Machine learning and physical modelling-1

julien.brajard@nersc.no

21-25 January 2018

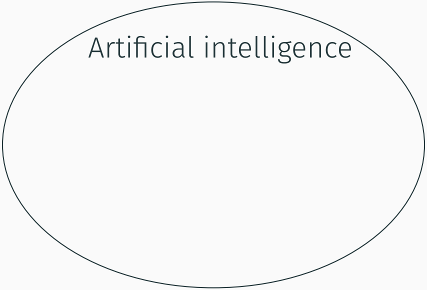
NERSC/Sorbonne University

https://github.com/brajard/Geilo-Winter-school

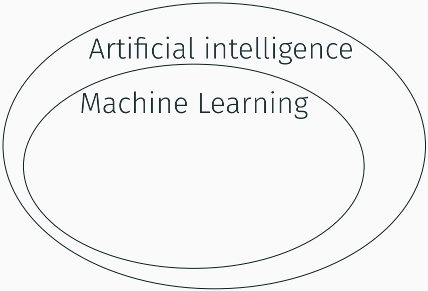
Table of contents

Introduction

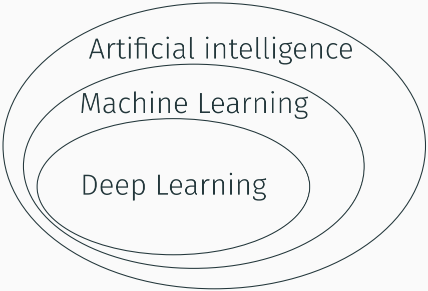
Scope of the lectures



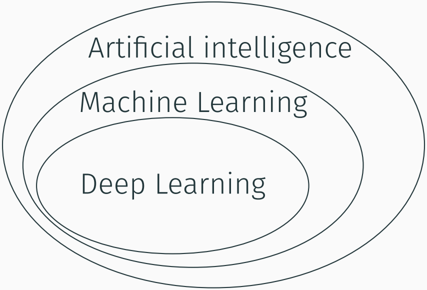
Scope of the lectures

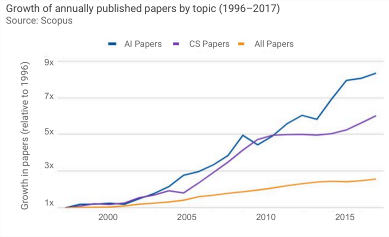


Scope of the lectures

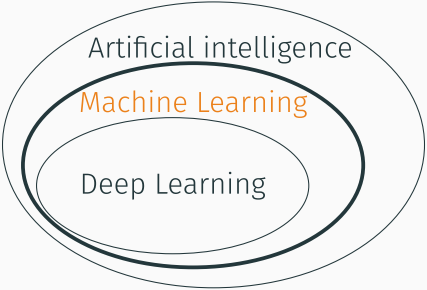


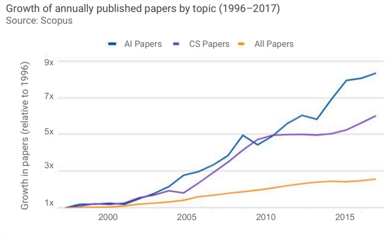
Scope of the lectures

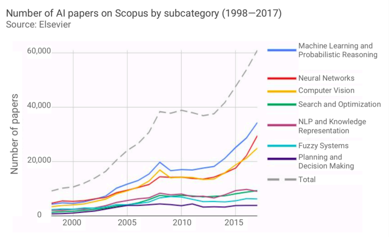




Scope of the lectures

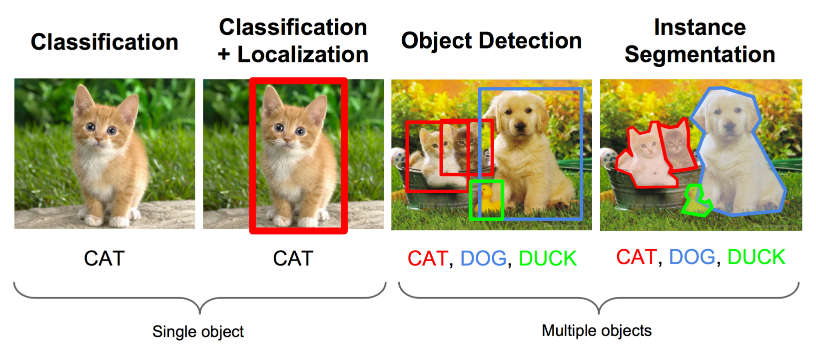






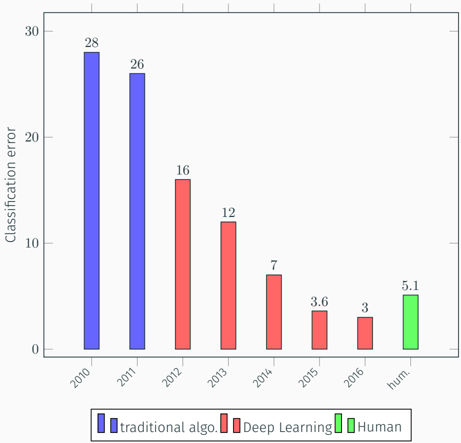
Shoham et al., AI Index 2018 Annual Report

Example 1: Computer Vision



Li, Karpathy and Johnson, 2016, Stanford CS231n course

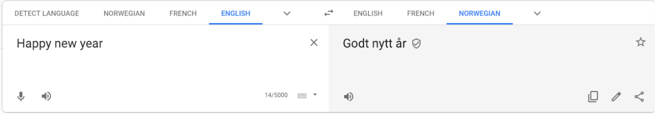
Example 1: Computer Vision



Deep learning architectures were based on Convolutional Neural Networks (CNN).

Machine Translation

Objective : translate a text from a language to another.



- **Oct. 2013**: Pionneering scientic paper about neural networks in machine translation (Kalchbrenner, N., and Blunsom, P).
- **2016**: Neural machine translation outperform tradiational approaches on public benchmarks
- **2017**: Major systems (Google, Systran, WIPO, Microsoft) switch to neural machine translation (using deep recurrent neural networks)

Playing Games

- **1997**: Deep Blue defeats Kasparov at Chess.
- **2016**: AlphaGo's victory again Lee Sedol at Go.
- **2017**: AphaGo Zero learns how to play Go only by playing against itself. It outperformed previous AlphaGo version (Reinforcement learning)
- **2017**: DeepStack beats professional human poker players.

Predicting sport results?

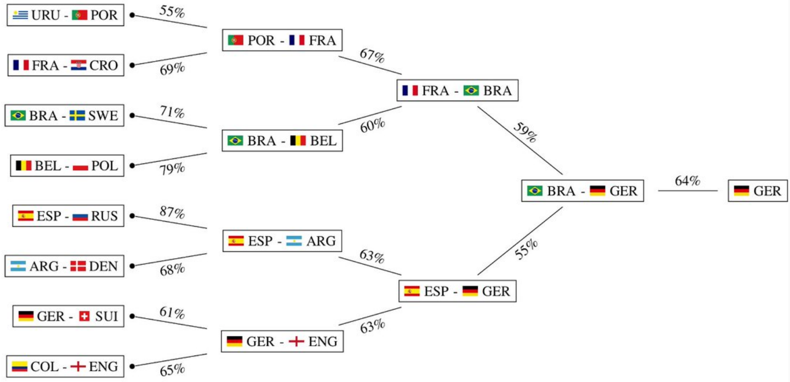


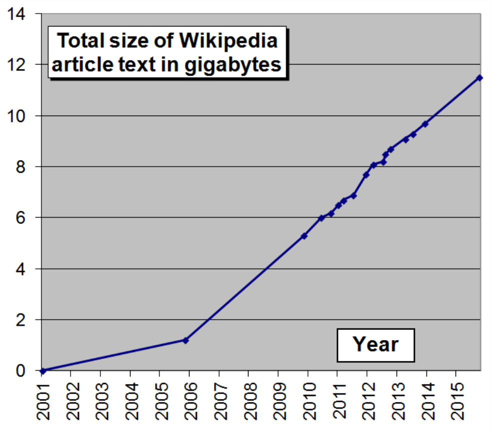
Figure 5: Most probable course of the knockout stage together with corresponding probabilities for the FIFA World Cup 2018 based on 100,000 simulation runs.

Prediction based on Random Forest.

Groll et al., June 2018

Reasons for these recent achievements?

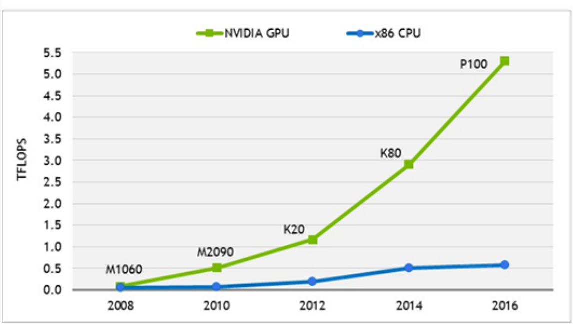
- Increasing of the datasets in size and quality



source: Wikipedia

Reasons for these recent achievements?

- Increasing of the datasets in size and quality
- Progress in computing resources.



source: NVDIA

Reasons for these recent achievements?

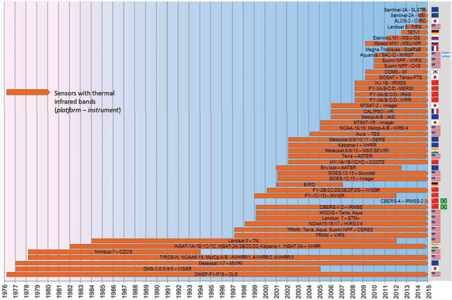
- Increasing of the datasets in size and quality
- Progress in computing resources.
- Scientific research on new algorithms (e.g adapted to image processing)

source: Deep Dream Generator

Apply Machine-Learning to physical modelling?

Why is it a good idea?

- A increasing number of geophysical data (one spatial mission: 24 TB/day)



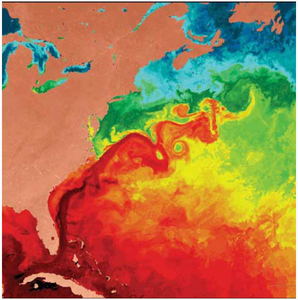
Satellites providing a semi-global daily coverage of surface temperature

source: Kuenzer et al. (2014)

Apply Machine-Learning to physical modelling?

Why is it a good idea?

- A increasing number of geophysical data (one spatial mission: 24 TB/day)
- Data with highly significant spatial patterns

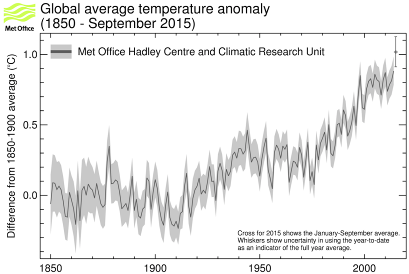


Sea Surface temperature of the gulf stream

source: Talley (2000)

Why is physical modelling specific?

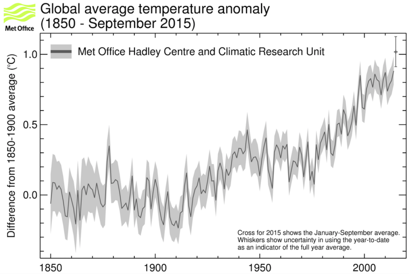
NASDAQ Composite sock market index over the last 10 years



Why is physical modelling specific?

NASDAQ Composite sock market index over the last 10 years

Mostly unknown dynamic processes



Mostly known dynamic processes (based on physical principles)

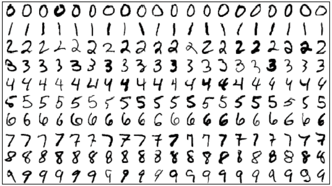
Generalities on Machine Learning

What is this about ?

Can we extract knowledge, make some predictions, determine a ”model” using this large amount of data ?

What is this about ?

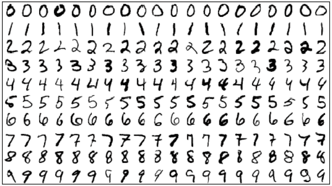
Can we extract knowledge, make some predictions, determine a ”model” using this large amount of data ?



Base of images

$\longrightarrow$ Digit $\in \{0, \ldots, 9\}$

Can we extract knowledge, make some predictions, determine a "model" using this large amount of data ?



$\longrightarrow$ Digit $\in \{0, \ldots, 9\}$

Base of images

- From high dimensional data (thousands to millions dimensions) to reduced dimensional data (less than 100)
- From disorganized data to comprehensive information
- Can we teach a machine how to do that ?

Two classes of Machine Learning problems

1. Regression: Determination of a quantitative variable from a set of data
    - The price of a building from various predictors (Surface, ...)
    - A physical value (Temperature, humidity, ...) in the future knowing the past
    - ...

Two classes of Machine Learning problems

1. Regression: Determination of a quantitative variable from a set of data
    - The price of a building from various predictors (Surface, ...)
    - A physical value (Temperature, humidity, ...) in the future knowing the past
    - ...
2. Classification: Determination of a class
    - A digit from a image
    - Identification of the content of an image
    - ...

Two types of objectives

1. Supervised learning: we have a set of labeled data with examples of targets.

Two types of objectives

1. Supervised learning: we have a set of labeled data with examples of targets.
2. Unsupervised learning: we only have unlabeled the data, we have no examples of what we want to obtain. We want to extract a ”useful” representation of these data, or some coherent categories.

Two types of objectives

1. Supervised learning: we have a set of labeled data with examples of targets.
2. Unsupervised learning: we only have unlabeled the data, we have no examples of what we want to obtain. We want to extract a ”useful” representation of these data, or some coherent categories.
   - Determine typical behaviors of clients in a supermarket knowing what the have bought.

Two types of objectives

1. Supervised learning: we have a set of labeled data with examples of targets.
2. Unsupervised learning: we only have unlabeled the data, we have no examples of what we want to obtain. We want to extract a ”useful” representation of these data, or some coherent categories.
    - Determine typical behaviors of clients in a supermarket knowing what the have bought.
3. Semi-Supervised Learning: Only a few subset of the data are labeled

Two types of objectives

1. Supervised learning: we have a set of labeled data with examples of targets.
2. Unsupervised learning: we only have unlabeled the data, we have no examples of what we want to obtain. We want to extract a ”useful” representation of these data, or some coherent categories.
    - Determine typical behaviors of clients in a supermarket knowing what the have bought.
3. Semi-Supervised Learning: Only a few subset of the data are labeled
4. Reinforcement Learning: We can initiate and observe the interaction of an agent with its environment. We want to optimize the behavior of the agent.

Two types of objectives

1. Supervised learning: we have a set of labeled data with examples of targets.
2. Unsupervised learning: we only have unlabeled the data, we have no examples of what we want to obtain. We want to extract a ”useful” representation of these data, or some coherent categories.
   - Determine typical behaviors of clients in a supermarket knowing what the have bought.
3. Semi-Supervised Learning: Only a few subset of the data are labeled
4. Reinforcement Learning: We can initiate and observe the interaction of an agent with its environment. We want to optimize the behavior of the agent.
   - Playing a chess game.

Formally

A Machine

$$y = \mathcal{M}(x, \theta)$$

- x: input
- y: output
- $\mathcal{M}$: a model (named ”machine”)
- θ : parameters of the model $\mathcal{M}$.

Machine learning consists in optimizing θ using a set of data. This is the training process.

A Machine

$$y = \mathcal{M}(x, \theta)$$

What are the ingredients?

The Machine Learning recipe

A Machine

$$y = \mathcal{M}(x, \theta)$$

What are the ingredients?

- Some data
  - x, y : supervised learning
  - only x: unsupervised learning
  - x and some subset of y: semi-supervised learning

The Machine Learning recipe

A Machine

$$y = \mathcal{M}(x, \theta)$$

What are the ingredients?

- Some data
  - x, y : supervised learning
  - only x: unsupervised learning
  - x and some subset of y: semi-supervised learning
- An objective
  - y is quantitative: regression
  - y is a class: classification

The Machine Learning recipe

A Machine

$$y = \mathcal{M}(x, \theta)$$

What are the ingredients?

- Some data
  - x, y : supervised learning
  - only x: unsupervised learning
  - x and some subset of y: semi-supervised learning
- An objective
  - y is quantitative: regression
  - y is a class: classification
- A computational architecture (the machine)
  - linear
  - non-linear
  - neural networks, random forest, ...

The Machine Learning recipe

A Machine

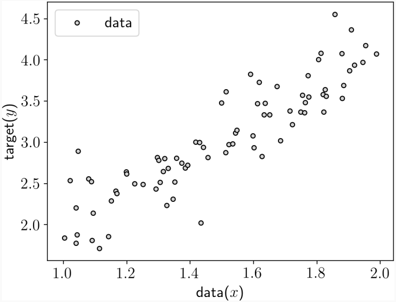
$$y = \mathcal{M}(x, \theta)$$

What are the ingredients?

- Some data
  - x, y : supervised learning
  - only x: unsupervised learning
  - x and some subset of y: semi-supervised learning
- An objective
  - y is quantitative: regression
  - y is a class: classification
- A computational architecture (the machine)
  - linear
  - non-linear
  - neural networks, random forest, ...
- A learning process
  - Estimation of θ

An illustration

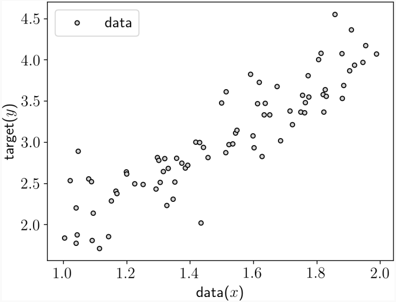
- Some Data



  - y is known: supervised learning
  - y is quantitative: regression

An illustration

- Some Data

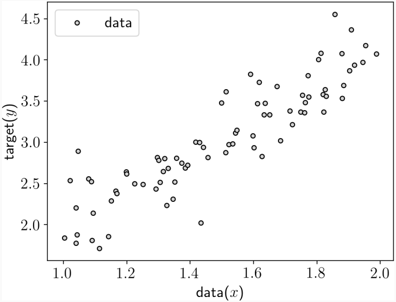


  - y is known: supervised learning
  - y is quantitative: regression

- An Objective: Estimate $\hat{y}$ from x by minimizing $(\hat{y} - y)^2$ (Least-square objective)

An illustration

- Some Data



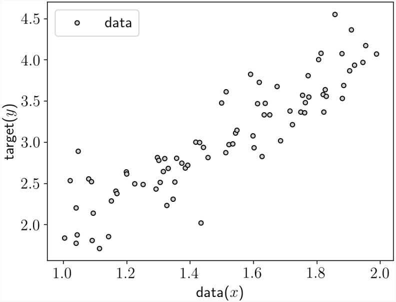
  - y is known: supervised learning
  - y is quantitative: regression

- An Objective: Estimate $\hat{y}$ from x by minimizing $(\hat{y} - y)^2$ (Least-square objective)

- A model: $y = \theta_1 X + \theta_0$ (linear)

An illustration

- Some Data



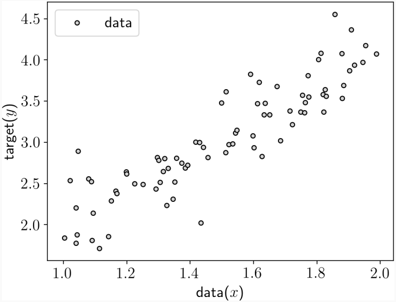
  - y is known: supervised learning
  - y is quantitative: regression

- An Objective: Estimate $\hat{y}$ from x by minimizing $(\hat{y} - y)^2$ (Least-square objective)

- A model: $y = \theta_1 X + \theta_0$ (linear)
- A learning process: $\theta = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T y$

An illustration

- Some Data

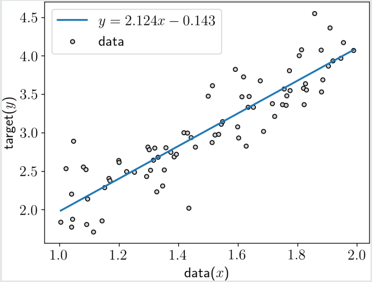


  - y is known: supervised learning
  - y is quantitative: regression

- An Objective: Estimate $\hat{y}$ from x by minimizing $(\hat{y} - y)^2$ (Least-square objective)

- A model: $y = \theta_1 X + \theta_0$ (linear)
- A learning process: $\theta = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T y$

Result



Choice of the model

Polynomial regression

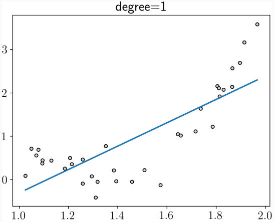
$y = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_d x^d = \sum_{i=0}^{d} \theta_i X^i$

Choice of the model

Polynomial regression

$$y = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_d x^d = \sum_{i=0}^{d} \theta_i X^i$$

degree = 1 (linear)

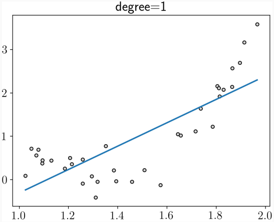


Choice of the model

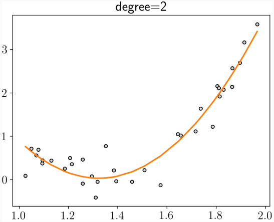
Polynomial regression

$$y = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_d x^d = \sum_{i=0}^{d} \theta_i X^i$$

degree = 1 (linear)



degree = 2

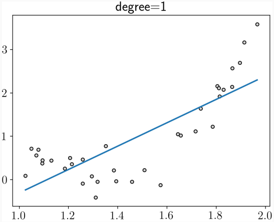


Choice of the model

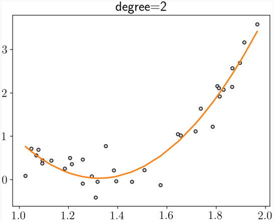
Polynomial regression

$$y = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_d x^d = \sum_{i=0}^{d} \theta_i X^i$$

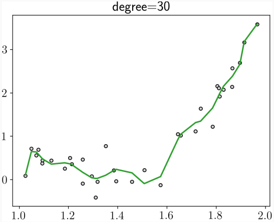
degree = 1 (linear)



degree = 2



degree = 30



What is the best model?

Validation Dataset

The idea

Evaluate a score on a independent dataset

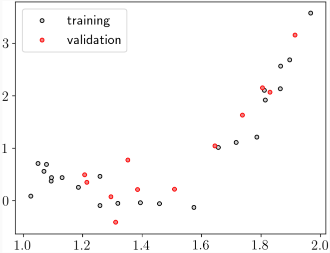
Validation Dataset

The idea

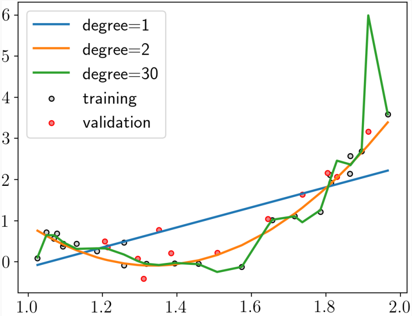
Evaluate a score on a independent dataset

In our example we can randomly divide (X, y) in two datasets:

- The training dataset X_{train}, y_{train} used to fit the model.
- The validation dataset X_{val}, y_{val} used to compute the score. e.g. core= correlation$(y_{val}, \hat{y})$

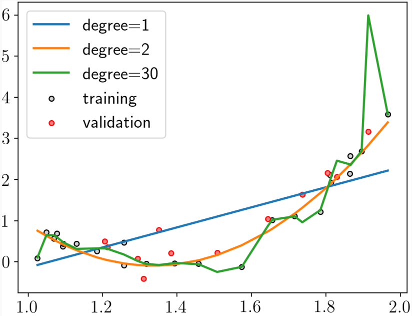


Choice of the model



Deg.	Train Score	Val. Score
1	0.60	0.68
2	0.94	0.86
30	0.99	0.19

Choice of the model



Deg.	Train Score	Val. Score
1	0.60	0.68
2	0.94	0.86
30	0.99	0.19

Drawbacks

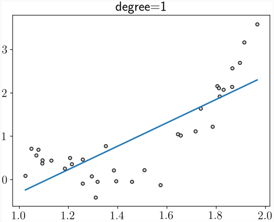
- drastically reduce the number of samples which can be used for learning the model
- Results can depend on a particular random choice for the pair of (train, validation) sets.

Choice of the model

Polynomial regression

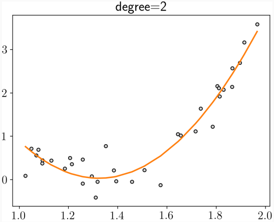
$$y = \theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_d x^d = \sum_{i=0}^{d} \theta_i X^i$$

degree = 1 (linear)



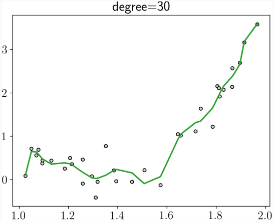
underfitting

degree = 2



good fit

degree = 30

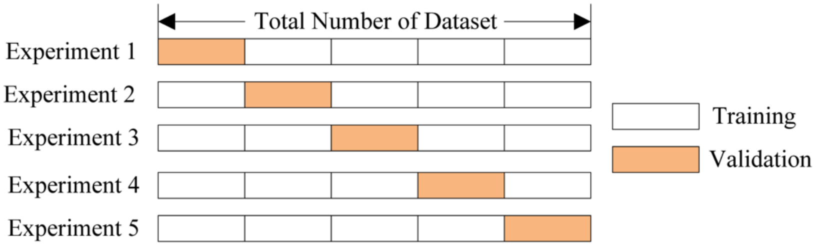


overfitting

More Robust: cross validation

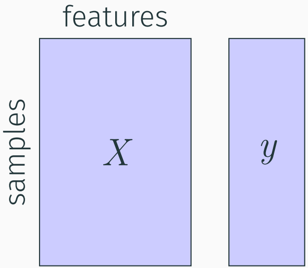
The idea

- Dividing the data in n folds,
- Learning n model (each time with a different training set),
- Compute the mean score over n validation set.



Multidimensional data

Generally, we have multidimensional data X and a one-dimensional target y.



Boston House Prices

Data challenge

Credits

Data challenge designed by Sophie Giffard-Roisin (University of Colorado, Boulder)
sophie.giffard@colorado.edu
Tusen takk!!

- One machine learning problem on climate
- Teams of 3/4 (please fill the form sent by mail if you did not!)
- Evaluation metric: leaderboard on hidden data
- Online submissions of python code

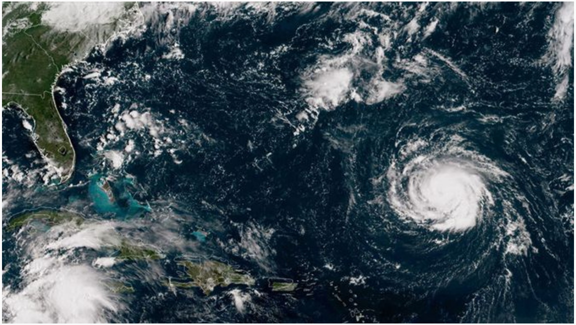
Big thank to the Paris-Saclay CDS team (Balázs Kégl & Alexandre Boucaud)

Friday Noon: Final results!

Get yourself ready!

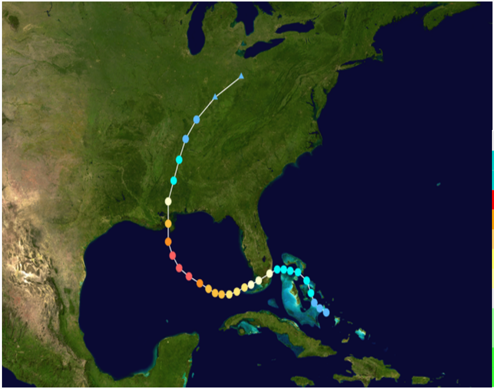
- `http://github.com/ramp-kits/storm_forecast` : notebook .ipynb for all informations
- clone starting kit ('git clone ...') / download data
- `www.ramp.studio`: create account! (wait for access)
- `www.ramp.studio/problems/storm_forecast` : register on the project!

Hurricane Intensity Forecast



Hurricane Florence, Monday Sept. 10th, 2018

Hurricanes



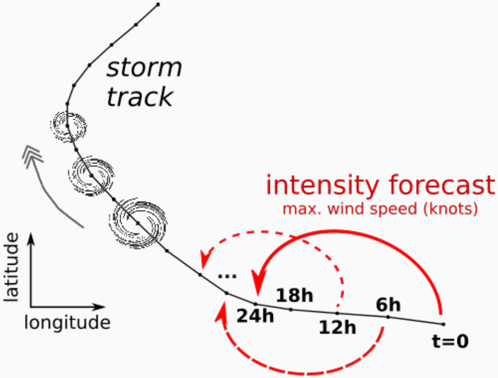
Saffir-Simpson Hurricane Scale

Category	Wind Speed mph	Wind Speed knots
5	>=156	>=135
4	131-155	114-134
3	111-130	96-113
2	96-110	84-95
1	74-95	65-83
Non-Hurricane Scale		
Tropical Storm	39-73	34-64
Tropical Depression	0-38	0-33

- Hurricane Katrina, 2005. (1 dot every 6 hours).
- **Tracks** and **Intensity** : Two main goals of the forecast

Your goal!

- Estimating the **24h-forecast intensity** *(wind speed)* of all hurricanes and storms.



Data sources

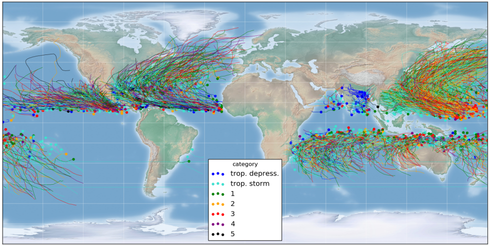
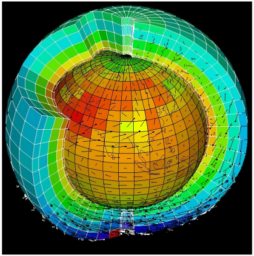


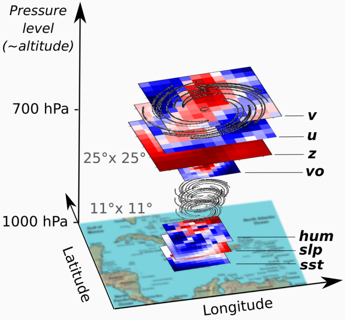
Figure 1: Database: 3000 tropical/extra-tropical storm tracks since 1979.

- 3 000 extra-tropical and tropical storm tracks (NOAA database IBTrACS)
- 90 000 total number of instants (every 6h)
- Public data (download) : $1/4$ random storms. On the server: different data + larger data

Data sources

- Reanalysis data:
  - Global atmospheric grids : wind fields, pressure, temperature, humidity...
  - Cropped and centered to the current storm location





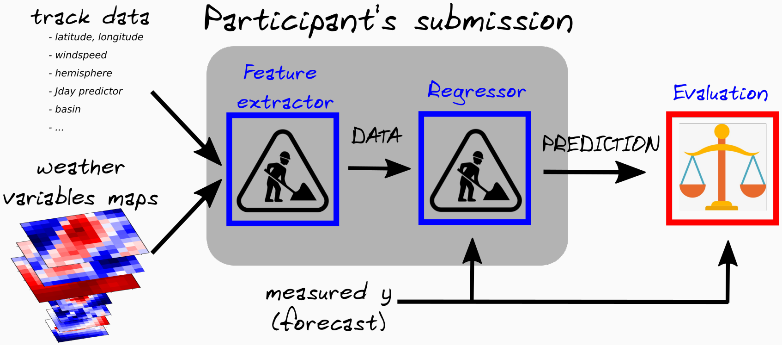
Feature data

	stormid		instant_t	latitude	longitude	windspeed	hemisphere	Jday_predictor	...
Storm #1	1982011S12119	6 h	0	-11.8	119.2	25	0	7.54774E-37	...
	1982011S12119	6 h	1	-12.2	119	25	0	7.54774E-37	...
	1982011S12119	6 h	2	-12.5	118.6	30	0	7.54774E-37	...
	1982011S12119	...	3	-12.8	118.3	35	0	7.54774E-37	...
	1982011S12119		4	-12.8	117.6	35	0	7.54774E-37	...
	1982011S12119		5	-12.8	117.2	35	0	7.54774E-37	...
	...		...	...	...	...	...	...	...
Storm #n	2014210N10323		0	9.6	-37.1	25	1	0.04677062	...
	2014210N10323		1	9.5	-38.6	30	1	0.04677062	...
	2014210N10323		2	9.5	-40.1	30	1	0.04677062	...
	2014210N10323		3	9.6	-41.5	30	1	0.04677062	...

Feature data

	stormid		instant_t	latitude	longitude	windspeed	hemisphere	Jday_predictor	...	24h forecast windspeed:
Storm #1	1982011S12119	6 h	0	-11.8	119.2	25	0	7.54774E-37	...	y_1
	1982011S12119	6 h	1	-12.2	119	25	0	7.54774E-37	...	y_2
	1982011S12119	6 h	2	-12.5	118.6	30	0	7.54774E-37	...	y_3
	1982011S12119	...	3	-12.8	118.3	35	0	7.54774E-37	...	...
	1982011S12119		4	-12.8	117.6	35	0	7.54774E-37	...	
	1982011S12119		5	-12.8	117.2	35	0	7.54774E-37	...	
	...		...	...	...	...	...	...	...	
Storm #n	2014210N10323		0	9.6	-37.1	25	1	0.04677062	...	
	2014210N10323		1	9.5	-38.6	30	1	0.04677062	...	
	2014210N10323		2	9.5	-40.1	30	1	0.04677062	...	
	2014210N10323		3	9.6	-41.5	30	1	0.04677062	...	

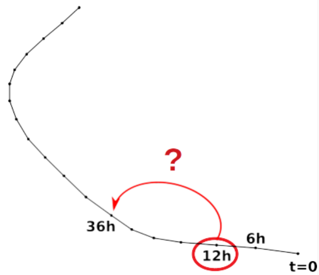
Pipeline: your work!



Data from previous steps

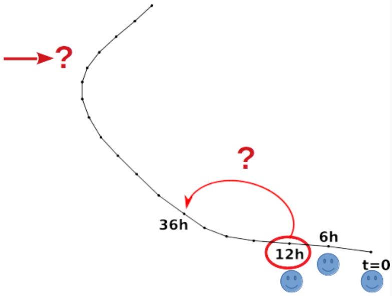
	stormid	instant_t	features
Storm #1	1982011S12119	0	...
	1982011S12119	1	...
	1982011S12119	2	...
	1982011S12119	3	...
	1982011S12119	4	...
	1982011S12119	5	...
	...	...	...
Storm #n	2014210N10323	0	...
	2014210N10323	1	...
	2014210N10323	2	...
	2014210N10323	3	...

→ ?



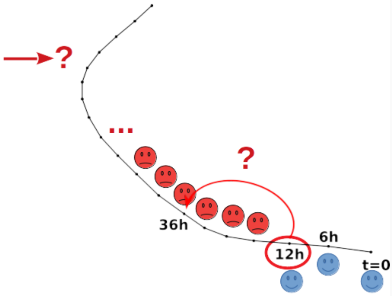
Data from previous steps

	stormid	instant_t	features
Storm #1	1982011S12119	0	...
	1982011S12119	1	...
	1982011S12119	2	...
	1982011S12119	3	...
	1982011S12119	4	...
	1982011S12119	5	...
	...	...	...
Storm #n	2014210N10323	0	...
	2014210N10323	1	...
	2014210N10323	2	...
	2014210N10323	3	...



Data from past steps

	stormid	instant_t	features
Storm #1	1982011S12119	0	...
	1982011S12119	1	...
	1982011S12119	2	...
	1982011S12119	3	...
	1982011S12119	4	...
	1982011S12119	5	...
	...	...	...
Storm #n	2014210N10323	0	...
	2014210N10323	1	...
	2014210N10323	2	...
	2014210N10323	3	...



Submissions

- Description:
  `http://github.com/ramp-kits/storm_forecast`
- Submissions:
  `www.ramp.studio/problems/storm_forecast`
- Available on the server: scikitlearn, pytorch, keras ... look at `ami_environment.yml` file (only CPUs)

Your turn to work!

- Team building
- `http://github.com/ramp-kits/storm_forecast` : notebook .ipynb for all informations
- clone starting kit (git) / download data
- `www.ramp.studio`: create account! (wait for access)
- `www.ramp.studio/problems/storm_forecast` : register on the project!
- Submit to the 'sandbox' (copy/paste your code)

julien.brajard@nersc.no