.

Further reading

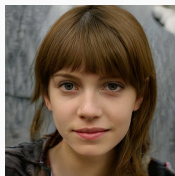
Further reading

Some fundamental topics not covered in this course (because of its limited extension) are:

- ANN y Deep learning. The second part of Geron's is a good introductory reference to this topic.
- Reinforcement learning. Two reference books in this field are "Reinforcement learning: An introduction" by Sutton and Barto (available [online](#)) and Bertsekas' "Neuro-dynamic programming."
- Online learning, which is partially related to reinforcement learning. The two references above can be consulted, or see the [link](#) and references therein, as an alternative.

Further reading (II)

Another foremost topic in ML, not discussed in these units, are the **Generative Adversarial networks** first described by Goodfellow in 2014. The core idea is to bind two neural networks contesting each other.



One (the generative network) generates candidates (with the same statistics as the data from the training set), while the second (the discriminative network) evaluates them (and both learn how to improve in their tasks). This way, artificial data can be obtained with high fidelity, like the face shown in this slide, which has been artificially generated. GANs and **autoencoders** are the base technologies for **deep fakes**.

Further reading (III)

Transfer learning is also a topic attracting a notable interest in ML. It focuses on storing knowledge gained while solving one problem and applying it to a different, but related, problem. For example, this technique is commonly used to teach to an already trained DNN to recognize new objects using less time and a minimal training set. Some starting links about transfer learning can be obtained in its [wikipedia entry](#) or in this [online tutorial](#).

[One-shot](#) and [zero-shot](#) learning are also concepts related to transfer learning. The idea is to classify data based on very few or even no labeled examples (online classifiers).

Further reading (IV)

Finally, it is worth noting that even if ML is under a great hype in the community and in companies, the current research roadmap has been heavily criticized by some of the AI field founders like John McCarthy, Marvin Minsky, Nils Nilsson or Patrick Winston, stating that less effort shall be put on specific tasks, and more on achieving a human-level AI (see [link](#)).

Futurist and transhumanist views and concerns about AI development are the core of the [Singularity theory](#) popularized by Vernor Vinge and Raymond Kurtzweil, which suggests that “there is a hypothetical future point in time at which technological growth becomes uncontrollable and irreversible, resulting in unforeseeable changes to human civilization”.