Introduction to optimization and machine learning

Lesson 1 : Introduction

Sébastien Verel

Laboratoire d'Informatique, Signal et Image de la Côte d'opale (LISIC)
Université du Littoral Côte d'Opale, Calais, France
http://www-lisic.univ-littoral.fr/~verel/

March, 2023

General outline

- Introduction to optimization problems
- Introduction to machine learning
- Fundamentals of optimization methods
- Fundamentals of machine learning methods
- Practice of some algorithms using python

Outline of the day

- Definition of optimization problems
- Definition of learning problems
- Optimization *vs.* machine learning

Artificial intelligence

AI = Learning and Reasoning

Main topics [1]

- Problem-solving
- Knowledge, reasoning, and planning
- Uncertain knowledge and reasoning
- Machine Learning
- Communicating, perceiving, and acting

[1] Artificial Intelligence : A Modern Approach, Fourth edition, 2020, Stuart Russell and Peter Norvig.

Artificial intelligence

AI = Learning and Reasoning

Main topics [1]

- Problem-solving
- Knowledge, reasoning, and planning
- Uncertain knowledge and reasoning
- Machine Learning
- Communicating, perceiving, and acting

And it is not only computer science, nor mathematics research field

[1] Artificial Intelligence : A Modern Approach, Fourth edition, 2020, Stuart Russell and Peter Norvig.

Problem Solving using optimization

A lot of problems consists of finding a "good" solution(s) using limited/comptable ressources

Real-world problems

Example of real-world problem

Products are in a depot

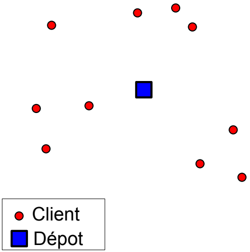
Goal : Deliver products to customers

Real-world problems

Example of real-world problem

Products are in a depot

Goal : Deliver products to customers



Abstract problem

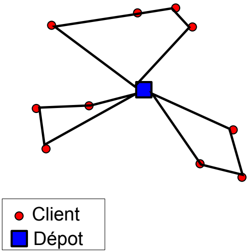
Minimize the travel distance (cost) with respect to contraints

Real-world problems

Example of real-world problem

Products are in a depot

Goal : Deliver products to customers



Abstract problem

Minimize the travel distance (cost) with respect to contraints

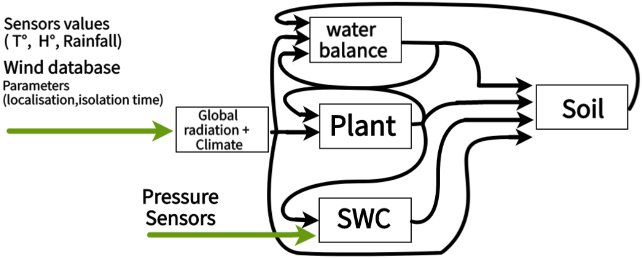
Another example [A. Dubois, F. Teytaud, E. Ramat]

Farming irrigation model for potatoes crop cultivation [a]

- Goal :
  Decision support provided to farmers
  to manage their irrigation plans
- How :
  Combination of several biological computational models

a. In collaboration with the *Weenat* company, PhD thesis of A. Dubois, 2020.

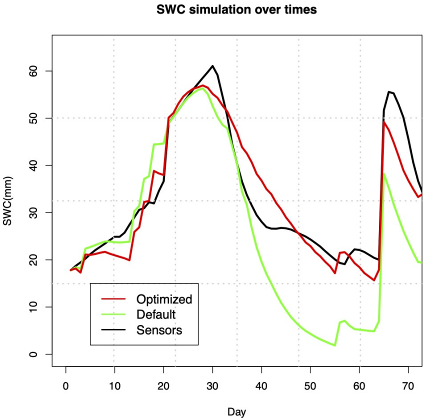
Computational model

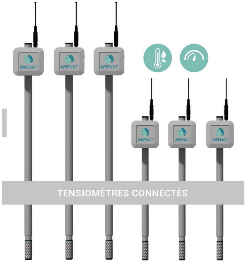


- Inputs : initial pressure, temperature, wind, rainfall, . . .
- Output : Soil Water Content, ...

Block model defined by 38 parameters (real values)

Soil Water Content predictions





Model calibration

- Minimize the distance
  between sensor data, and model simulation

Solving real-world problems

Throw this example, different elements are enlighted :

- Set of all candidate solutions :
  all possible travels
  all possible parameters values
- Cost function :
  travel distance
  distance data/model

The problem is solved
when a candidate solution with minimal cost is found (computed)

Single-objective optimization

Definition

An optimization problem is a couple $(\mathcal{X}, f)$ with :

- Search space : set of candidate solutions

$$\mathcal{X}$$

- Objective fonction : quality criteria (or non-quality)

$$f : \mathcal{X} \to \mathbb{R}$$

$\mathcal{X}$ discrete : combinatorial optimization

$\mathcal{X} \subset \mathbb{R}^n$: numerical optimization

Solve an optimization problem (minimization)

$$\mathcal{X}^\star = \text{argmin}_{\mathcal{X}} \; f$$

or find an approximation of $\mathcal{X}^\star$.

White-box optimization scenario

Objective function f for $x \in \mathbb{R}^d$,

$$f(x) = \frac{x_2^3 e^{-0.4 x_1}}{\sum_k e^{x_k}}$$

White-box optimization definition

Analytic expression of the objective function f is known

In this case, usually the objective fonction is :

- continuous, and differentiable (if we are lucky)

Black-box optimization scenario

$x \longrightarrow$  $\longrightarrow f(x)$

No information on the objective function definition f

Objective fonction :

- can be irregular, non continuous, non differentiable . . .
- given by a computation or a simulation

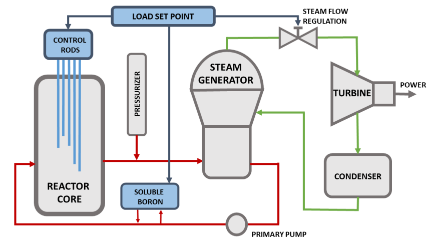
Real-world black-box optimization : an example

PhD of Valentin Drouet, Saclay Nuclear Research Centre (CEA), Paris

$x \longrightarrow$

$\longrightarrow f(x)$

$(73, \ldots, 8) \longrightarrow$



$\longrightarrow \Delta_z P$

Multi-physic simulator

Optimization problem solving

Solve an optimization problem (minimization)

$$\mathcal{X}^\star = \mathrm{argmin}_{\mathcal{X}} \; f$$

or find an approximation of $\mathcal{X}^\star$.

Multi-objective optimization

Definition

An optimization problem is a couple $(\mathcal{X}, f)$ with :

- Decision space : set of candidate solutions

$$\mathcal{X}$$

- Objective fonction : quality criteria (or non-quality)

$$f : \mathcal{X} \to \mathbb{R}^d$$

d : number of objective, criteria

$d = 2$: bi-objective optimization

$d = 3, 4, 5$: multi-objective optimization

$d > 5$: many-objective optimization

Solve an multi-objective optimization problem

Pareto optimal set : See later

Do you have an example of optimization problem ?

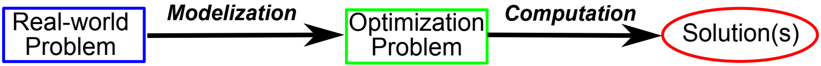
Exercice

Write examples of optimization problems
numerical, discrete, black/white-box, etc.

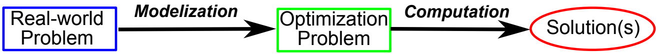
Solving methodology

2 steps :

- **Modelization** :
  Defined the optimization problem
- **Computation** :
  Compute an optimal solution (or near optimal)



Modelization



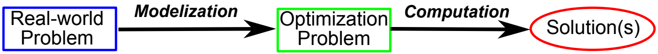
Definition : modelization

Transform a real-world problem
into an abstract optimization problem

Modelization

- Abstraction of the reality
- Simplification of the reality :
  number of parameters, noice, defaults, etc.
- Keep relevant elements with respect to problem to solve

Modelization



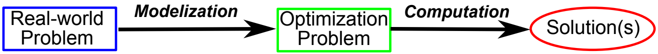
Design of (good) model

Difficult step, but with a good team of :

- Expert of the domain
- Expert in algorithms, abstract representation

it is a very powerful experience

Modelization



Design of (good) model

Difficult step, but with a good team of :

- Expert of the domain
- Expert in algorithms, abstract representation

it is a very powerful experience

Tools for designing a model (representation)

- Binary numbers, integer numbers, floating point numbers
- Combinatoric structure (vector, permutation, list, graph,...)
- Automata, abstract computing machines, etc.

Typology of optimization problems

Classification according to decision variables

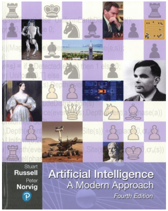
- **Combinatorial optimisation** :
  search space is discrete (sometime finite) : NP-hard
- **Numerical optimization** :
  search space is subset of $\mathbb{R}^d$
- **Others** :
  discrete and numerical, program, morphology, topology, etc.

Classification according to information

- **White-box optimisation** :
  Some useful properties are known
- **Black-box optimization** :
  A minimum of *a priori* information is used
  Computation time can be expensive (simulator, in vivo, etc.)
- **Grey-box optimization** : in between

Bibliography

Data Science : fondamentaux et études de cas, Machine Learning avec Python et R
Eric Biernat, Michel Lutz, Eyrolles, 2015.



Artificial Intelligence : A Modern Approach, Fourth edition, 2020, Stuart Russell and Peter Norvig.

Example

Problem

Predict the water in the ground

Problem

How to proceed ?

Machine Learning

"Slopy" definition

Study, and design of systems able to learn from data.
(system : computational methods on a computer)

Example

A system able to distinguish spam, and non-spam emails.

Machine Learning

E : set of all possible tasks.
S : a system (a machine)

A more formal definition [T.M. Mitchell, 1997]

$T \subset E$: set of tasks called *training set*
$P : \mathcal{S} \times E \to \mathbb{R}$: performance metric of a system on tasks.

A system S **learn** from an experience Exp if
the performance of S on tasks T, measured by P, is improving.

$$P(S_{\text{before Exp}}, T) \leq P(S_{\text{after Exp}}, T)$$

Example

Task T : Classifier of emails during one day
Performance P : rejection rate of spams by S
Experience Exp : 1 weak of emails from users

Learning from L. Valliant, 1984 [Turing award, 2010]

PAC ("Probably Approximately Correct")

In model of PAC Learning under the uniform distribution on X, a learning problem is defined with a concept class $\mathcal{C}$, which is just a collection of functions $f : X \to \mathbb{R}$; *"We learn a class $\mathcal{C}$ of functions"*.

A learning algorithm A for $\mathcal{C}$ is a randomized algorithm which has limited access to an unknown target function $f \in \mathcal{C}$.
The two access models are :

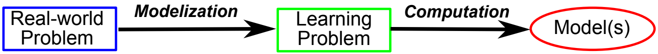
- random : A can draw pairs $(x, f(x))$ where $x \in X$ is uniformly random
- queries : A can request the value $f(x)$ for any $x \in X$ of its choice.

A is given as input an accuracy parameter $\epsilon \in [0, 1/2]$.
Output of A : a hypothesis function $h : X \to \mathbb{R}$.

PAC learning

A learns $\mathcal{C}$ with error ϵ if for any $f \in \mathcal{C}$, with high probability,
A outputs an h which is ϵ-close to f : $\text{dist}(f, h) \leq \epsilon$.

Modelization



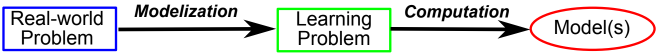
Definition : modelization

Transform a real-world problem
into an abstract learning problem

Modelization

- Abstraction of the reality
- Simplification of the reality :
  number of parameters, noice, defaults, etc.
- Keep relevant elements with respect to problem to learn

Modelization



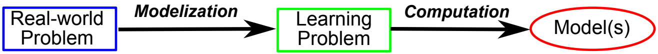
Design of (good) model

Difficult step, but with a good team of :

- Expert of the domain
- Expert in algorithms, abstract representation

it is a very powerful experience

Modelization



Design of (good) model

Difficult step, but with a good team of :

- Expert of the domain
- Expert in algorithms, abstract representation

it is a very powerful experience

Tools for designing a model (representation)

- Binary numbers, integer numbers, floating point numbers
- Combinatoric structure (vector, permutation, list, graph,...)
- Automata, abstract computing machines, etc.

Learn from Data

To learn with a computer,
we need some information, in particular **data**

Definition

Data : "The result of an observation on a population, or a sample"

Statistic, dictionnaire encyclopédique, Springer [Dodge, 2007]

A data is **number**, or a **feature** which gives an **information** on an individual, an object, or an observation.

Example

Sébastien : "I am 10 year old."

Variable

Link between one variable et data :

The features fluctuate according to the individual/object.

Notations :

- Variable X_j
- For the individual/object/observation i : X_{ij}.

Variable X_{age} for the individuals $1, 2, \ldots$: $X_{1age}, X_{2age}, \ldots$

Data type

- **Quantitative** data

  mesurable quantity, answer to "how much ?"

  allow computation (mean, etc.),

  comparaisons (equality, difference, inferior/superior)

  - Numerical : $\in \mathbb{R}$
  - Discrete : number of values are limited

- **Qualitative** data

  quality or features

  answer to the "category"

  - Nominale (categorial), ex : eyes color

    comparison (equality / difference)

  - Ordinal

    Order between elements (degree to test, etc.)

    comparison : superior / inférior

- **Structured** data

  relations, etc.

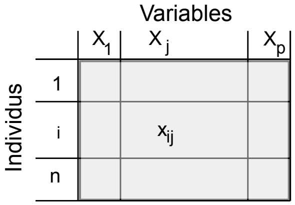
  - Tree, graph, complex data, etc.

Matrice representation of data
(expect structured data)

Several variables $X_1, X_2, \ldots, X_j$ for j de 1 à p
represent features of individual/objets/observations.
Number of individuals i from 1 to n.

Value of variable j for the individual i is denoted by x_{ij}

$$X = \begin{pmatrix} x_{11} & \cdots & x_{1p} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{np} \end{pmatrix}$$

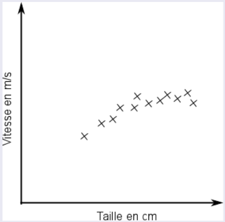


Typology according to available information

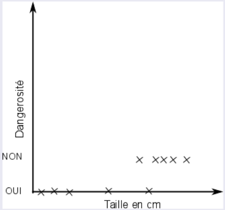
- Supervised learning :
  Learn from a set of examples :
  $\{(x_i, y_i) \mid i \in 1..n\}$
- Non-supervised learning :
  Learn from a set of example without labels (cf. clustering)
  $\{x_i \mid i \in 1..n\}$
- Semi-supervised learning :
  Learn from a set of examples with, and without labels
- Reinforcement learning :
  Learn when the actions on environment
  are rewarded by a score
- ...

Typology according to data

- Regression : (x_i, y_i) with $y_i \in \mathbb{R}$



- Classification : (x_i, y_i) with y_i discrete



Do you have an example of learning problem ?

Exercice

Write examples of learning problem

regression, classification, supervised, non-supervised, etc.

Machine Learning and Optimization

Similarities, and differences between ML and optimization ?

Machine Learning and Optimization

Similarities, and differences between ML and optimization ?

Machine learning, optimization share many things :

- Abstract representation of real-world
- Information processing
- Data guided methods
- ...

Machine learning, optimization are different :

- Learn : model of data
- Optimization : solution from a set of possible ones

Machine Learning and Optimization

Similarities, and differences between ML and optimization ?

Machine learning, optimization share many things :

- Abstract representation of real-world
- Information processing
- Data guided methods
- ...

Machine learning, optimization are different :

- Learn : model of data
- Optimization : solution from a set of possible ones

But what is the difference between a model, and a solution...

AI : Machine Learning, Optimization, perception, etc.

Learning :

Minimize an error function

$\{M_\theta\}$: models to learn on data

Search $\theta^\star = \arg\min_\theta Error(M_\theta, data)$

According to the model dimension, variables, error function, etc., huge number of optimization algorithms

Optimization :

Learn a design algorithm

$\{A_\theta\}$: search algorithms for problems (X, f)

Learn $\theta^\star$ such that $x = A_{\theta^\star}(X, f)$ is a good solution

According to the class of algorithms, search spaces, functions, etc., huge number of learning algorithms