Optimization for Data Science

F. Rinaldi[1]

[1]  Università degli Studi di Padova | Dipartimento Matematica

Padova
2020

Outline

Optimization for Data Science

Basic Course Info

Here are some Info

- Instructor: Francesco Rinaldi
- **Aim of the course:** Understanding optimization models and methods for Data Science.
- **Course structure:** (Kaltura based) Lectures posted on Moodle
  Zoom meetings (twice a week) for questions
- **(Kaltura based) Lectures:** available on Tue 16:30 and Thu 14:30
- **Zoom lecture meetings:** Tue (16:30-18:30) - Thu (14:30-16:30)
- **Office hours:** Zoom meeting under request - Please send an email to `rinaldi@math.unipd.it`

Exam

Main steps:

- **Written exam**: 5 open questions (85% of the grade)
- **Homeworks**: 2 homeworks (15% of the grade)
- Homeworks can be handled by groups of up to 4 students

Project (Optional):

- It can be requested to better analyze specific topics
- Theoretical/Computational analysis of one ore more papers
- Roughly Speaking: Understand the papers, write some code, test it on real instances, write your essay
- Project can be carried by groups of up to 4 students
- The project might integrate/replace the written exam

Outline of the course

Here are the main topics we will cover:

- Introduction
- Basic and useful notions
- A taste of convex analysis
- Methods for unconstrained optimization:
  - Gradient based methods
  - Coordinate descent methods
  - Stochastic methods
- Unconstrained optimization models:
  - e.g., Logistic regression, Boosting,...

Outline of the course II

And...

- Methods for constrained optimization:
  - Projected gradient methods
  - Conditional gradient methods
  - Interior Point Method
  - Linear programming and the simplex method
- Constrained optimization models:
  - e.g., PageRank, SVMs, Portfolio optimization,...
- Distributed optimization

Data Science

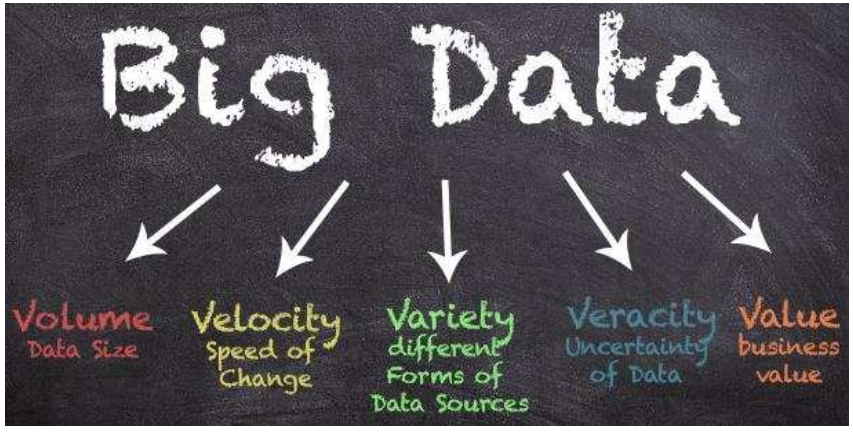
Data science is an interdisciplinary field that intersects ideas coming from

- applied mathematics,
- statistics,
- machine learning,
- computer programming,
- data engineering.

It mainly focuses on the extraction of knowledge from data, a task that is usually accomplished through the use of algorithms on what are often large or huge datasets.

Big Data

In industry, this trend has been also referred to as "Big Data"



Big Data Era

Big Data has had a significant impact in areas as varied as

- artificial intelligence,
- internet applications,
- computational biology,
- medicine,
- finance,
- marketing,
- network analysis.

All problems arising in different application domains anyway share some key features.

Common Features

Feature Big Data Problems share are the following:

- datasets are often extremely large, consisting of hundreds of millions or billions of training examples;
- the data is often very high-dimensional, because it is now possible to measure and store very detailed information about each example;
- the data is often stored (and sometimes collected) in a distributed manner (because of the large scale of many applications).

As a result, it has become crucial to develop algorithms that are both

- rich enough to capture the complexity of modern data,
- scalable enough to process huge datasets (eventually in a parallelized or fully decentralized fashion).

Mathematical Optimization

Mathematical optimization (or *mathematical programming*) is an important area of applied mathematics that deals, roughly speaking, with

the selection of a best element (with regard to some specific criterion) from some set of available alternatives.

Definition

an optimization problem consists of maximizing or minimizing a real function f by systematically choosing input values from within an allowed set C and computing the value of the function:

$$\min_{x \in C} f(x)$$

Goal of People Working in Optimization

Goal

Develop algorithms to efficiently solve a given optimization problem.

Data Science and Optimization

Data science and optimization are significantly intertwined:

- Data scientists make a wide use of optimization formulations and algorithms.
- Data science has contributed to optimization, driving the development of new optimization approaches (for huge scale applications).

This bond gets stronger and stronger, thus producing a growing literature at the intersection of the two fields while attracting leading researchers to the effort.

Why using Optimization?

Optimization approaches have enjoyed prominence in data science because of their wide applicability and attractive theoretical properties.

Keep in mind that

- Classic optimization mainly focuses on accuracy, speed, and robustness of the algorithms.
- Increased complexity, size, and variety of today's data science models requires a new fresh view of existing techniques.
- Accuracy and small speed improvements are of little concern in data science (we deal with huge scale problems here)

What Data scientists usually prefer

Simpler algorithms that work in reasonable computational time for specific classes of problems.

Convex Optimization Problems

In here, we will focus on **convex optimization** problems.

Main reason

Under minimal "computability assumptions", a convex optimization program is "computationally tractable".

Means that

the computational effort required to solve the problem to a given accuracy grows moderately with the dimensions of the problem and the required number of accuracy digits.

Non-convex Optimization Problems

Non-convex problems are too difficult to handle.

Non-convexity issues

Required computational effort grows prohibitively fast with the dimensions of the problem and the number of accuracy digits.

Non-convex is hard

there are theoretical reasons to guess that this is a specific feature of non-convex problems rather than a drawback of the existing optimization techniques.

The Importance of Convex Formulations

The importance of convex formulations increased in the last decade.

Reasons

Rise of new theory in signal analysis and statistical learning models (like, e.g., support vector machines).

New questions

We deal with increasingly large data sets and solve problems in unprecedented dimensions. Internet, text, and imaging problems no longer produce data sizes from megabytes to gigabytes, but rather from terabytes to exabytes.

The Importance of Models

Definition

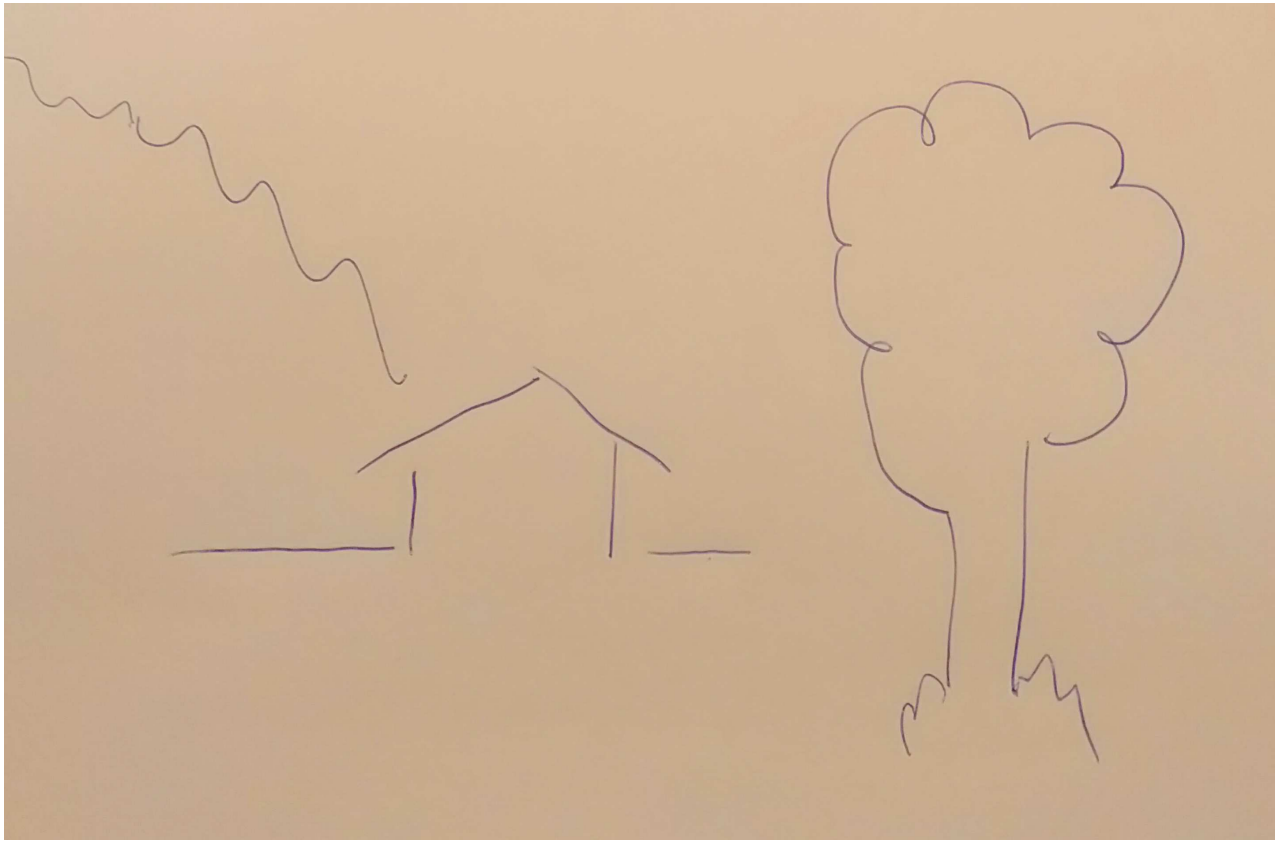
The term *model* is commonly used to indicate a specific structure that describes the main features of some real object or phenomenon.

Main features

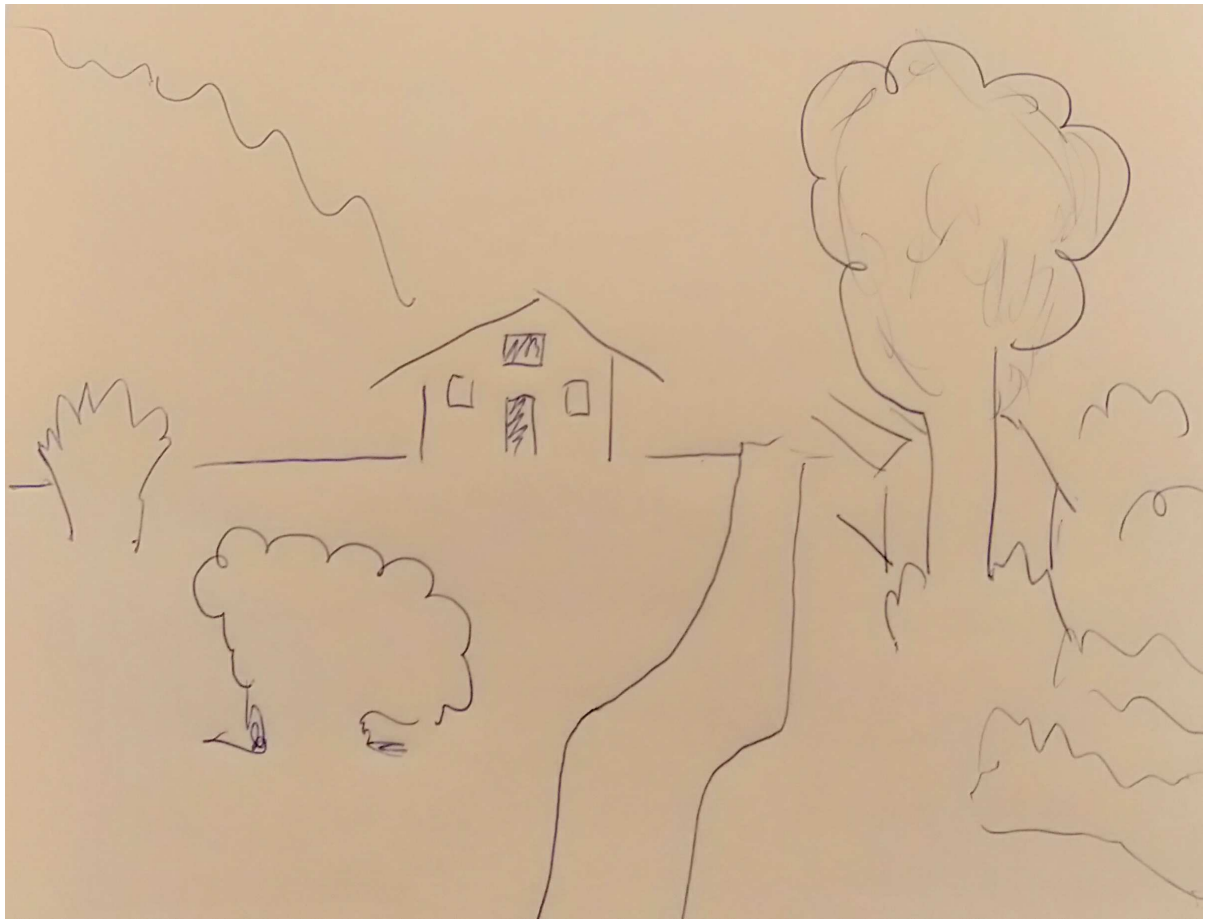
- A model is always an abstraction that is obviously simpler than the real situation.
- Elements that are irrelevant are to be ignored, hopefully leaving sufficient detail so that the model still represents the original problem.
- Models must hence be both tractable and valid (i.e., representative of the real phenomenon or system).

Real System: Abstraction and Interpretation

Simple Model



Complex Model



Real System: Abstraction and Interpretation

Abstraction

Taking most important components of real system and ignore less important components

Interpretation

Model components and model behavior can be related to components and behavior of real system

George Box Quote

All models are wrong...but some of them are useful! :-)

Mathematical Models

Here we focus on mathematical models (i.e., models composed of relationships and variables).

- *Relationships* can be described by operators (i.e., algebraic operators, functions, differential operators, etc.);
- *Variables* mainly represent the (quantifiable) parameters related to the system under analysis.

Mathematical Models Everywhere

Nowadays, mathematical models are widely used in many different areas:

- physics,
- engineering,
- social sciences,
- biology,
- economics,
- environmental science,
- psychology.

Remarkable diffusion of mathematical modeling mainly due to the fact that models can be easily *analyzed with modern computers*.

Conclusion

Data scientists need to be able to both build up good models and solve them by means of the appropriate algorithmic tools.

Model-based Approach

We then consider a *model-based approach*. This kind of approach to modeling consists of six different phases:

- Analysis of the real problem;
- Construction of the model;
- Analysis of the mathematical model;
- Choice of a solver;
- Model solution;
- Model validation.

Now we give some details about those phases.

Phase 1 to 3

First Phase (Analysis of the real problem)

We analyze the real problem and try to get the main relationships and variables that characterize it.

Second Phase (Construction of the model)

We build up our mathematical model and describe the best way we can all the details.

Third Phase (Analysis of the model)

After we build the model we need to analyze its mathematical properties. In particular,

- existence of solutions;
- conditions to identify an optimal solution;
- other mathematical features (so called global features).

Phase 4 to 6

Fourth Phase (Choice of solver)

Choose the algorithm that best fits the problem. This is a very important step for a data scientist. Indeed, we need to keep in mind that problems we want to solve are usually huge dimensional.

Fifth Phase (Model solution)

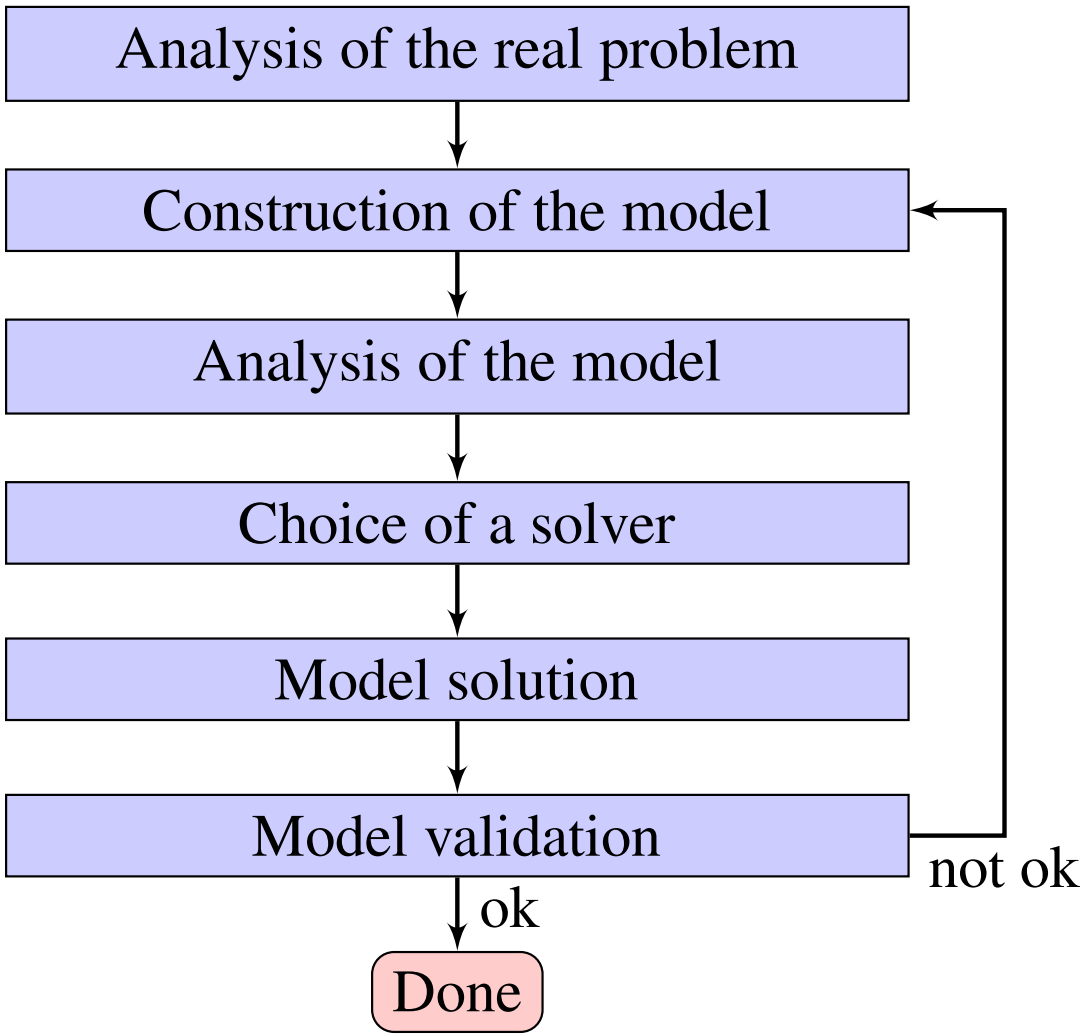
Once we choose the method that can be used to solve our problem, we actually solve it and find a solution.

Sixth Phase (Model validation)

We analyze in depth the solution in order to understand if it is useful or realistic (i.e., it can be implemented in practice).

- If this is the case, we are done.
- Otherwise, we need to properly modify the model and hence restart the process from **phase two**.

Model-based Approach Scheme



Choice of the Solver and the No Free Lunch Theorem

No free lunch theorem (Wolpert and Macready 1997)

All algorithms perform exactly the same when averaged over all possible problems. So, for any optimization algorithm, any elevated performance over one class of problems is exactly paid for in performance over another class.

Lesson learned

- Knowing that your model/problem has some kind of regularity property (e.g., convexity - we will better analyze this later on), hence belongs to some specific class, should help you choosing the algorithm.
- There is no universal algorithm which outperforms the other methods.
- Always care about the theoretical reasons why a given method is well adapted to some class of problems.

REMARK: Consider this theorem as a useful reminder against all *"my algorithm is always the best"* attitudes.