# Project Work in Optimization Methods/Optimization Techniques for Machine Learning

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

February 9, 2024

## 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)** Gregorio Piqué

**Master's degree program** Artificial Intelligence

**Project topic** Almost multisecant quasi-Newton method

**Topic description** The key idea of Quasi-Newton methods is that of approximating the Hessian matrix, constructing a matrix that satisfies the Quasi-Newton equation for $x^{k+1}$ and $x^k$.

Multi-secant extensions of QN methods exploit the available knowledge of gradients at all previous iterates to construct a more accurate approximation of the Hessian. However, for general convex optimization problems, multi-secant quasi-Newton methods do not ensure that the obtained direction is of descent.

For this reason, in [1] an almost multi-secant approach is proposed, that enforces symmetry and positive definiteness of the approximated matrix $B_k$, at the cost of possibly losing the Quasi-Newton property of the updates. This approach apparently leads to more stable and effective versions of the multi-secant QN methods

**Project goals** The student is asked to

- Implement (in Python language, exploit `numpy` library) classes and functions allowing to load data to build instances of least squares linear regression and $\ell_2$-regularized logistic regression problems, computing the associated loss and the gradient functions (see Lecture notes, Sec. 2).
- Implement the following algorithms:
  - gradient descent;
  - BFGS;
  - Multi-Secant BFGS in the following variants:
    - multi-secant system is solved exactly, no symmetry nor positive definiteness enforced;
    - symmetry enforcing perturbation only;
    - positive-definiteness enforcing perturbation only;
    - both perturbations are employed - $B_k$ is symmetric positive definite.
- Identify a small set of real-world instances (2 datasets for linear regression and 2 datasets for binary classification), to carry out experiments.
- Test the efficiency of the considered methods on this benchmark of problems, measuring the progress of training loss w.r.t. iterations (see Figure 1 in [1]); show the same comparison w.r.t. runtime. Also, similarly as in Figure 2 in [1], measure the violation of the Quasi-Newton equation through iterations.
- **lecturer's note:** it seems to me that authors do not specify in [1] how the stepsize $\alpha$ is chosen at each iteration; from equation (2) in the paper, we might assume that they consider constant stepsizes, as there is no dependence on $k$ in the stepsize symbol (they write $\alpha$ and not $\alpha_k$); however, BFGS might incur in critical issues if Wolfe line search is not employed; on the other hand, Wolfe line search might not possess finite termination properties if $B_k$ is not symmetric positive definite and $d_k$ is possibly not a descent direction. I suggest to first try to implement the algorithms using a fixed stepsize; if things turn out not to work properly, the Wolfe line search (see Grippo and Sciandrone's book) shall be implemented and employed within BFGS and the almost multi-secant BFGS (the one with both perturbations).

**References**

[1] Lee, M., & Sun, Y. (2023, December). Almost multisecant BFGS quasi-Newton method. *In OPT 2023: Optimization for Machine Learning*, `https://opt-ml.org/papers/2023/paper86.pdf`.