INFO-F524 Project : Technical Description

Bernard Fortz et Dimitri Papadimitriou

Année académique: 2024-2025

1 Project Description

A startup specialized in optimization solvers wants to extend its activities to the domain of data analysis; more precisely, to the domain of regression analysis due to its ever-growing applicability in various domains.

To perform regression tasks, software available on the market relies, among others, on the Iterative Soft-Thresholding Algorithm (ISTA). The latter is a variant of the proximal gradient method for solving ℓ_1-regularized regression models a.k.a. LASSO [6] in the regression analysis jargon as well as $\ell_1-\ell_2$ regularized models a.k.a. elastic net [9], defined as a convex combination of the LASSO and Ridge regression (cf. Course notes - Section 2.6). To penetrate the market in regression analysis tools, the startup has recently learned that FISTA (Fast Iterative Shrinkage-Thresholding Algorithm), a variant of the proximal gradient method, could provide a significant competitive advantage both in terms of time performance and functionality (cf. Course notes, Section 2.6). FISTA offers the advantage of handling general composite (convex) optimization problems where one the terms of the objective function is not everywhere differentiable and in particular not differentiable at its minimizers (maximizers), i.e., this term is often referred to as "non-smooth".

However, the startup also knows that emphasizing only theoretical arguments (e.g., in terms of convergence rate) will not provide sufficiently convincing arguments in favor of FISTA to persuade its potential customers. Indeed, potential customers are not expert in optimization algorithms whereas classical methods and algorithms (such as LARS(-EN) [4]) to perform linear regression tasks are already included in statistical computing tools/frameworks such as R [7]. Hence, the startup must investigate linear regression problems involving representative datasets (of increasing scale) in order to verifiably showcase its newly developed regression analysis solver against ISTA. Moreover, scientifically analyzing and comparing the results of numerical experiments would certainly boost the adoption of the FISTA algorithm. It is also expected that, exploiting duality, the dual version of the FISTA algorithm (a.k.a. Dual-FISTA, see Course notes - Section 5.6) could help accelerate the computation of the primal optimal solution.

The startup also knows that handling more general composite optimization problems would provide a serious competitive advantage. In fact, composite optimization finds an applicability wider than solving least squares optimization problems with ℓ_1 norm regularization. In this case, one has to account that the local storage capacity would be limited since potential customers have indicated the use of FISTA-based solvers in embedded systems. Requiring the storage of entire libraries of formulas for computing the proximity operator of various convex functions could limit the applicability and thus the adoption of the solver. To circumvent this limitation, computing the proximity operator of smooth(ed) functions can be realized by means of limited-memory quasi-Newton methods, especially, the L-BFGS algorithm (cf. Course notes - Section 3.5.6). Even further, when the optimization problem is constrained, this computation could be performed by means of a projection method. In this context, the startup also aims to determine whether a hybrid strategy would not offer the best performance tradeoff.

2 Technical report

The project technical report must include the following sections.

1. A section motivating the selection and documenting the specification of
   - **both** ISTA and FISTA (and possibly variants of the latter).
   - the iterative algorithms used for performance comparison purposes: the gradient algorithm **and** either a quasi-Newton method (such as L-BFGS) or Dual-FISTA.
   - the line search method used for each of these algorithms.
2. A section documenting the complete design of the iterative solver that includes i) the input/output of each algorithm, ii) the function(s) developed and/or reused for the evaluation of the proximity operator (in case a library is used, document the calls), and iii) the algorithms, the functions, and the modules implemented including the line search routines. In this section, at least one diagram depicting the (modular) structure of the solver shall be included.
3. A section documenting the regularized regression models (at least ℓ_1, ℓ_2, and $\ell_1 - \ell_2$) and other composite optimization problems that the solver can handle together with a subsection that details the statistical properties of the data set considered for performance benchmark purposes. Data sets for (regularized) linear regression can be found on Kaggle `https://www.kaggle.com/datasets`; some examples are also included in the code section `https://www.kaggle.com/code`. Some of the references included in the bibliography section also provide links to various datasets. The selection of the appropriate data sets is an integral part of the project work.
4. A section that includes
   - The quality and the properties of the solution(s) obtained for each of these problems/datasets with respect to, e.g., different setting of the regularization parameter(s)
   - The following performance measures i) the execution time (min/max, average, variance/standard deviation), ii) the number of iterations (min/max, average, variance/standard deviation), and iii) possibly the memory space for each algorithm implemented by the solver. As each algorithm may involve subroutines (line search, computation of proximity operator, etc.) record these results separately too to identify which parts of the algorithm are the most time consuming.

- Comparative performance analysis obtained by executing these algorithms, including (at least one) measure of regression error over number of iterations.

5. A concluding section
6. Annex: dully documented code and other programs developed to realize this project must be included in Annex of the technical report.

3 Instructions

- The project MUST be realized by groups of 2 students.
- First and Last Name + Student ID for each group shall be provided by **March 15, 2025**. Any student unable to pair with a teammate should contact us (`mailto:dimitrios.papadimitriou@ulb.be`) urgently.
- The programming language shall be one of the following : C/C++, FORTRAN, Octave, Julia, Python. Groups selecting either Julia or Python shall also motivate the selection of linear algebra and other numerical computation libraries.
- The project technical report
  - must be written in `LATEX` and compiled in **.pdf** format. No other format will be accepted.
  - must include as header page, i) first and last name, ii) student ID and iii) study year of **each** member of the project group
  - must include **at least** 14 pages in single column (the documentclass `[a4paper,11pt]{article}` is recommended), excluding header page, bibliography and annexes. The maximum number of pages is not fixed, i.e., there is no upper limit to the number of pages. Nevertheless, the following ratios give rough indications for a well-balanced report : at most 1/3 of background, at least 1/2 of foreground, and the remainder accounts for introductory, concluding sections, etc.
  - must include dully documented code and other programs developed by the group and included in Annex of the technical report. Note well: verifiable proof that the code delivered is executable is required.
- Important instructions
  - There is no specific style imposed on the writing of the technical report as long as its content is scientifically and technically sound (this implies a precise and accurate description of the notations used throughout the technical report).
  - The evaluation of the report follows the grid (see `https://uv.ulb.ac.be/pluginfile.php/4054761/mod_assign/intro/Evaluation_grid_2024-25.pdf`) whose indicators are documented in `https://uv.ulb.ac.be/pluginfile.php/4054761/mod_assign/intro/Instructions_Project_2024-25.pdf`
  - The technical report must be delivered (as a single .pdf document) by **23-05-2025**.

- Exam (oral): The total duration of the oral exam is 1 hour. During each 1 hour time slot, the project group takes the oral exam (that is, the group that produced and delivered the project technical report passes the oral exam together). The oral exam does not involve the presentation of the project technical report.

  The oral exam comprises two distinct parts:

  - <u>Part 1</u> (duration : 30 min) : review of the project technical report and questions related to the content of the project technical report (see evaluation grid).
  - <u>Part 2</u> (duration : 30 min) : Questions related to the course material.

References

[1] Jean-Pierre Aubin and Ivar Ekeland, *Applied Nonlinear Analysis*, Dover Books on Mathematics, 1984.

[2] H. H. Bauschke and P. L. Combettes: *Convex Analysis and Monotone Operator Theory in Hilbert Spaces* (2nd Edition). Springer, New York, 2017.

[3] A. Beck, *First-Order Methods in Optimization*, MOS-SIAM Series on Optimization, 2017.

[4] Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R. (2004) Least angle regression. Ann. Statist., 32, 407–499.

[5] Boris S. Mordukhovich and Nguyen Mau Nam, *Convex Analysis and Beyond*, Volume I: Basic Theory, Springer International Publishing, 2022.

[6] Tibshirani, R. (1996) Regression shrinkage and selection via the lasso. J. R. Statist. Soc. B, 58, 267-288.

[7] The R Foundation https://www.r-project.org/foundation/

[8] R. Tyrrell Rockafellar and Roger J-B Wets, *Varational Analysis*, 1997, 2nd printing 2004, 3rd printing 2009, https://sites.math.washington.edu/~rtr/papers/rtr169-VarAnalysis-RockWets.pdf

[9] Hui Zou and Trevor Hastie, Regularization and variable selection via the elastic net, J. R. Statist. Soc. B, 67, Part 2, pp. 301-320, 2005.