IG.2411 – AI and Optimization

General information

Title: AI and optimization
Module ID: IG.2411
Person in charge: Hélène URIEN
ECTS: 5
Average amount of work per student: from 100 to 150 hours (42 hours are supervised)
Teamwork: yes
Keywords: local and global optimization, AI, data analysis, descent direction algorithms, dynamic programming, gradient, Hessian, constraints, cross-validation, hyperparameter optimization

Presentation

Optimization is a branch of mathematics that allows minimize automatically a function defined on a specific domain, whether or not subject to constraints. There are many applications, in research as well as in industry, and require modeling a concrete problem as an optimization problem, and then choosing a solution method adapted to the specificities of the data. This module allows more particularly to understand the formalization of classical data analysis methods as an optimization problem

Educational objectives

At the end of the module, students will be able to:

- Analyze complex scientific problems
- Evaluate and interpret scientific results
- Formalize an optimization problem from a statement
- Choose a suitable method and setting for optimization

Prerequisite

- Algorithmics (II.1102 or II.1202)
- Data Science and Processing (IF.1105 / IF.1205)
- Basics in Linear Algebra

Content/Program

Concepts

- Static optimization without constraints (Nelder-Mead, Powell, gradient, Hessian, Jacobian, gradient descent, conjugate gradient, (quasi-) Newton method, Levenberg-Marquardt)
- Static optimization under constraints (Lagrange multipliers, KKT conditions, linear or nonlinear programming)
- Global static optimization (brute force, simulated annealing, genetic algorithms, ant colonies)
- Dynamic programming (formalism, optimality principle, examples of application algorithms such as Dijkstra or Bellman–Ford)
- Optimization for AI (differences between machine learning and pure optimization, formulation of data analysis methods as optimization problems, hyperparameter optimization strategy, stochastic gradient descent, and memory-limited BFGS)

Know-how

- Formalizing an optimization problem from a statement
- Choice of suitable method and parameterization for optimization and AI

- Evaluation and interpretation of scientific results
- Applications of optimization and AI in public works to real problems in image processing, in the medical field, in planning, etc.

Tools used

- Python libraries for optimization (Scipy) and for AI (Scikit-learn)

Pedagogical methods

Learning methods

- 20 hours of face-to-face lectures (10 x 2 hours).
- 22 hours of practical work to be done alone or in teams of 2-3 students, with or without report to be handed in at the end of the session or a few days later.
- Personal work will be necessary after each course, in particular to read the bibliography that complements each course material.

Evaluation methods

- Evaluation of practical reports: 50%
- Individual final exam at the end of the semester: 50%

Language of work

- English

Bibliography, Webography, Other sources

- Nocedal, J., & Wright, S. J. (Eds.). (1999). *Numerical optimization*. New York, NY: Springer New York, chapters 1, 3, 4, 5, 6, 7, 9, 10, 12, 13, 15, 16 and 18.
- Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (2007). *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge university press, chapter 10.
- Goodfellow, Y., Bengio, J., 2016, Courville, A., "Deep Learning" (2016), chapters 4, 5 and 8