Pill 1: Machine Learning.

Oriol Pujol

Machine Learning

2022-23

Acknowledgements

This work is inspired in the courses of T. Jaakkola, M. Collins, L. Kaelbling, and T. Poggio at MIT, Andrew Ng at Stanford, Y. Abu-Mostafa at CalTech, E. Xing at CMU, P. Domingos at Georgia Tech, D. MacKay at Cambridge, and all my mentors and people who made me realize Machine Learning is one of my passions.

Learning goals.

- To know what machine learning is and what kind of problems can be addressed with it.
- Identify the different tasks in the machine learning pipeline.
- Understand the concepts of training, test, and validation.

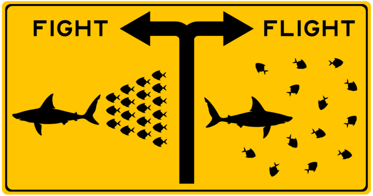
Outline

- Warm up on artificial intelligence and machine learning
- What is machine learning?
- Types of machine learning
- Examples of machine learning
- Introducing the basics, Machine learning pipeline
- Applying the machine learning model.

Warm up (I)

Shall we first talk about (Artificial) Intelligence?

- Intelligence as the capability of an agent to adapt to different scenarios and problems.
- Intelligence is reflected in the different amount of ways of giving answers or solutions to a specific scenario.



(a)

(b)

Figure: (a) Accute stress response; (b) Tools

Warm up (II)

Shall we talk about Artificial Intelligence?

- Intelligence on an artificial substract.
- Computacionalism, functionalism, and other birds of the philosophy of mind.

Warm up (III)

Let us talk about Artificial General Intelligence?

- Do we really want a colleague?

Why machine learning now?

- Digital transformation stuff and leveraging new value propositions based on data.

Let us start the lecture

Ping pong time...

- What do you think machine learning is?
- Give me examples of machine learning in production.

What is machine learning?

Definition 1

Improve the performance of a software system, based on previous experience.

Definition 2

Set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data, or to perform other kinds of decision making under uncertainty. - Kevin P. Murphy

Machine learning everywhere

Figure: Document classification and email spam filtering

Machine learning everywhere

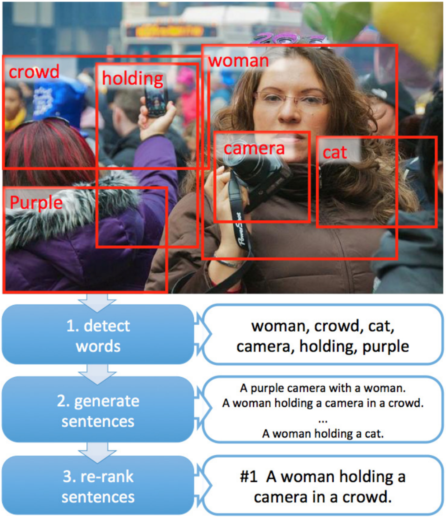


Figure: Image classification

Machine learning everywhere

Figure: Style transfer

Machine learning everywhere

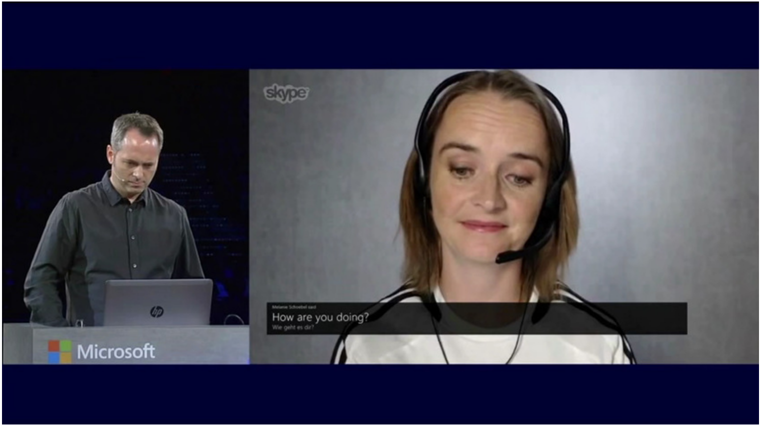


Figure: Skype translator

Machine learning everywhere

Figure: Advertisement optimizers CTR

Machine learning everywhere

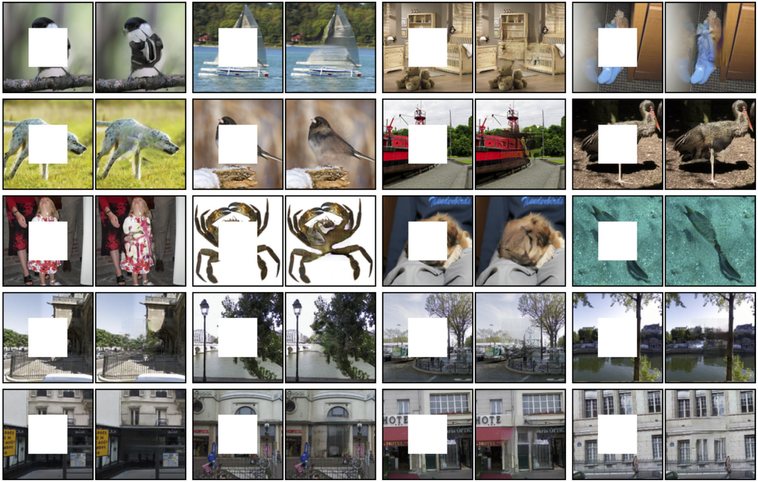


Figure: Image impainting.

Key elements for machine learning.

Machine learning is used when

- There is a pattern
- We can not pin it down mathematically
- We have data on it

Which is the most important of the three?

Key elements for machine learning.

Machine learning is used when

- There is a pattern
- We can not pin it down mathematically
- We have data on it

Which is the most important of the three? That's why we also call it Learning from data.

Types of machine learning

- **Predictive** or **supervised learning**: given a labelled data set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ pairs called *training set*, find a mapping from x to y, e.g. regression, classification (sometimes called *pattern recognition*).
- **Descriptive** or **unsupervised learning** (aka *knowledge discovery*): given a data set of $\{x\}_{i=1}^{N}$, find something interesting or useful about their structure, e.g. density estimation, clustering, dimensionality reduction.
- **Reinforcement learning** : given an external system upon which you can exert control action a and receive percepts p, a reward signal r indicating good performance, find a mapping from $P \rightarrow A$ that maximizes some long-term measure of r

Let us discuss about the following problems

- I want to know if tomorrow is going to rain.
- I want to retrieve similar movies to the one i like the most.
- I want to know if a customer is susceptible to a certain marketing strategy.
- I want to build a twitter bot that learns to post contents to please its followers.

First steps in modeling a supervised machine learning problem

Identifying an interesting question to answer:

- If our question is answered by YES/NO or a finite set of answers, we are in front of a **classification** problem.
  - Given the results of a clinical test, does this patient suffers from diabetes?
  - Given the past activity associated to a credit card, is the current operation a fraud?
  - Given my skills and marks in computer science and maths, will I pass the data science course?
- If our question is a prediction of a (usually real valued) quantity, we are in front of a **regression** problem
  - Given the past records of user activities on Apps, how long is a certain client be hocked to our App?
  - Given my skills and marks in computer science and maths, what mark will I achieve?

Let us discuss about the following problems

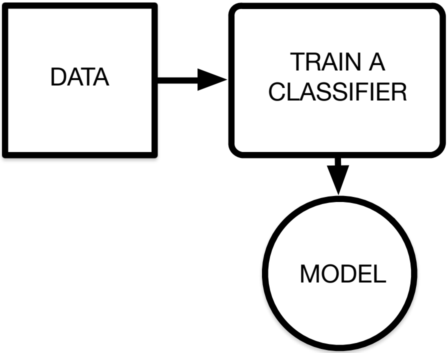
- Stock options prediction.
- Where is my car in an aerial image of a parking lot?

Generalization in supervised learning

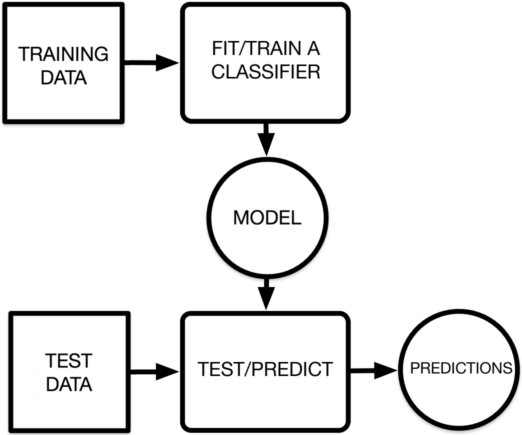
We want to find a model that will perform the best on future examples

- We don't know what the future data will be.
- We have some past data
- We **hope** that future data will remember past data in a way that while let us use the past data to construct a model that will perform well in the future.

Pragmatic machine learning pipeline



Pragmatic machine learning pipeline



Two regimes

Training and exploitation

A simple example

Suppose you want to know which grade I may obtain at the end this course.

If you know information (students record) about people who passed this course (data $\mathbf{x}$), and their performance (label y), then, given your own record you may ask the system to predict your grade based on the previous experience and the power of the learning algorithm.

An simple example

Av. math grade	Grade in ML
5.0	4.2
6.2	5.9
7.4	8.1
$\vdots$	$\vdots$

Supervised learning:
Given the "right" answer for each example in de data.

Regression problem:
Predict a real-value output.

Notation

Input (features): $\mathbf{x} \in \mathbb{R}$ (your grade in maths)
Output (labels): $y \in \mathbb{R}$ (your grade in machine learning)
Data: Examples of inputs and output pairs: $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_N, y_N)\}$

Dataset jargon

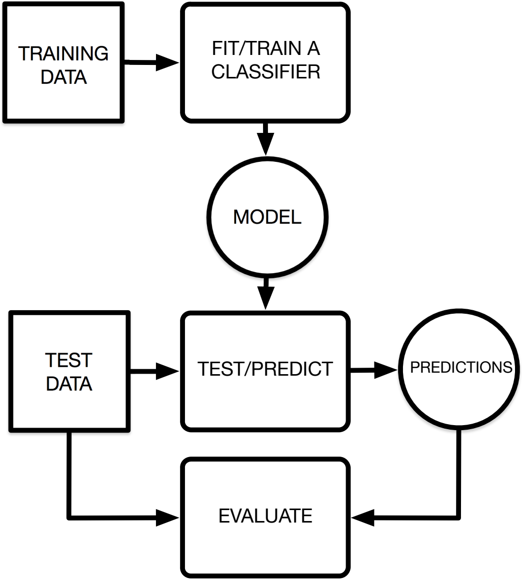
We will usually use column-wise notation for each sample.

Common jargon:

Rows: features/ attributes/ dimensions.
Columns: instances/ examples/ samples.
The feature to be predicted: target/ outcome/ response/ label/ dependent variable.
The other features: independent variables/ covariates/ predictors/ regressors.

Data is observable, can be raw or derived, and we believe it tells something about the relationship we are looking for. Ex. the student record can have direct information (the grades), or derived information (the average number of credits per semester, average grade on math courses, etc).

Pragmatic machine learning pipeline

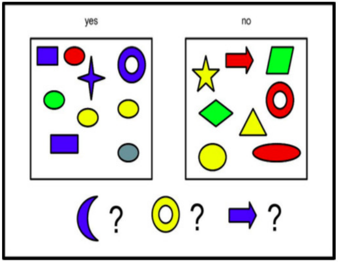


To the notebook!

To the notebook...

Let us code a little.

Knowledge representation and feature extraction



D features (attributes)

N cases

Color	Shape	Size (cm)	Label
Blue	Square	10	1
Red	Ellipse	2.4	1
Red	Ellipse	20.7	0

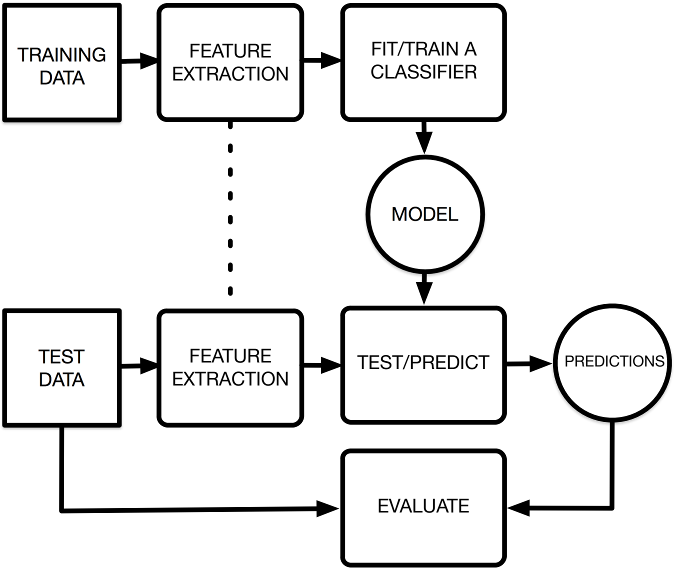
Notation

Input (features): $\mathbf{x}$ (a description of the figures)
Output (labels): $y \in \{\text{left}, \text{right}\}$ (box containing the figure)
Data: Examples of inputs and output pairs: $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$

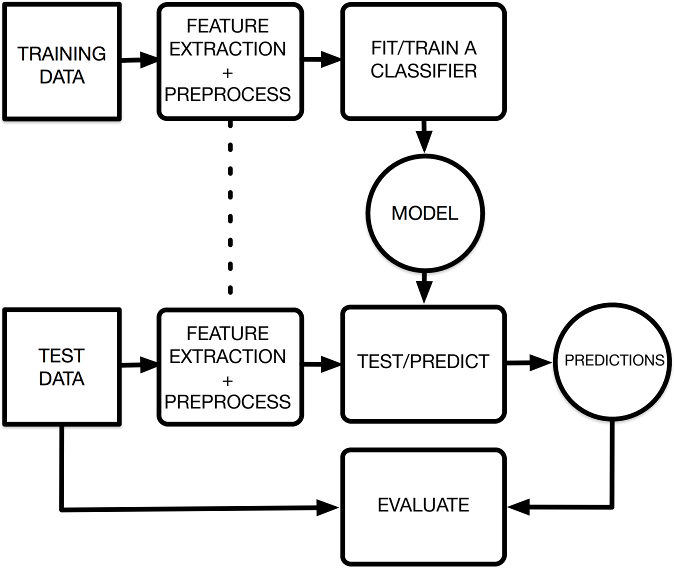
According to the type of data, we can talk about:

- iid (independent identically distributed) vectors
- Time series (dependent vectors)
- Images (matrices)
- Variable-size non-vector data (e.g. strings, trees, graphs, text)
- Objects (e.g. within a relational schema)

Pragmatic machine learning pipeline



Pragmatic machine learning pipeline



To the notebook!

To the notebook...

Let us code a little.

Application of the learning model.

Training

We are given a dataset $\mathcal{D}$ which we use to learn our model parameters, e.g. the parameters of the linear regressor.

Exploitation

Apply the learned model to new data and **hope** it predicts correctly.

A first evaluation of the learning model.

But... we want to know approximately how well it will perform during the exploitation step!!. What to do?

A first evaluation of the learning model.

But... we want to know approximately how well it will perform during the exploitation step!!. What to do?

The simplest evaluation strategy: Simulate exploitation data.

We are given a dataset $\mathcal{D}$ and it is divided in two sets

$$\mathcal{D} = \mathcal{D}_{train} \cup \mathcal{D}_{test}$$

A first evaluation of the learning model.

But... we want to know approximately how well it will perform during the exploitation step!!. What to do?

The simplest evaluation strategy: Simulate exploitation data.

We are given a dataset $\mathcal{D}$ and it is divided in two sets

$$\mathcal{D} = \mathcal{D}_{train} \cup \mathcal{D}_{test}$$

Training

Use $\mathcal{D}_{train}$ to learn the model.

Evaluation

Use $\mathcal{D}_{test}$ to compute the performance of the method.

Exploitation

Apply the model to new data and **hope** it predicts correctly.

A first evaluation of the learning model.

Suppose that we have two different models. We want to select the one that has a better performance during the exploitation step. But... we don't have exploitation data to evaluate on. What to do?

A first evaluation of the learning model.

Suppose that we have two different models. We want to select the one that has a better performance during the exploitation step. But... we don't have exploitation data to evaluate on. What to do?

Simulate exploitation data.

We are given a dataset $\mathcal{D}$ and it is divided in two sets

$$\mathcal{D} = \mathcal{D}_{train} \cup \mathcal{D}_{validation}$$

A first evaluation of the learning model.

Suppose that we have two different models. We want to select the one that has a better performance during the exploitation step. But... we don't have exploitation data to evaluate on. What to do?

Simulate exploitation data.

We are given a dataset $\mathcal{D}$ and it is divided in two sets

$$\mathcal{D} = \mathcal{D}_{train} \cup \mathcal{D}_{validation}$$

Training

Use $\mathcal{D}_{train}$ to learn both models.

Evaluation

Use $\mathcal{D}_{validation}$ to decide which of the two models better adapts to our problem.

Apply the selected model to new data and **hope** it predicts correctly.

A first evaluation of the learning model.

But... I want to know the expected performance of the best method!!!
What to do?

A first evaluation of the learning model.

But... I want to know the expected performance of the best method!!!
What to do?

Simulate exploitation data.

We are given a dataset $\mathcal{D}$ and it is divided in three sets

$$\mathcal{D} = \mathcal{D}_{train} \cup \mathcal{D}_{validation} \cup \mathcal{D}_{test}$$

A first evaluation of the learning model.

But... I want to know the expected performance of the best method!!!
What to do?

Simulate exploitation data.

We are given a dataset $\mathcal{D}$ and it is divided in three sets

$$\mathcal{D} = \mathcal{D}_{train} \cup \mathcal{D}_{validation} \cup \mathcal{D}_{test}$$

Training and model selection

Use $\mathcal{D}_{train}$ to learn both models. Use $\mathcal{D}_{validation}$ to decide which of the two models better adapts to our problem.

Evaluation

Use $\mathcal{D}_{test}$ to compute the performance of the selected method.

Apply the selected model to new data and **hope** it predicts correctly.

Question

Supose that we also evaluate the second model and realize that it has better performance in the test set. What do we do?