Project Work in Optimization Methods/Optimization Techniques for Machine Learning

Lecturer: **Matteo Lapucci**
matteo.lapucci@unifi.it

July 31, 2023

1 Project Work general rules

1. Students (both from INM and IAM programs) interested in carrying out the 3CFU project on Optimization can contact (anytime) me to discuss possible assignments. It is not required that the student(s) have already passed the exam of the main Optimization course, however basic knowledge of nonlinear optimization and some in depth study of a subset of the course topics will clearly be necessary for the project work.

2. In general, the project work will consist of the implementation of some optimization algorithm and possibly the (re)production of some numerical results on given problem instances, or the conduction of a small experimentation.

3. The project work is 3CFU worth, i.e., 75 hours of work per person on average (if it is well calibrated).

4. The topic and the goals of the project work are established by the professor, together with the student(s), at the time of project assignment. Since it is not always easy to assess beforehand the workload associated with a project work, the goals of a project might occasionally be adjusted halfway (do not worry, I will not increase the workload, except for unlikely cases where students themselves are so interested in the topic to ask me to do so!).

5. There is no deadline, completion time is generally not a metric of evaluation; however, students are kindly encouraged to only ask for a project work to be assigned when they are actually willing to work on it.

6. Project works can be either carried out individually or in groups of two students; groups of three students can exceptionally be accepted only if there are substantial reasons to do so. Of course, the workload grows proportionally.

7. There is no grade associated with this module; evaluation is based on a quality assessment.

8. To conclude the assignment, a 3-5 pages report shall be set via email to the professor. The report should summarize all the work done, including mathematical formulations of problems and algorithms, relevant background theoretical results (if any), experiments description and results discussion. The report MUST be prepared with LaTex and sent in pdf format.

9. After the approval of the report, there will be a final, brief oral discussion of the entire project work. Slides for presentation are not required, but feel free to prepare them if you find them useful for the discussion.

2 Assignment

Student(s) Dragos Ionut Tanasa

Master's degree program Artificial Intelligence

Project topic Methods for ℓ_1-regularized logistic regression training

Topic description The use of the ℓ_1-regularizer in logistic regression has several beneficial effects, including the reduction of overfitting risk and solution sparsity, i.e., implicit feature selection [1,2]. The price to pay for this kind of regularization appears at an optimization level, as the objective function of the unconstrained optimization problem of fitting the logistic model now has a non-differentiable term. Several techniques can be employed to properly hand this irregular term; in fact, ℓ_1 regularization is handled within the LIBLINEAR software library, the state-of-the-art for linear models training.

The goal of the project is that of comparing two alternative approaches to solve the problem:

- Using the proximal gradient algorithm [1, Sec. 2.1.1]
- Solving the equivalent reformulation based on additional variables and non-negativity constraints [1, Sec. 2.1.4]

Project goals The student is asked to

- Implement (Python language, exploit `numpy` library) classes and functions allowing to load data to build an instance of ℓ_1-regularized logistic regression problem, compute the loss and its derivatives (see [2, Sec. 2.3]).
- Datasets for binary classification problems can be found, for instance at the page `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`; datasets in LIBSVM format can be easily loaded via the `sklearn.datasets.load_svmlight_file` module.
- Implement a proximal gradient solver [1, Sec. 2.1.1] for this class of problems.
- Implement the reformulated problem ([1, equation (7)]); exploit `scipy.optimize` module (use L-BFGS-B) to implement the solution procedure.
- Carry out experiments on various instances of the problem: use 6-8 different datasets and 3-5 different values for the regularization parameter λ. Compare the performance of the two algorithms in terms of runtime.
- Check the correctness of the implementation using as a reference the results of LIBLINEAR. LIBLINEAR can either be used via the official library `https://www.csie.ntu.edu.tw/~cjlin/liblinear/` or exploiting the interface provided by `sklearn.linear_model.LogisticRegression`. **Pay attention to the correct way of setting the bias and the regularization parameter.**

References [1] M. Lapucci, Ottimizzazione sparsa, lecture notes (2021)

[2] M. Lapucci, OTML Lecture Notes (2023)