Optimal Computational Trade-Off of Inexact Proximal Methods (short version)

Pierre Machart, Luca Baldassarre, Sandrine Anthoine

Optimal Computational Trade-Off of Inexact Proximal Methods

Pierre Machart
LIF, LSIS, CNRS, Aix-Marseille University
pierre.machart@lif.univ-mrs.fr

Sandrine Anthoine
LATP, CNRS, Aix-Marseille University
anthoine@cmi.univ-mrs.fr

Luca Baldassarre
LIONS, École Polytechnique Fédérale de Lausanne
luca.baldassarre@epfl.ch

Abstract

In this paper, we investigate the trade-off between convergence rate and computational cost when minimizing a composite functional with proximal-gradient methods, which are popular optimisation tools in machine learning. We consider the case when the *proximity operator* is approximated via an iterative procedure, which yields algorithms with two nested loops. We show that the strategy minimizing the computational cost to reach a desired accuracy in finite time is to keep the number of inner iterations constant, which differs from the strategy indicated by a convergence rate analysis.

1 Introduction

Designing learning procedures which ensure good generalization properties is a central and recurring problem in Machine Learning. When dealing with *small-scale* problems, this issue is mainly covered by the traditional trade-off between *approximation* (i.e. considering the use of predictors that are complex enough to handle sophisticated prediction tasks) and *estimation* (i.e. considering the use of predictors that are simple enough not to overfit on the training data). However, when dealing with *large-scale* problems, the amount of available data can make it impossible to precisely solve for the optimal trade-off between approximation and estimation. Building on that observation, [3] has highlighted that *optimization* should be taken into account as a third crucial component, in addition to approximation and estimation, leading to more complex (multiple) trade-offs.

Meanwhile, the vast literature in optimization provides us with iterative algorithms to solve a broad class of problems. Over the last decades, constant efforts have been made to analyse the properties of the provided algorithms, noticeably their convergence properties. For instance, [6] has studied the complexity of various problems and given upper bounds on the maximal rates of convergence that optimization algorithms can achieve, for a given class of problems. Following this work, many efforts have been put into building algorithms with so-called *optimal convergence rates*.

In fact, this quest for optimal rates of convergence is quite relevant when one needs to solve problems (such as (1)) with an arbitrarily high precision (i.e. small-scale problems). However, dealing with the aforementioned multiple trade-off in a finite amount of computation time urges machine learners to consider solving problems with a lower precision and pay closer attention to the computational cost of the optimization procedures. The present paper precisely aims at providing an analysis of *inexact proximal-gradient* algorithms that takes their computational costs into account. We derive a strategy that directly minimizes the computational cost of the algorithm, under the constraint that problem (1) is approximated with some desired accuracy. We show that this new strategy is fundamentally different from those that achieve optimal convergence rates, as described in [7, 8].

2 Setting

Recent advances in machine learning and signal processing have led to more involved optimisation problems, while abundance of data calls for more efficient optimization algorithms. First-order methods are now extensively employed to tackle these issues and, among them, proximal-gradient algorithms [2] become increasingly popular. They solve very general convex non-smooth problems:

$$\min_x f(x) := g(x) + h(x), \tag{1}$$

where $g : \mathbb{R}^n \to \mathbb{R}$ is convex and smooth with a L-Lipschitz continuous gradient and $h : \mathbb{R}^n \to \mathbb{R}$ is lower semi-continuous proper convex. Let us denote by x^* the minimizer of problem (1).

Proximal-gradient algorithms consist in generating a sequence $\{x_k\}$, where

$$x_k = \text{prox}_{h/L}\left[x_{k-1} - \tfrac{1}{L}\nabla g(x_{k-1})\right], \text{ with } \text{prox}_{h/L}(z) = \operatorname*{argmin}_x \tfrac{L}{2}\|x - z\|^2 + h(x).$$

Classically, the *proximity operator* $\text{prox}_{h/L}$ is computed exactly. The sequence $\{x_k\}$ then converges to the solution of problem (1). However, in many situations no closed-form solution is known and one can only provide an approximation of the proximal point. Let us denote by ϵ_k an upper bound on the error induced in the proximal objective function by this approximation, at the k-th iteration:

$$\frac{L}{2}\|x_k - z\|^2 + h(x_k) \leq \epsilon_k + \min_x \left\{ \frac{L}{2}\|x - z\|^2 + h(x) \right\}. \tag{2}$$

The authors of [7] go beyond the study of the convergence of the inexact proximal method: they establish its rate of convergence.

Proposition 1 (Proximal-gradient method (Proposition 1 in [7]))**.** *For all $k \geq 1$,*

$$f(x_k) - f(x^*) \leq \frac{L}{2k}\left(\|x_0 - x^*\| + 2\sum_{i=1}^{k} \sqrt{\frac{2\epsilon_i}{L}} + \sqrt{\sum_{i=1}^{k} \frac{2\epsilon_i}{L}} \right)^2. \tag{3}$$

The approximation of the proximity operator yields two additional terms: the last two ones in (3). When the ϵ_i's are set to 0 (i.e. the proximity operator is computed exactly), one obtains the usual bound of the exact proximal method. These additional terms are summable if $\{\epsilon_k\}$ converges at least as fast as $O\left(\frac{1}{k^{(2+\delta)}}\right)$, for any $\delta > 0$. One direct consequence is that the optimal convergence rate in the error-free setting is still achievable, with such conditions on the $\{\epsilon_k\}$'s. However, [7] empirically notices that imposing too fast a decrease rate on $\{\epsilon_k\}$ is computationally counter-productive, as the precision required on the proximal approximation becomes computationally demanding. In other words, there is a subtle trade-off between the number of iterations needed to reach a certain solution and the cost of those iterations. This is the object of study of the present paper.

3 Defining the Problem

The main contribution of this paper is to define a *computationally optimal* way of setting the trade-off between the number of iterations and their cost. We consider the case where the proximity operator is approximated *via* an iterative procedure. The global algorithm thus consists in an iterative proximal method, where at each (outer-)iteration, one performs (inner-)iterations.

As stated earlier, our goal is to take into account the complexity of the global cost of inexact proximal methods. Using iterative procedures to estimate the proximity operator at each step, it is possible to formally express this cost. Let us assume that each inner-iteration has a constant computational cost C_{in} and that, in addition to the cost induced by the inner-iterations, each outer-iteration has a constant computational cost C_{out}. Let k be the number of outer iterations and l_i denote the number of inner iterations performed at the i-th iteration of the outer-loop. It immediately follows that the global cost of the algorithm is:

$$C_{\text{glob}}(k, \{l_i\}_{i=1}^k) = C_{\text{in}} \sum_{i=1}^{k} l_i + kC_{\text{out}}, \tag{4}$$

and the question we are interested in is to minimize this cost. In order to formulate our problem as a minimization of this cost, subject to some guarantees on the global precision of the solution, we now need to relate the number of inner iterations to the precision of the proximal point estimation.

Classical methods to approximate the proximity operator achieve *sublinear rates* of the form $O\left(\frac{1}{k^\alpha}\right)$ ($\alpha = \frac{1}{2}$ for sub-gradient or stochastic gradient descent; $\alpha = 1$ for gradient and proximal descent or $\alpha = 2$ for accelerated descent/proximal schemes). Let A_i be a positive constant. The error ϵ_i defined in (2) is thus bounded by:

$$\epsilon_i = \frac{A_i}{l_i^\alpha}, \tag{5}$$

Plugging (5) into (3), we can get the following bound:

$$f(x_k) - f(x^*) \leq \frac{L}{2k}\left(\|x_0 - x^*\| + 3\sum_{i=1}^{k}\sqrt{\frac{2A_i}{Ll_i^\alpha}}\right)^2 =: B(k, \{l_i\}_{i=1}^k).$$

This bound highlights the aforementioned trade-off. To achieve a fixed global error $\rho = f(x_k) - f(x^*)$, there is a natural trade-off that needs to be set by the user, between the number k of outer-iterations and the numbers of inner-iterations $\{l_i\}_{i=1}^k$, which can be seen as hyper-parameters of the global algorithm. As mentioned earlier, and witnessed in [7] the choice of those parameters will have a crucial impact on the computational efficiency (see equation (4)) of the algorithm.

Our aim to "optimally" set the hyper-parameters (k and $\{l_i\}_{i=1}^k$) may be conveyed by the following optimization problem. Given an accuracy ρ, we want to minimize the global cost C_{glob} of the algorithm, under the constraint that our bound on the error B is smaller than ρ:

$$\min_{k\in\mathbb{N},\{l_i\}_{i=1}^k\in\mathbb{N}^k} C_{\text{in}}\sum_{i=1}^{k} l_i + kC_{\text{out}} \qquad \text{s.t. } B(k, \{l_i\}_{i=1}^k) \leq \rho. \tag{6}$$

The rest of the paper is devoted to this optimisation problem.

4 Results

Problem (6) is an integer optimization problem as the variables of interest are numbers of (inner and outer) iterations. As such, this is a complex (NP-hard) problem and one cannot find a closed form for the integer solution, but if we relax our problem in l_i to a continuous one:

$$\min_{k\in\mathbb{N},\{l_i\}_{i=1}^k\in[1,\infty)^k} C_{\text{in}}\sum_{i=1}^{k} l_i + kC_{\text{out}} \qquad \text{s.t. } B(k, \{l_i\}_{i=1}^k) \leq \rho, \tag{7}$$

It is actually possible to find an *analytic expression* of the optimal $\{l_i\}_{i=1}^k$ and to numerically find the optimal k:

Proposition 2. *Let* $C(k) = \frac{\sqrt{L}}{3\sqrt{2A}}\left(\sqrt{\frac{2k\rho}{L}} - \|x_0 - x^*\|\right)$. *If* $\rho < 6\sqrt{2LA}\|x_0 - x^*\|$, *the solution of problem (6) is:*

$$\forall\, i, l_i^* = \left(\frac{C(k^*)}{k^*}\right)^{-\frac{2}{\alpha}}, \text{ with } k^* = \operatorname*{argmin}_{k\in\mathbb{N}^*} kC_{in}\left(\frac{C(k)}{k}\right)^{-\frac{2}{\alpha}} + kC_{out}. \tag{8}$$

Sketch of proof. First note that:

$$\min_{k,\{l_i\}_{i=1}^k} C_{\text{in}}\sum_{i=1}^{k} l_i + kC_{\text{out}} = \min_k \min_{\{l_i\}_{i=1}^k} C_{\text{in}}\sum_{i=1}^{k} l_i + kC_{\text{out}}.$$

We can solve problem (6) by first solving, for any k, the minimization problem over $\{l_i\}_{i=1}^k$. This is done using the standard Karush-Kuhn-Tucker approach [4]. Plugging the analytic expression of those optimal $\{l_i^*\}_{i=1}^k$ into our functional, we get our problem in k. □

In [5], we provide similar results for other widely used versions of proximal methods (e.g. accelerated version [2]) as well as numerical experiments.

5 Discussion and Conclusion

Our theoretical result urges to use a constant number of inner iterations. Coincidentally, many actual efficient implementations of such two nested algorithms, in [1] or in packages like SLEP[1] or PQN[2], use these constant number schemes. However, the theoretical grounds for such an implementation choice were not explicit. Our result can give some deeper understanding on why and how those practical implementations perform well. They also help acknowledging that the computation gain comes at the cost of an intrinsic limitation to the precision of the obtained solution.

The original motivation of this study is to show how, in the inexact proximal methods setting, optimization strategies that are the most computationally efficient, given some desired accuracy ρ, are *fundamentally different* from those that achieve optimal convergence rates.

The computationally-optimal strategy imposes constant number of inner iterations. Given our parametrisation, Eq. (5), this also means that the errors ϵ_i on the proximal computation remains constant. On the opposite, the optimal convergence rates can only be achieved for sequences of ϵ_i decreasing strictly faster than $1/i^2$. Obviously, the optimal convergence rates strategies also yield a bound on the minimal number of outer iterations needed to reach precision ρ by inverting the bound (3). However, this strategy is provably less efficient (computationally-wise) than the optimal one we have derived.

In fact, the pivotal difference between "optimal convergence rates" and "computationally optimal" strategies lies in the fact that the former ones arise from an asymptotic analysis while the latter arise from a finite-time analysis. While the former ensures that the optimization procedure will converge to the optimum of the problem (with optimal rates in the worst case), the latter only ensures that after k^* iterations, the solution found by the algorithm is not further than ρ from the optimum.

To conclude, we analysed the computational cost of proximal-gradient methods when the proximity operator is computed numerically. Building upon the results in [7], we proved that the optimization strategies, using a constant number of inner iterations, significantly impact the computational efficiency, at the cost of obtaining only a suboptimal solution.

Finally, although we focused on inexact proximal-gradient methods, the present work was inspired by the paper "The Trade-offs of Large-Scale Learning" [3]. Bottou and Bousquet studied the trade-offs between computational accuracy and statistical performance of machine learning methods and advocate for sacrificing the rate of convergence of optimization algorithms in favour of lighter computational costs. At a higher-level, future work naturally includes finding other situations where such trade-offs appear and analyse them using a similar methodology.

References

[1] S. Anthoine, J.-F. Aujol, C. Mélot, and Y. Boursier. Some proximal methods for cbct and pet tomography. inverse problems in imaging. Accepted to Inverse Problems and Imaging, 2012.

[2] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[3] L. Bottou and O. Bousquet. The trade-offs of large scale learning. In *Adv. in Neural Information Processing Systems (NIPS)*, 2007.

[4] H. W. Kuhn and A. W. Tucker. Nonlinear programming. In *Proc. of the Second Berkeley Symposium on Mathematical Statistics and Probability, 1950*, pages 481–492. University of California Press, 1951.

[5] P. Machart, S. Anthoine, and L. Baldassarre. Optimal Computational Trade-Off of Inexact Proximal Methods. Technical report, 2012.

[6] A.S. Nemirovsky and D.B. Yudin. *Problem complexity and method efficiency in optimization.* Wiley-Interscience Series in Discrete Mathematics. Johon Wiley & Sons, New York, 1983.

[7] M. Schmidt, N. Le Roux, and F. Bach. Convergence rates of inexact proximal-gradient methods for convex optimization. In *Adv. in Neural Information Processing Systems (NIPS)*, 2011.

[8] S. Villa, S. Salzo, L. Baldassarre, and A. Verri. Accelerated and inexact forward-backward algorithms. Technical report, Optimization Online, 2011.

[1] http://www.public.asu.edu/ jye02/Software/SLEP/index.htm

[2] http://www.di.ens.fr/ mschmidt/Software/PQN.html