Project Work in Optimization Methods/Optimization Techniques for Machine Learning

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

January 10, 2025

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.

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.

5. There is no deadline, however completion time will be one of the metrics of evaluation; students are encouraged to 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 numerical grade associated with this module; evaluation is based on a quality assessment. The assessment is mainly based on 1) correctness of the work, 2) independence in carrying out the job, 3) awareness of the topic and critical spirit towards the results and 4) time required to complete the assignment.

8. To conclude the assignment, a short report shall be sent 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) Carlo Lucchesi

Master's degree program Computer Engineering

Project topic Decomposition Algorithms for Linear SVMs

Topic description Decomposition methods addressing the dual formulation are the current standard to solve SVM training problems. While the SMO algorithm (see [1,2]) is in principle always employable, in the particular case of linear kernel SVMs another path proved to be better suited. This different approach exploits the structure of an unbiased formulation of the SVM problem, leading to a bound constrained dual. The new dual is separable an thus can be handled by a 1-variable update decomposition method (Dual Coordinate Descent) (see the discussion in [1, Sec. 7.3] and [3]). In the special case of linear kernels, the cost of the update is much lower than for SMO, resulting in an overall more efficient procedure - even if active variables are chosen randomly.

Project goals The student is asked to

- Implement (in Python language, exploit `numpy` library) classes and functions to build an instance of linear SVM training problems, and in particular the dual problem of both the biased and unbiased formulation. In the unbiased formulation, model the bias via an additional constant feature.
- Implement the SMO algorithm with the working set selection rule based on Most Violating Pairs. For the closed form solution of the subproblem, refer for example to [4, Appendix B].
- Implement the DCD algorithm proposed in [3].
- Identify a small benchmark of binary classification problems from the webpage `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/` to carry out experiments. Note that this dataset format can be easily read with the module `load_svmlight_file` from `scikit-learn` library.
- Verify the correctness of implementation of the two algorithms comparing the accuracy of the obtained models with those obtained using `LinearSVC` and `SVM` classes from `scikit-learn`.
- For all selected test problems, run the two algorithms using as a stopping condition a threshold on the runtime. Repeat each experiment with different thresholds. Measure the training loss attained by the model obtained for each algorithm and threshold. Finally compare the results of the two algorithms.

References

[1] "Official Lecture Notes of the course" (2024), M. Lapucci

[2] "Nonlinear optimization and support vector machines", *Annals of Operations Research* (2022), V. Piccialli, M. Sciandrone.

[3] "A Dual Coordinate Descent Method for Large-scale Linear SVM", *ICML* (2008), C-J Hsie et al..

[4] "Support Vector Machines - Lecture Notes" (2005), M. Sciandrone.