Journal of Machine Learning Research 18 (2018) 1-54 Submitted 12/17; Revised 12/17; Published 4/18

Catalyst Acceleration for First-order Convex Optimization: from Theory to Practice

Hongzhou Lin HONGZHOU@MIT.EDU
Massachusetts Institute of Technology
Computer Science and Artificial Intelligence Laboratory
Cambridge, MA 02139, USA

Julien Mairal JULIEN.MAIRAL@INRIA.FR
Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK,*
Grenoble, 38000, France

Zaid Harchaoui ZAID@UW.EDU
University of Washington
Department of Statistics
Seattle, WA 98195, USA

Editor: Léon Bottou

Abstract

We introduce a generic scheme for accelerating gradient-based optimization methods in the sense of Nesterov. The approach, called Catalyst, builds upon the inexact accelerated proximal point algorithm for minimizing a convex objective function, and consists of approximately solving a sequence of well-chosen auxiliary problems, leading to faster convergence. One of the keys to achieve acceleration in theory and in practice is to solve these sub-problems with appropriate accuracy by using the right stopping criterion and the right warm-start strategy. We give practical guidelines to use Catalyst and present a comprehensive analysis of its global complexity. We show that Catalyst applies to a large class of algorithms, including gradient descent, block coordinate descent, incremental algorithms such as SAG, SAGA, SDCA, SVRG, MISO/Finito, and their proximal variants. For all of these methods, we establish faster rates using the Catalyst acceleration, for strongly convex and non-strongly convex objectives. We conclude with extensive experiments showing that acceleration is useful in practice, especially for ill-conditioned problems.

Keywords: convex optimization, first-order methods, large-scale machine learning

1. Introduction

A large number of machine learning and signal processing problems are formulated as the minimization of a convex objective function:

$$\min_{x \in \mathbb{R}^p} \left\{ f(x) \triangleq f_0(x) + \psi(x) \right\}, \tag{1}$$

where f_0 is convex and L-smooth, and ψ is convex but may not be differentiable. We call a function L-smooth when it is differentiable and its gradient is L-Lipschitz continuous.

[*] Institute of Engineering Univ. Grenoble Alpes

In statistics or machine learning, the variable x may represent model parameters, and the role of f_0 is to ensure that the estimated parameters fit some observed data. Specifically, f_0 is often a large sum of functions and (1) is a regularized empirical risk which writes as

$$\min_{x \in \mathbb{R}^p} \left\{ f(x) \triangleq \frac{1}{n} \sum_{i=1}^{n} f_i(x) + \psi(x) \right\}. \tag{2}$$

Each term $f_i(x)$ measures the fit between x and a data point indexed by i, whereas the function ψ acts as a regularizer; it is typically chosen to be the squared ℓ_2-norm, which is smooth, or to be a non-differentiable penalty such as the ℓ_1-norm or another sparsity-inducing norm (Bach et al., 2012).

We present a unified framework allowing one to accelerate gradient-based or first-order methods, with a particular focus on problems involving large sums of functions. By "accelerating", we mean generalizing a mechanism invented by Nesterov (1983) that improves the convergence rate of the gradient descent algorithm. When $\psi = 0$, gradient descent steps produce iterates $(x_k)_{k \geq 0}$ such that $f(x_k) - f^* \leq \varepsilon$ in $O(1/\varepsilon)$ iterations, where f^* denotes the minimum value of f. Furthermore, when the objective f is μ-strongly convex, the previous iteration-complexity becomes $O((L/\mu)\log(1/\varepsilon))$, which is proportional to the condition number L/μ. However, these rates were shown to be suboptimal for the class of first-order methods, and a simple strategy of taking the gradient step at a well-chosen point different from x_k yields the optimal complexity—$O(1/\sqrt{\varepsilon})$ for the convex case and $O(\sqrt{L/\mu}\log(1/\varepsilon))$ for the μ-strongly convex one (Nesterov, 1983). Later, this acceleration technique was extended to deal with non-differentiable penalties ψ for which the proximal operator defined below is easy to compute (Beck and Teboulle, 2009; Nesterov, 2013).

$$\text{prox}_{\psi}(x) \triangleq \arg\min_{z \in \mathbb{R}^p} \left\{ \psi(z) + \frac{1}{2}\|x - z\|^2 \right\}, \tag{3}$$

where $\|.\|$ denotes the Euclidean norm.

For machine learning problems involving a large sum of n functions, a recent effort has been devoted to developing fast incremental algorithms such as SAG (Schmidt et al., 2017), SAGA (Defazio et al., 2014a), SDCA (Shalev-Shwartz and Zhang, 2012), SVRG (Johnson and Zhang, 2013; Xiao and Zhang, 2014), or MISO/Finito (Mairal, 2015; Defazio et al., 2014b), which can exploit the particular structure (2). Unlike full gradient approaches, which require computing and averaging n gradients $(1/n)\sum_{i=1}^{n} \nabla f_i(x)$ at every iteration, incremental techniques have a cost per-iteration that is independent of n. The price to pay is the need to store a moderate amount of information regarding past iterates, but the benefits may be significant in terms of computational complexity. In order to achieve an ε-accurate solution for a μ-strongly convex objective, the number of gradient evaluations required by the methods mentioned above is bounded by $O\left(\left(n + \frac{\bar{L}}{\mu}\right)\log(\frac{1}{\varepsilon})\right)$, where $\bar{L}$ is either the maximum Lipschitz constant across the gradients ∇f_i, or the average value, depending on the algorithm variant considered. Unless there is a big mismatch between $\bar{L}$ and L (global Lipschitz constant for the sum of gradients), incremental approaches significantly outperform the full gradient method, whose complexity in terms of gradient evaluations is bounded by $O\left(n\frac{L}{\mu}\log(\frac{1}{\varepsilon})\right)$.

Yet, these incremental approaches do not use Nesterov's extrapolation steps and whether or not they could be accelerated was an important open question when these methods were introduced. It was indeed only known to be the case for SDCA (Shalev-Shwartz and Zhang, 2016) for strongly convex objectives. Later, other accelerated incremental algorithms were proposed such as Katyusha (Allen-Zhu, 2017), or the method of Lan and Zhou (2017).

We give here a positive answer to this open question. By analogy with substances that increase chemical reaction rates, we call our approach "Catalyst". Given an optimization method $\mathcal{M}$ as input, Catalyst outputs an accelerated version of it, eventually the same algorithm if the method $\mathcal{M}$ is already optimal. The sole requirement on the method in order to achieve acceleration is that it should have linear convergence rate for strongly convex problems. This is the case for full gradient methods (Beck and Teboulle, 2009; Nesterov, 2013) and block coordinate descent methods (Nesterov, 2012; Richtárik and Takáč, 2014), which already have well-known accelerated variants. More importantly, it also applies to the previou incremental methods, whose complexity is then bounded by $\tilde{O}\left(\left(n + \sqrt{n\bar{L}/\mu}\right)\log(\frac{1}{\varepsilon})\right)$ after Catalyst acceleration, where $\tilde{O}$ hides some logarithmic dependencies on the condition number $\bar{L}/\mu$. This improves upon the non-accelerated variants, when the condition number is larger than n. Besides, acceleration occurs regardless of the strong convexity of the objective—that is, even if $\mu = 0$—which brings us to our second achievement.

Some approaches such as MISO, SDCA, or SVRG are only defined for strongly convex objectives. A classical trick to apply them to general convex functions is to add a small regularization term $\varepsilon\|x\|^2$ in the objective (Shalev-Shwartz and Zhang, 2012). The drawback of this strategy is that it requires choosing in advance the parameter ε, which is related to the target accuracy. The approach we present here provides a *direct support for non-strongly convex objectives*, thus removing the need of selecting ε beforehand. Moreover, we can immediately establish a faster rate for the resulting algorithm. Finally, some methods such as MISO are numerically unstable when they are applied to strongly convex objective functions with small strong convexity constant. By defining better conditioned auxiliary subproblems, Catalyst also provides better numerical stability to these methods.

A short version of this paper has been published at the NIPS conference in 2015 (Lin et al., 2015a); in addition to simpler convergence proofs and more extensive numerical evaluation, we extend the conference paper with a new Moreau-Yosida smoothing interpretation with significant theoretical and practical consequences as well as new practical stopping criteria and warm-start strategies.

The paper is structured as follows. We complete this introductory section with some related work in Section 1.1, and give a short description of the two-loop Catalyst algorithm in Section 1.2. Then, Section 2 introduces the Moreau-Yosida smoothing and its inexact variant. In Section 3, we introduce formally the main algorithm, and its convergence analysis is presented in Section 4. Section 5 is devoted to numerical experiments and Section 6 concludes the paper.

1.1 Related Work

Catalyst can be interpreted as a variant of the proximal point algorithm (Rockafellar, 1976; Güler, 1991), which is a central concept in convex optimization, underlying augmented Lagrangian approaches, and composite minimization schemes (Bertsekas, 2015; Parikh and

Boyd, 2014). The proximal point algorithm consists of solving (1) by minimizing a sequence of auxiliary problems involving a quadratic regularization term. In general, these auxiliary problems cannot be solved with perfect accuracy, and several notions of inexactness were proposed by Güler (1992); He and Yuan (2012) and Salzo and Villa (2012). The Catalyst approach hinges upon (i) an acceleration technique for the proximal point algorithm originally introduced in the pioneer work of Güler (1992); (ii) a more practical inexactness criterion than those proposed in the past.[1] As a result, we are able to control the rate of convergence for approximately solving the auxiliary problems with an optimization method $\mathcal{M}$. In turn, we are also able to obtain the computational complexity of the global procedure, which was not possible with previous analysis (Güler, 1992; He and Yuan, 2012; Salzo and Villa, 2012). When instantiated in different first-order optimization settings, our analysis yields systematic acceleration.

Beyond Güler (1992), several works have inspired this work. In particular, accelerated SDCA (Shalev-Shwartz and Zhang, 2016) is an instance of an inexact accelerated proximal point algorithm, even though this was not explicitly stated in the original paper. Catalyst can be seen as a generalization of their algorithm, originally designed for stochastic dual coordinate ascent approaches. Yet their proof of convergence relies on different tools than ours. Specifically, we introduce an approximate sufficient descent condition, which, when satisfied, grants acceleration to any optimization method, whereas the direct proof of Shalev-Shwartz and Zhang (2016), in the context of SDCA, does not extend to non-strongly convex objectives. Another useful methodological contribution was the convergence analysis of inexact proximal gradient methods of Schmidt et al. (2011) and Devolder et al. (2014). Finally, similar ideas appeared in the independent work (Frostig et al., 2015). Their results partially overlap with ours, but the two papers adopt rather different directions. Our analysis is more general, covering both strongly-convex and non-strongly convex objectives, and comprises several variants including an almost parameter-free variant.

Then, beyond accelerated SDCA (Shalev-Shwartz and Zhang, 2016), other accelerated incremental methods have been proposed, such as APCG (Lin et al., 2015b), SDPC (Zhang and Xiao, 2015), RPDG (Lan and Zhou, 2017), Point-SAGA (Defazio, 2016) and Katyusha (Allen-Zhu, 2017). Their techniques are algorithm-specific and cannot be directly generalized into a unified scheme. However, we should mention that the complexity obtained by applying Catalyst acceleration to incremental methods matches the optimal bound up to a logarithmic factor, which may be the price to pay for a generic acceleration scheme.

A related recent line of work has also combined smoothing techniques with outer-loop algorithms such as Quasi-Newton methods (Themelis et al., 2016; Giselsson and Fält, 2016). Their purpose was not to accelerate existing techniques, but rather to derive new algorithms for nonsmooth optimization.

To conclude this survey, we mention the broad family of extrapolation methods (Sidi, 2017), which allow one to extrapolate to the limit sequences generated by iterative algorithms for various numerical analysis problems. Scieur et al. (2016) proposed such an approach for convex optimization problems with smooth and strongly convex objectives. The approach we present here allows us to obtain global complexity bounds for strongly

[1] Note that our inexact criterion was also studied, among others, by Salzo and Villa (2012), but their analysis led to the conjecture that this criterion was too weak to warrant acceleration. Our analysis refutes this conjecture.

Algorithm 1 Catalyst - Overview

```
input initial estimate $x_0$ in $\mathbb{R}^p$, smoothing parameter $\kappa$, optimization method $\mathcal{M}$.
Initialize $y_0 = x_0$.
while the desired accuracy is not achieved do
    Find $x_k$ using $\mathcal{M}$
        $x_k \approx \operatorname*{arg\,min}_{x \in \mathbb{R}^p} \left\{ h_k(x) \triangleq f(x) + \frac{\kappa}{2} \|x - y_{k-1}\|^2 \right\}.$    (4)
    Compute $y_k$ using an extrapolation step, with $\beta_k$ in $(0,1)$
        $y_k = x_k + \beta_k(x_k - x_{k-1}).$
end while
output $x_k$ (final estimate).
```

convex and non strongly convex objectives, which can be decomposed into a smooth part and a non-smooth proximal-friendly part.

1.2 Overview of Catalyst

Before introducing Catalyst precisely in Section 3, we give a quick overview of the algorithm and its main ideas. Catalyst is a generic approach that wraps an algorithm $\mathcal{M}$ into an accelerated one $\mathcal{A}$, in order to achieve the same accuracy as $\mathcal{M}$ with reduced computational complexity. The resulting method $\mathcal{A}$ is an inner-outer loop construct, presented in Algorithm 1, where in the *inner loop* the method $\mathcal{M}$ is called to solve an auxiliary strongly-convex optimization problem, and where in the *outer loop* the sequence of iterates produced by $\mathcal{M}$ are *extrapolated* for faster convergence. There are therefore three main ingredients in Catalyst: a) a smoothing technique that produces strongly-convex sub-problems; b) an extrapolation technique to accelerate the convergence; c) a balancing principle to optimally tune the inner and outer computations.

Smoothing by infimal convolution Catalyst can be used on any algorithm $\mathcal{M}$ that enjoys a linear-convergence guarantee when minimizing strongly-convex objectives. However the objective at hand may be poorly conditioned or even might not be strongly convex. In Catalyst, we use $\mathcal{M}$ to *approximately minimize* an auxiliary objective h_k at iteration k, defined in (4), which is strongly convex and better conditioned than f. Smoothing by infimal convolution allows one to build a well-conditioned convex function F from a poorly-conditioned convex function f (see Section 3 for a refresher on Moreau envelopes). We shall show in Section 3 that a notion of *approximate Moreau envelope* allows us to define precisely the information collected when approximately minimizing the auxiliary objective.

Extrapolation by Nesterov acceleration Catalyst uses an extrapolation scheme " à la Nesterov " to build a sequence $(y_k)_{k \geq 0}$ updated as

$$y_k = x_k + \beta_k(x_k - x_{k-1}) \ ,$$

where $(\beta_k)_{k \geq 0}$ is a positive decreasing sequence, which we shall define in Section 3.

	Without Catalyst		With Catalyst	
	$\mu > 0$	$\mu = 0$	$\mu > 0$	$\mu = 0$
FG	$O\left(n\frac{L}{\mu}\log\left(\frac{1}{\varepsilon}\right)\right)$	$O\left(n\frac{L}{\varepsilon}\right)$	$\tilde{O}\left(n\sqrt{\frac{L}{\mu}}\log\left(\frac{1}{\varepsilon}\right)\right)$	$\tilde{O}\left(n\sqrt{\frac{L}{\varepsilon}}\right)$
SAG/SAGA	$O\left(\left(n+\frac{\bar{L}}{\mu}\right)\log\left(\frac{1}{\varepsilon}\right)\right)$	$O\left(n\frac{\bar{L}}{\varepsilon}\right)$	$\tilde{O}\left(\left(n+\sqrt{\frac{n\bar{L}}{\mu}}\right)\log\left(\frac{1}{\varepsilon}\right)\right)$	$\tilde{O}\left(\sqrt{\frac{n\bar{L}}{\varepsilon}}\right)$
MISO		not avail.		
SDCA				
SVRG				
Acc-FG	$O\left(n\sqrt{\frac{L}{\mu}}\log\left(\frac{1}{\varepsilon}\right)\right)$	$O\left(n\frac{L}{\sqrt{\varepsilon}}\right)$	no acceleration	
Acc-SDCA	$\tilde{O}\left(\left(n+\sqrt{\frac{n\bar{L}}{\mu}}\right)\log\left(\frac{1}{\varepsilon}\right)\right)$	not avail.		

Table 1: Comparison of rates of convergence, before and after the Catalyst acceleration, in the strongly-convex and non strongly-convex cases, respectively. The notation $\tilde{O}$ hides logarithmic factors. The constant L is the global Lipschitz constant of the gradient's objective, while $\bar{L}$ is the average Lipschitz constants of the gradients ∇f_i, or the maximum value, depending on the algorithm's variants considered.

We shall show in Section 4 that we can get faster rates of convergence thanks to this extrapolation step when the smoothing parameter κ, the inner-loop stopping criterion, and the sequence $(\beta_k)_{k\geq 0}$ are carefully built.

Balancing inner and outer complexities The optimal balance between inner loop and outer loop complexity derives from the complexity bounds established in Section 4. Given an estimate about the condition number of f, our bounds dictate a choice of κ that gives the optimal setting for the inner-loop stopping criterion and all technical quantities involved in the algorithm. We shall demonstrate in particular the power of an appropriate warm-start strategy to achieve near-optimal complexity.

Overview of the complexity results Finally, we provide in Table 1 a brief overview of the complexity results obtained from the Catalyst acceleration, when applied to various optimization methods $\mathcal{M}$ for minimizing a large finite sum of n functions. Note that the complexity results obtained with Catalyst are optimal, up to some logarithmic factors (see Agarwal and Bottou, 2015; Arjevani and Shamir, 2016; Woodworth and Srebro, 2016).

2. The Moreau Envelope and its Approximate Variant

In this section, we recall a classical tool from convex analysis called the Moreau envelope or Moreau-Yosida smoothing (Moreau, 1962; Yosida, 1980), which plays a key role for understanding the Catalyst acceleration. This tool can be seen as a smoothing technique, which can turn any convex lower semicontinuous function f into a smooth function, and an ill-conditioned smooth convex function into a well-conditioned smooth convex function.

The Moreau envelope results from the infimal convolution of f with a quadratic penalty:

$$F(x) \triangleq \min_{z \in \mathbb{R}^p} \left\{ f(z) + \frac{\kappa}{2} \|z - x\|^2 \right\}, \tag{5}$$

where κ is a positive regularization parameter. The proximal operator is then the unique minimizer of the problem—that is,

$$p(x) \triangleq \text{prox}_{f/\kappa}(x) = \operatorname*{arg\,min}_{z \in \mathbb{R}^p} \left\{ f(z) + \frac{\kappa}{2} \|z - x\|^2 \right\}.$$

Note that $p(x)$ does not admit a closed form in general. Therefore, computing it requires to solve the sub-problem to high accuracy with some iterative algorithm.

2.1 Basic Properties of the Moreau Envelope

The smoothing effect of the Moreau regularization can be characterized by the next proposition (see Lemaréchal and Sagastizábal, 1997, for elementary proofs).

Proposition 1 (Regularization properties of the Moreau Envelope) *Given a convex continuous function f and a regularization parameter $\kappa > 0$, consider the Moreau envelope F defined in (5). Then,*

1. *F is convex and minimizing f and F are equivalent in the sense that*

$$\min_{x \in \mathbb{R}^p} F(x) = \min_{x \in \mathbb{R}^p} f(x) .$$

Moreover the solution set of the two above problems coincide with each other.

2. *F is continuously differentiable even when f is not and*

$$\nabla F(x) = \kappa(x - p(x)) . \tag{6}$$

Moreover the gradient ∇F is Lipschitz continuous with constant $L_F = \kappa$.

3. *If f is μ-strongly convex, then F is μ_F-strongly convex with $\mu_F = \frac{\mu\kappa}{\mu+\kappa}$.*

Interestingly, F is friendly from an optimization point of view as it is convex and differentiable. Besides, F is κ-smooth with condition number $\frac{\mu+\kappa}{\mu}$ when f is μ-strongly convex. Thus F can be made arbitrarily well conditioned by choosing a small κ. Since both functions f and F admit the same solutions, a naive approach to minimize a non-smooth function f is to first construct its Moreau envelope F and then apply a smooth optimization method on it. As we will see next, Catalyst can be seen as an accelerated gradient descent technique applied to F with inexact gradients.

2.2 A Fresh Look at Catalyst

First-order methods applied to F provide us several well-known algorithms.

The proximal point algorithm. Consider gradient descent steps on F:

$$x_{k+1} = x_k - \frac{1}{L_F} \nabla F(x_k).$$

By noticing that $\nabla F(x_k) = \kappa(x_k - p(x_k))$ and $L_f = \kappa$, we obtain in fact

$$x_{k+1} = p(x_k) = \arg\min_{z \in \mathbb{R}^p} \left\{ f(z) + \frac{\kappa}{2} \|z - x_k\|^2 \right\},$$

which is exactly the proximal point algorithm (Martinet, 1970; Rockafellar, 1976).

Accelerated proximal point algorithm. If gradient descent steps on F yields the proximal point algorithm, it is then natural to consider the following sequence

$$x_{k+1} = y_k - \frac{1}{L_F} \nabla F(y_k) \quad \text{and} \quad y_{k+1} = x_{k+1} + \beta_{k+1}(x_{k+1} - x_k),$$

where β_{k+1} is Nesterov's extrapolation parameter (Nesterov, 2004). Again, by using the closed form of the gradient, this is equivalent to the update

$$x_{k+1} = p(y_k) \quad \text{and} \quad y_{k+1} = x_{k+1} + \beta_{k+1}(x_{k+1} - x_k),$$

which is known as the accelerated proximal point algorithm of Güler (1992).

While these algorithms are conceptually elegant, they suffer from a major drawback in practice: each update requires to evaluate the proximal operator $p(x)$. Unless a closed form is available, which is almost never the case, we are not able to evaluate $p(x)$ exactly. Hence an iterative algorithm is required for each evaluation of the proximal operator which leads to the inner-outer construction (see Algorithm 1). Catalyst can then be interpreted as an accelerated proximal point algorithm that calls an optimization method $\mathcal{M}$ to compute inexact solutions to the sub-problems. The fact that such a strategy could be used to solve non-smooth optimization problems was well-known, but the fact that it could be used for acceleration is more surprising. The main challenge that will be addressed in Section 3 is how to control the complexity of the inner-loop minimization.

2.3 The Approximate Moreau Envelope

Since Catalyst uses inexact gradients of the Moreau envelope, we start with specifying the inexactness criteria.

Inexactness through absolute accuracy. Given a proximal center x, a smoothing parameter κ, and an accuracy $\varepsilon > 0$, we denote the set of ε-approximations of the proximal operator $p(x)$ by

$$p^{\varepsilon}(x) \triangleq \{z \in \mathbb{R}^p \quad \text{s.t.} \quad h(z) - h^* \leq \varepsilon\} \quad \text{where} \quad h(z) = f(z) + \frac{\kappa}{2}\|x - z\|^2, \tag{C1}$$

and h^* is the minimum function value of h.

Checking whether $h(z) - h^* \leq \varepsilon$ may be impactical since h^* is unknown in many situations. We may then replace h^* by a lower bound that can be computed more easily. We may use the Fenchel conjugate for instance. Then, given a point z and a lower-bound

$d(z) \leq h^*$, we can guarantee $z \in p^\varepsilon(x)$ if $h(z) - d(z) \leq \varepsilon$. There are other choices for the lower bounding function d which result from the specific construction of the optimization algorithm. For instance, dual type algorithms such as SDCA (Shalev-Shwartz and Zhang, 2012) or MISO (Mairal, 2015) maintain a lower bound along the iterations, allowing one to compute $h(z) - d(z) \leq \varepsilon$.

When none of the options mentioned above are available, we can use the following fact, based on the notion of gradient mapping; see Section 2.3.2 of (Nesterov, 2004). The intuition comes from the smooth case: when h is smooth, the strong convexity yields

$$h(z) - \frac{1}{2\kappa}\|\nabla h(z)\|^2 \leq h^*.$$

In other words, the norm of the gradient provides enough information to assess how far we are from the optimum. From this perspective, the gradient mapping can be seen as an extension of the gradient for the composite case where the objective decomposes as a sum of a smooth part and a non-smooth part (Nesterov, 2004).

Lemma 2 (Checking the absolute accuracy criterion) *Consider a proximal center x, a smoothing parameter κ and an accuracy $\varepsilon > 0$. Consider an objective with the composite form (1) and we set function h as*

$$h(z) = f(z) + \frac{\kappa}{2}\|x - z\|^2 = \underbrace{f_0(z) + \frac{\kappa}{2}\|x - z\|^2}_{\triangleq\, h_0} + \psi(x).$$

For any $z \in \mathbb{R}^p$, we define

$$[z]_\eta = \text{prox}_{\eta\psi}\left(z - \eta\nabla h_0\left(z\right)\right), \quad \textit{with} \quad \eta = \frac{1}{\kappa + L}. \tag{7}$$

Then, the gradient mapping of h at z is defined by $\frac{1}{\eta}(z - [z]_\eta)$ and

$$\frac{1}{\eta}\left\|z - [z]_\eta\right\| \leq \sqrt{2\kappa\varepsilon} \quad \textit{implies} \quad [z]_\eta \in p^\varepsilon(x).$$

The proof is given in Appendix B. The lemma shows that it is sufficient to check the norm of the gradient mapping to ensure condition (C1). However, this requires an additional full gradient step and proximal step at each iteration.

As soon as we have an approximate proximal operator z in $p^\varepsilon(x)$ in hand, we can define an approximate gradient of the Moreau envelope,

$$g(z) \triangleq \kappa(x - z), \tag{8}$$

by mimicking the exact gradient formula $\nabla F(x) = \kappa(x - p(x))$. As a consequence, we may immediately draw a link

$$z \in p^\varepsilon(x) \quad \Longrightarrow \quad \|z - p(x)\| \leq \sqrt{\frac{2\varepsilon}{\kappa}} \quad \Longleftrightarrow \quad \|g(z) - \nabla F(x)\| \leq \sqrt{2\kappa\varepsilon}, \tag{9}$$

where the first implication is a consequence of the strong convexity of h at its minimum $p(x)$. We will then apply the approximate gradient g instead of ∇F to build the inexact proximal point algorithm. Since the inexactness of the approximate gradient can be bounded by an absolute value $\sqrt{2\kappa\varepsilon}$, we call (C1) the absolute accuracy criterion.

Relative error criterion. Another natural way to bound the gradient approximation is by using a relative error, namely in the form $\|g(z) - \nabla F(x)\| \leq \delta' \|\nabla F(x)\|$ for some $\delta' > 0$. This leads us to the following inexactness criterion.

Given a proximal center x, a smoothing parameter κ and a relative accuracy δ in $[0, 1)$, we denote the set of δ-relative approximations by

$$g^\delta(x) \triangleq \left\{ z \in \mathbb{R}^p \quad \text{s.t.} \quad h(z) - h^* \leq \frac{\delta\kappa}{2}\|x - z\|^2 \right\}, \tag{C2}$$

At a first glance, we may interpret the criterion (C2) as (C1) by setting $\varepsilon = \frac{\delta\kappa}{2}\|x - z\|^2$. But we should then notice that the accuracy depends on the point z, which is is no longer an absolute constant. In other words, the accuracy varies from point to point, which is proportional to the squared distance between z and x. First one may wonder whether $g^\delta(x)$ is an empty set. Indeed, it is easy to see that $p(x) \in g^\delta(x)$ since $h(p(x)) - h^* = 0 \leq \frac{\delta\kappa}{2}\|x - p(x)\|^2$. Moreover, by continuity, $g^\delta(x)$ is closed set around $p(x)$. Then, by following similar steps as in (9), we have

$$z \in g^\delta(x) \quad \Longrightarrow \quad \|z - p(x)\| \leq \sqrt{\delta}\|x - z\| \leq \sqrt{\delta}(\|x - p(x)\| + \|p(x) - z\|).$$

By defining the approximate gradient in the same way $g(z) = \kappa(x - z)$ yields,

$$z \in g^\delta(x) \quad \Longrightarrow \quad \|g(z) - \nabla F(x)\| \leq \delta'\|\nabla F(x)\| \quad \text{with} \quad \delta' = \frac{\sqrt{\delta}}{1 - \sqrt{\delta}},$$

which is the desired relative gradient approximation.

Finally, the discussion about bounding $h(z) - h^*$ still holds here. In particular, Lemma 2 may be used by setting the value $\varepsilon = \frac{\delta\kappa}{2}\|x - z\|^2$. The price to pay is as an additional gradient step and an additional proximal step per iteration.

A few remarks on related works. Inexactness criteria with respect to subgradient norms have been investigated in the past, starting from the pioneer work of Rockafellar (1976) in the context of the inexact proximal point algorithm. Later, different works have been dedicated to more practical inexactness criteria (Auslender, 1987; Correa and Lemaréchal, 1993; Solodov and Svaiter, 2001; Fuentes et al., 2012). These criteria include duality gap, ε-subdifferential, or decrease in terms of function value. Here, we present a more intuitive point of view using the Moreau envelope.

While the proximal point algorithm has caught a lot of attention, very few works have focused on its accelerated variant. The first accelerated proximal point algorithm with inexact gradients was proposed by Güler (1992). Then, Salzo and Villa (2012) proposed a more rigorous convergence analysis, and more inexactness criteria, which are typically stronger than ours. In the same way, a more general inexact oracle framework has been proposed later by Devolder et al. (2014). To achieve the Catalyst acceleration, our main effort was to propose and analyze criteria that allow us to control the complexity for finding approximate solutions of the sub-problems.

3. Catalyst Acceleration

Catalyst is presented in Algorithm 2. As discussed in Section 2, this scheme can be interpreted as an inexact accelerated proximal point algorithm, or equivalently as an accelerated gradient descent method applied to the Moreau envelope of the objective with inexact gradients. Since an overview has already been presented in Section 1.2, we now present important details to obtain acceleration in theory and in practice.

```
Algorithm 2 Catalyst
input Initial estimate $x_0$ in $\mathbb{R}^p$, smoothing parameter $\kappa$, strong convexity parameter $\mu$,
    optimization method $\mathcal{M}$ and a stopping criterion based on a sequence of accuracies
    $(\varepsilon_k)_{k\geq 0}$, or $(\delta_k)_{k\geq 0}$, or a fixed budget $T$.
Initialize $y_0 = x_0$, $q = \frac{\mu}{\mu+\kappa}$. If $\mu > 0$, set $\alpha_0 = \sqrt{q}$, otherwise $\alpha_0 = 1$.
while the desired accuracy is not achieved do
    Compute an approximate solution of the following problem with $\mathcal{M}$

        $x_k \approx \arg\min_{x\in\mathbb{R}^p} \left\{ h_k(x) \triangleq f(x) + \frac{\kappa}{2}\|x - y_{k-1}\|^2 \right\},$

    using the warm-start strategy of Section 3 and one of the following stopping criteria:
        (a) absolute accuracy: find $x_k$ in $p^{\varepsilon_k}(y_{k-1})$ by using criterion (C1);
        (b) relative accuracy: find $x_k$ in $g^{\delta_k}(y_{k-1})$ by using criterion (C2);
        (c) fixed budget: run $\mathcal{M}$ for $T$ iterations and output $x_k$.
    Update $\alpha_k$ in $(0, 1)$ by solving the equation

        $\alpha_k^2 = (1 - \alpha_k)\alpha_{k-1}^2 + q\alpha_k.$    (10)

    Compute $y_k$ with Nesterov's extrapolation step

        $y_k = x_k + \beta_k(x_k - x_{k-1}) \quad \text{with} \quad \beta_k = \frac{\alpha_{k-1}(1 - \alpha_{k-1})}{\alpha_{k-1}^2 + \alpha_k}.$    (11)

end while
output $x_k$ (final estimate).
```

Requirement: linear convergence of the method $\mathcal{M}$. One of the main characteristic of Catalyst is to apply the method $\mathcal{M}$ to strongly-convex sub-problems, without requiring strong convexity of the objective f. As a consequence, Catalyst provides direct support for convex but non-strongly convex objectives to $\mathcal{M}$, which may be useful to extend the scope of application of techniques that need strong convexity to operate. Yet, Catalyst requires solving these sub-problems efficiently enough in order to control the complexity of the inner-loop computations. When applying $\mathcal{M}$ to minimize a strongly-convex function h, we assume that $\mathcal{M}$ is able to produce a sequence of iterates $(z_t)_{t\geq 0}$ such that

$$h(z_t) - h^* \leq C_{\mathcal{M}}(1 - \tau_{\mathcal{M}})^t (h(z_0) - h^*), \tag{12}$$

where z_0 is the initial point given to $\mathcal{M}$, and $\tau_{\mathcal{M}}$ in $(0, 1)$, $C_{\mathcal{M}} > 0$ are two constants. In such a case, we say that $\mathcal{M}$ admits a linear convergence rate. The quantity $\tau_{\mathcal{M}}$ controls the speed of convergence for solving the sub-problems: the larger is $\tau_{\mathcal{M}}$, the faster is the convergence. For a given algorithm $\mathcal{M}$, the quantity $\tau_{\mathcal{M}}$ depends usually on the condition number of h. For instance, for the proximal gradient method and many first-order algorithms, we simply have $\tau_{\mathcal{M}} = O((\mu+\kappa)/(L+\kappa))$, as h is $(\mu+\kappa)$-strongly convex and $(L+\kappa)$-smooth. Catalyst can also be applied to randomized methods $\mathcal{M}$ that satisfy (12) in expectation:

$$\mathbb{E}[h(z_t) - h^*] \leq C_{\mathcal{M}}(1-\tau_{\mathcal{M}})^t(h(z_0) - h^*), \tag{13}$$

Then, the complexity results of Section 4 also hold in expectation. This allows us to apply Catalyst to randomized block coordinate descent algorithms (see Richtárik and Takáč, 2014, and references therein), and some incremental algorithms such as SAG, SAGA, or SVRG. For other methods that admit a linear convergence rates in terms of duality gap, such as SDCA, MISO/Finito, Catalyst can also be applied as explained in Appendix C.

Stopping criteria. Catalyst may be used with three types of stopping criteria for solving the inner-loop problems. We now detail them below.

(a) *absolute accuracy*: we predefine a sequence $(\varepsilon_k)_{k\geq 0}$ of accuracies, and stop the method $\mathcal{M}$ by using the absolute stopping criterion ($\mathsf{C1}$). Our analysis suggests

- if f is μ-strongly convex,

$$\varepsilon_k = \frac{1}{2}(1-\rho)^k(f(x_0) - f^*) \quad \text{with} \quad \rho < \sqrt{q}\ .$$

- if f is convex but not strongly convex,

$$\varepsilon_k = \frac{f(x_0) - f^*}{2(k+2)^{4+\gamma}} \quad \text{with} \quad \gamma > 0\ .$$

Typically, $\gamma = 0.1$ and $\rho = 0.9\sqrt{q}$ are reasonable choices, both in theory and in practice. Of course, the quantity $f(x_0) - f^*$ is unknown and we need to upper bound it by a duality gap or by Lemma 2 as discussed in Section 2.3.

(b) *relative accuracy*: To use the relative stopping criterion ($\mathsf{C2}$), our analysis suggests the following choice for the sequence $(\delta_k)_{k\geq 0}$:

- if f is μ-strongly convex,

$$\delta_k = \frac{\sqrt{q}}{2-\sqrt{q}}\ .$$

- if f is convex but not strongly convex,

$$\delta_k = \frac{1}{(k+1)^2}\ .$$

(c) *fixed budget*: Finally, the simplest way of using Catalyst is to fix in advance the number T of iterations of the method $\mathcal{M}$ for solving the sub-problems without checking any optimality criterion. Whereas our analysis provides theoretical budgets that are compatible with this strategy, we found them to be pessimistic and impractical. Instead, we propose an aggressive strategy for incremental methods that simply consists of setting $T = n$. This setting was called the "one-pass" strategy in the original Catalyst paper (Lin et al., 2015a).

Warm-starts in inner loops. Besides linear convergence rate, an adequate warm-start strategy needs to be used to guarantee that the sub-problems will be solved in reasonable computational time. The intuition is that the previous solution may still be a good approximation of the current subproblem. Specifically, the following choices arise from the convergence analysis that will be detailed in Section 4.

Consider the minimization of the $(k+1)$-th subproblem $h_{k+1}(z) = f(z) + \frac{\kappa}{2}\|z - y_k\|^2$, we warm start the optimization method $\mathcal{M}$ at z_0 as following:

(a) when using criterion $(\mathsf{C1})$ to find x_{k+1} in $p^{\varepsilon_k}(y_k)$,

- if f is smooth ($\psi = 0$), then choose $z_0 = x_k + \frac{\kappa}{\kappa+\mu}(y_k - y_{k-1})$.
- if f is composite as in (1), then define $w_0 = x_k + \frac{\kappa}{\kappa+\mu}(y_k - y_{k-1})$ and

$$z_0 = [w_0]_\eta = \text{prox}_{\eta\psi}(w_0 - \eta g) \text{ with } \eta = \frac{1}{L+\kappa} \text{ and } g = \nabla f_0(w_0) + \kappa(w_0 - y_k).$$

(b) when using criteria $(\mathsf{C2})$ to find x_{k+1} in $g^{\delta_k}(y_k)$,

- if f is smooth ($\psi = 0$), then choose $z_0 = y_k$.
- if f is composite as in (1), then choose

$$z_0 = [y_k]_\eta = \text{prox}_{\eta\psi}(y_k - \eta\nabla f_0(y_k)) \quad \text{with} \quad \eta = \frac{1}{L+\kappa}.$$

(c) when using a fixed budget T, choose the same warm start strategy as in (b).

Note that the earlier conference paper (Lin et al., 2015a) considered the the warm start rule $z_0 = x_{k-1}$. That variant is also theoretically validated but it does not perform as well as the ones proposed here in practice.

Optimal balance: choice of parameter κ. Finally, the last ingredient is to find an optimal balance between the inner-loop (for solving each sub-problem) and outer-loop computations. To do so, we minimize our global complexity bounds with respect to the value of κ. As we shall see in Section 5, this strategy turns out to be reasonable in practice. Then, as shown in the theoretical section, the resulting rule of thumb is

We select κ by maximizing the ratio $\tau_{\mathcal{M}}/\sqrt{\mu+\kappa}$.

We recall that $\tau_{\mathcal{M}}$ characterizes how fast $\mathcal{M}$ solves the sub-problems, according to (12); typically, $\tau_{\mathcal{M}}$ depends on the condition number $\frac{L+\kappa}{\mu+\kappa}$ and is a function of κ.[2] In Table 2, we illustrate the choice of κ for different methods. Note that the resulting rule for incremental methods is very simple for the pracitioner: select κ such that the condition number $\frac{\bar{L}+\kappa}{\mu+\kappa}$ is of the order of n; then, the inner-complexity becomes $O(n \log(1/\varepsilon))$.

Method $\mathcal{M}$	Inner-complexity	$\tau_{\mathcal{M}}$	Choice for κ
FG	$O\left(n\frac{L+\kappa}{\mu+\kappa}\log\left(\frac{1}{\varepsilon}\right)\right)$	$\propto \frac{\mu+\kappa}{L+\kappa}$	$L-2\mu$
SAG/SAGA/SVRG	$O\left(\left(n+\frac{\bar{L}+\kappa}{\mu+\kappa}\right)\log\left(\frac{1}{\varepsilon}\right)\right)$	$\propto \frac{\mu+\kappa}{n(\mu+\kappa)+\bar{L}+\kappa}$	$\frac{\bar{L}-\mu}{n+1}-\mu$

Table 2: Example of choices of the parameter κ for the full gradient (FG) and incremental methods SAG/SAGA/SVRG. See Table 1 for details about the complexity.

4. Convergence and Complexity Analysis

We now present the complexity analysis of Catalyst. In Section 4.1, we analyze the convergence rate of the outer loop, regardless of the complexity for solving the sub-problems. Then, we analyze the complexity of the inner-loop computations for our various stopping criteria and warm-start strategies in Section 4.2. Section 4.3 combines the outer- and inner-loop analysis to provide the global complexity of Catalyst applied to a given optimization method $\mathcal{M}$.

4.1 Complexity Analysis for the Outer-Loop

The complexity analysis of the first variant of Catalyst we presented in (Lin et al., 2015a) used a tool called "estimate sequence", which was introduced by Nesterov (2004). Here, we provide a simpler proof. We start with criterion (C1), before extending the result to (C2).

4.1.1 Analysis for Criterion (C1)

The next theorem describes how the errors $(\varepsilon_k)_{k\geq 0}$ accumulate in Catalyst.

Theorem 3 (Convergence of outer-loop for criterion (C1)) *Consider the sequences $(x_k)_{k\geq 0}$ and $(y_k)_{k\geq 0}$ produced by Algorithm 2, assuming that x_k is in $p^{\varepsilon_k}(y_{k-1})$ for all $k \geq 1$, Then,*

$$f(x_k) - f^* \leq A_{k-1}\left(\sqrt{(1-\alpha_0)(f(x_0)-f^*) + \frac{\gamma_0}{2}\|x^*-x_0\|^2} + 3\sum_{j=1}^{k}\sqrt{\frac{\varepsilon_j}{A_{j-1}}}\right)^2,$$

[2] Note that the rule for the non strongly convex case, denoted here by $\mu = 0$, slightly differs from Lin et al. (2015a) and results from a tighter complexity analysis.

where

$$\gamma_0 = (\kappa + \mu)\alpha_0(\alpha_0 - q) \quad \textit{and} \quad A_k = \prod_{j=1}^{k}(1 - \alpha_j) \;\; \textit{with } A_0 = 1 \; . \tag{14}$$

Before we prove this theorem, we note that by setting $\varepsilon_k = 0$ for all k, the speed of convergence of $f(x_k) - f^*$ is driven by the sequence $(A_k)_{k \geq 0}$. Thus we first show the speed of A_k by recalling the Lemma 2.2.4 of Nesterov (2004).

Lemma 4 (Lemma 2.2.4 of Nesterov 2004) *Consider the quantities γ_0, A_k defined in (14) and the α_k's defined in Algorithm 2. Then, if $\gamma_0 \geq \mu$,*

$$A_k \leq \min \left\{ (1 - \sqrt{q})^k \, , \frac{4}{\left(2 + k\sqrt{\frac{\gamma_0}{\kappa}}\right)^2} \right\} .$$

For non-strongly convex objectives, A_k follows the classical accelerated $O(1/k^2)$ rate of convergence, whereas it achieves a linear convergence rate for the strongly convex case. Intuitively, we are applying an inexact Nesterov method on the Moreau envelope F, thus the convergence rate naturally depends on the inverse of its condition number, which is $q = \frac{\mu}{\mu+\kappa}$. We now provide the proof of the theorem below.

Proof We start by defining an approximate sufficient descent condition inspired by a remark of Chambolle and Pock (2015) regarding accelerated gradient descent methods. A related condition was also used by Paquette et al. (2018) in the context of non-convex optimization.

Approximate sufficient descent condition. Let us define the function

$$h_k(x) = f(x) + \frac{\kappa}{2}\|x - y_{k-1}\|^2.$$

Since $p(y_{k-1})$ is the unique minimizer of h_k, the strong convexity of h_k yields: for any $k \geq 1$, for all x in $\mathbb{R}^p$ and any $\theta_k > 0$,

$$\begin{aligned} h_k(x) &\geq h_k^* + \frac{\kappa + \mu}{2}\|x - p(y_{k-1})\|^2 \\ &\geq h_k^* + \frac{\kappa + \mu}{2}\left(1 - \theta_k\right)\|x - x_k\|^2 + \frac{\kappa + \mu}{2}\left(1 - \frac{1}{\theta_k}\right)\|x_k - p(y_{k-1})\|^2 \\ &\geq h_k(x_k) - \varepsilon_k + \frac{\kappa + \mu}{2}\left(1 - \theta_k\right)\|x - x_k\|^2 + \frac{\kappa + \mu}{2}\left(1 - \frac{1}{\theta_k}\right)\|x_k - p(y_{k-1})\|^2, \end{aligned}$$

where the $(\mu + \kappa)$-strong convexity of h_k is used in the first inequality; Lemma 19 is used in the second inequality, and the last one uses the relation $h_k(x_k) - h_k^* \leq \varepsilon_k$. Moreover, when $\theta_k \geq 1$, the last term is positive and we have

$$h_k(x) \geq h_k(x_k) - \varepsilon_k + \frac{\kappa + \mu}{2}\left(1 - \theta_k\right)\|x - x_k\|^2.$$

If instead $\theta_k \leq 1$, the coefficient $\frac{1}{\theta_k} - 1$ is non-negative and we have

$$-\frac{\kappa+\mu}{2}\left(\frac{1}{\theta_k}-1\right)\|x_k - p(y_{k-1})\|^2 \geq -\left(\frac{1}{\theta_k}-1\right)(h_k(x_k) - h_k^*) \geq -\left(\frac{1}{\theta_k}-1\right)\varepsilon_k.$$

In this case, we have

$$h_k(x) \geq h_k(x_k) - \frac{\varepsilon_k}{\theta_k} + \frac{\kappa+\mu}{2}(1-\theta_k)\|x - x_k\|^2.$$

As a result, we have for all value of $\theta_k > 0$,

$$h_k(x) \geq h_k(x_k) + \frac{\kappa+\mu}{2}(1-\theta_k)\|x - x_k\|^2 - \frac{\varepsilon_k}{\min\{1,\theta_k\}}.$$

After expanding the expression of h_k, we then obtain the approximate descent condition

$$f(x_k)+\frac{\kappa}{2}\|x_k-y_{k-1}\|^2+\frac{\kappa+\mu}{2}(1-\theta_k)\|x-x_k\|^2 \leq f(x)+\frac{\kappa}{2}\|x-y_{k-1}\|^2+\frac{\varepsilon_k}{\min\{1,\theta_k\}}. \quad (15)$$

Definition of the Lyapunov function. We introduce a sequence $(S_k)_{k\geq 0}$ that will act as a Lyapunov function, with

$$S_k = (1-\alpha_k)(f(x_k) - f^*) + \alpha_k \frac{\kappa\eta_k}{2}\|x^* - v_k\|^2. \quad (16)$$

where x^* is a minimizer of f, $(v_k)_{k\geq 0}$ is a sequence defined by $v_0 = x_0$ and

$$v_k = x_k + \frac{1-\alpha_{k-1}}{\alpha_{k-1}}(x_k - x_{k-1}) \quad \text{for } k \geq 1,$$

and $(\eta_k)_{k\geq 0}$ is an auxiliary quantity defined by

$$\eta_k = \frac{\alpha_k - q}{1-q}.$$

The way we introduce these variables allow us to write the following relationship,

$$y_k = \eta_k v_k + (1-\eta_k)x_k, \quad \text{for all } k \geq 0,$$

which follows from a simple calculation. Then by setting $z_k = \alpha_{k-1}x^* + (1-\alpha_{k-1})x_{k-1}$ the following relations hold for all $k \geq 1$.

$$\begin{aligned} f(z_k) &\leq \alpha_{k-1}f^* + (1-\alpha_{k-1})f(x_{k-1}) - \frac{\mu\alpha_{k-1}(1-\alpha_{k-1})}{2}\|x^* - x_{k-1}\|^2, \\ z_k - x_k &= \alpha_{k-1}(x^* - v_k), \end{aligned}$$

and also the following one

$$\begin{aligned} \|z_k - y_{k-1}\|^2 &= \|(\alpha_{k-1} - \eta_{k-1})(x^* - x_{k-1}) + \eta_{k-1}(x^* - v_{k-1})\|^2 \\ &= \alpha_{k-1}^2 \left\| \left(1 - \frac{\eta_{k-1}}{\alpha_{k-1}}\right)(x^* - x_{k-1}) + \frac{\eta_{k-1}}{\alpha_{k-1}}(x^* - v_{k-1}) \right\|^2 \\ &\leq \alpha_{k-1}^2 \left(1 - \frac{\eta_{k-1}}{\alpha_{k-1}}\right)\|x^* - x_{k-1}\|^2 + \alpha_{k-1}^2 \frac{\eta_{k-1}}{\alpha_{k-1}}\|x^* - v_{k-1}\|^2 \\ &= \alpha_{k-1}(\alpha_{k-1} - \eta_{k-1})\|x^* - x_{k-1}\|^2 + \alpha_{k-1}\eta_{k-1}\|x^* - v_{k-1}\|^2, \end{aligned}$$

where we used the convexity of the norm and the fact that $\eta_k \leq \alpha_k$. Using the previous relations in (15) with $x = z_k = \alpha_{k-1}x^* + (1-\alpha_{k-1})x_{k-1}$, gives for all $k \geq 1$,

$$\begin{aligned} f(x_k) &+ \frac{\kappa}{2}\|x_k - y_{k-1}\|^2 + \frac{\kappa+\mu}{2}\left(1-\theta_k\right)\alpha_{k-1}^2\|x^* - v_k\|^2 \\ &\leq \alpha_{k-1}f^* + (1-\alpha_{k-1})f(x_{k-1}) - \frac{\mu}{2}\alpha_{k-1}(1-\alpha_{k-1})\|x^* - x_{k-1}\|^2 \\ &+ \frac{\kappa\alpha_{k-1}(\alpha_{k-1}-\eta_{k-1})}{2}\|x^* - x_{k-1}\|^2 + \frac{\kappa\alpha_{k-1}\eta_{k-1}}{2}\|x^* - v_{k-1}\|^2 + \frac{\varepsilon_k}{\min\{1,\theta_k\}}. \end{aligned}$$

Remark that for all $k \geq 1$,

$$\alpha_{k-1} - \eta_{k-1} = \alpha_{k-1} - \frac{\alpha_{k-1}-q}{1-q} = \frac{q(1-\alpha_{k-1})}{1-q} = \frac{\mu}{\kappa}(1-\alpha_{k-1}),$$

and the quadratic terms involving $x^* - x_{k-1}$ cancel each other. Then, after noticing that for all $k \geq 1$,

$$\eta_k\alpha_k = \frac{\alpha_k^2 - q\alpha_k}{1-q} = \frac{(\kappa+\mu)(1-\alpha_k)\alpha_{k-1}^2}{\kappa},$$

which allows us to write

$$f(x_k) - f^* + \frac{\kappa+\mu}{2}\alpha_{k-1}^2\|x^* - v_k\|^2 = \frac{S_k}{1-\alpha_k}. \tag{17}$$

We are left, for all $k \geq 1$, with

$$\frac{1}{1-\alpha_k}S_k \leq S_{k-1} + \frac{\varepsilon_k}{\min\{1,\theta_k\}} - \frac{\kappa}{2}\|x_k - y_{k-1}\|^2 + \frac{(\kappa+\mu)\alpha_{k-1}^2\theta_k}{2}\|x^* - v_k\|^2. \tag{18}$$

Control of the approximation errors for criterion (C1). Using the fact that

$$\frac{1}{\min\{1,\theta_k\}} \leq 1 + \frac{1}{\theta_k},$$

we immediately derive from equation (18) that

$$\frac{1}{1-\alpha_k}S_k \leq S_{k-1} + \varepsilon_k + \frac{\varepsilon_k}{\theta_k} - \frac{\kappa}{2}\|x_k - y_{k-1}\|^2 + \frac{(\kappa+\mu)\alpha_{k-1}^2\theta_k}{2}\|x^* - v_k\|^2. \tag{19}$$

By minimizing the right-hand side of (19) with respect to θ_k, we obtain the following inequality

$$\frac{1}{1-\alpha_k}S_k \leq S_{k-1} + \varepsilon_k + \sqrt{2\varepsilon_k(\mu+\kappa)}\alpha_{k-1}\|x^* - v_k\|,$$

and after unrolling the recursion,

$$\frac{S_k}{A_k} \leq S_0 + \sum_{j=1}^{k}\frac{\varepsilon_j}{A_{j-1}} + \sum_{j=1}^{k}\frac{\sqrt{2\varepsilon_j(\mu+\kappa)}\alpha_{j-1}\|x^* - v_j\|}{A_{j-1}}.$$

From Equation (17), the lefthand side is larger than $\frac{(\mu+\kappa)\alpha_{k-1}^2\|x^*-v_k\|^2}{2A_{k-1}}$. We may now define $u_j = \frac{\sqrt{(\mu+\kappa)}\alpha_{j-1}\|x^*-v_j\|}{\sqrt{2A_{j-1}}}$ and $a_j = 2\frac{\sqrt{\varepsilon_j}}{\sqrt{A_{j-1}}}$, and we have

$$u_k^2 \leq S_0 + \sum_{j=1}^{k} \frac{\varepsilon_j}{A_{j-1}} + \sum_{j=1}^{k} a_j u_j \quad \text{for all } k \geq 1.$$

This allows us to apply Lemma 20, which yields

$$\begin{aligned} \frac{S_k}{A_k} &\leq \left(\sqrt{S_0 + \sum_{j=1}^{k} \frac{\varepsilon_j}{A_{j-1}}} + 2\sum_{j=1}^{k} \sqrt{\frac{\varepsilon_j}{A_{j-1}}} \right)^2, \\ &\leq \left(\sqrt{S_0} + 3\sum_{j=1}^{k} \sqrt{\frac{\varepsilon_j}{A_{j-1}}} \right)^2 \end{aligned}$$

which provides us the desired result given that $f(x_k) - f^* \leq \frac{S_k}{1-\alpha_k}$ and that $v_0 = x_0$. ■

We are now in shape to state the convergence rate of the Catalyst algorithm with criterion (C1), without taking into account yet the cost of solving the sub-problems. The next two propositions specialize Theorem 3 to the strongly convex case and non strongly convex cases, respectively. Their proofs are provided in Appendix B.

Proposition 5 (μ-strongly convex case, criterion (C1))
In Algorithm 2, choose $\alpha_0 = \sqrt{q}$ and

$$\varepsilon_k = \frac{2}{9}(f(x_0) - f^*)(1-\rho)^k \quad \textit{with} \quad \rho < \sqrt{q}.$$

Then, the sequence of iterates $(x_k)_{k\geq 0}$ satisfies

$$f(x_k) - f^* \leq \frac{8}{(\sqrt{q}-\rho)^2}(1-\rho)^{k+1}(f(x_0) - f^*).$$

Proposition 6 (Convex case, criterion (C1))
When $\mu = 0$, choose $\alpha_0 = 1$ and

$$\varepsilon_k = \frac{2(f(x_0) - f^*)}{9(k+1)^{4+\gamma}} \quad \textit{with} \quad \gamma > 0.$$

Then, Algorithm 2 generates iterates $(x_k)_{k\geq 0}$ such that

$$f(x_k) - f^* \leq \frac{8}{(k+1)^2}\left(\frac{\kappa}{2}\|x_0 - x^*\|^2 + \frac{4}{\gamma^2}(f(x_0) - f^*)\right).$$

4.1.2 Analysis for Criterion (C2)

Then, we may now analyze the convergence of Catalyst under criterion (C2), which offers similar guarantees as (C1), as far as the outer loop is concerned.

Theorem 7 (Convergence of outer-loop for criterion (C2)) *Consider the sequences $(x_k)_{k\geq 0}$ and $(y_k)_{k\geq 0}$ produced by Algorithm 2, assuming that x_k is in $g^{\delta_k}(y_{k-1})$ for all $k \geq 1$ and δ_k in $(0,1)$. Then,*

$$f(x_k) - f^* \leq \frac{A_{k-1}}{\prod_{j=1}^{k}(1-\delta_j)} \left((1-\alpha_0)(f(x_0)-f^*) + \frac{\gamma_0}{2}\|x_0 - x^*\|^2 \right),$$

where γ_0 and $(A_k)_{k\geq 0}$ are defined in (14) in Theorem 3.

Proof Remark that x_k in $g^{\delta_k}(y_{k-1})$ is equivalent to x_k in $p^{\varepsilon_k}(y_{k-1})$ with an adaptive error $\varepsilon_k = \frac{\delta_k \kappa}{2}\|x_k - y_{k-1}\|^2$. All steps of the proof of Theorem 3 hold for such values of ε_k and from (18), we may deduce

$$\frac{S_k}{1-\alpha_k} - \frac{(\kappa+\mu)\alpha_{k-1}^2\theta_k}{2}\|x^* - v_k\|^2 \leq S_{k-1} + \left(\frac{\delta_k \kappa}{2\min\{1,\theta_k\}} - \frac{\kappa}{2} \right) \|x_k - y_{k-1}\|^2.$$

Then, by choosing $\theta_k = \delta_k < 1$, the quadratic term on the right disappears and the left-hand side is greater than $\frac{1-\delta_k}{1-\alpha_k}S_k$. Thus,

$$S_k \leq \frac{1-\alpha_k}{1-\delta_k} S_{k-1} \leq \frac{A_k}{\prod_{j=1}^{k}(1-\delta_j)} S_0,$$

which is sufficient to conclude since $(1-\alpha_k)(f(x_k)-f^*) \leq S_k$. ■

The next propositions specialize Theorem 7 for specific choices of sequence $(\delta_k)_{k\geq 0}$ in the strongly and non strongly convex cases.

Proposition 8 (μ-strongly convex case, criterion (C2))
In Algorithm 2, choose $\alpha_0 = \sqrt{q}$ and

$$\delta_k = \frac{\sqrt{q}}{2-\sqrt{q}}.$$

Then, the sequence of iterates $(x_k)_{k\geq 0}$ satisfies

$$f(x_k) - f^* \leq 2\left(1 - \frac{\sqrt{q}}{2}\right)^k (f(x_0) - f^*).$$

Proof This is a direct application of Theorem 7 by remarking that $\gamma_0 = (1-\sqrt{q})\mu$ and

$$S_0 = (1-\sqrt{q})\left(f(x_0) - f^* + \frac{\mu}{2}\|x^* - x_0\|^2 \right) \leq 2(1-\sqrt{q})(f(x_0)-f^*).$$

And $\alpha_k = \sqrt{q}$ for all $k \geq 0$ leading to

$$\frac{1-\alpha_k}{1-\delta_k} = 1 - \frac{\sqrt{q}}{2}$$

■

Proposition 9 (Convex case, criterion (C2))
When $\mu = 0$, choose $\alpha_0 = 1$ and

$$\delta_k = \frac{1}{(k+1)^2}.$$

Then, Algorithm 2 generates iterates $(x_k)_{k \geq 0}$ such that

$$f(x_k) - f^* \leq \frac{4\kappa \|x_0 - x^*\|^2}{(k+1)^2}. \tag{20}$$

Proof This is a direct application of Theorem 7 by remarking that $\gamma_0 = \kappa$, $A_k \leq \frac{4}{(k+2)^2}$ (Lemma 4) and

$$\prod_{i=1}^{k} \left(1 - \frac{1}{(i+1)^2}\right) = \prod_{i=1}^{k} \frac{i(i+2)}{(i+1)^2} = \frac{k+2}{2(k+1)} \geq \frac{1}{2}.$$

■

Remark 10 *In fact, the choice of δ_k can be improved by taking $\delta_k = \frac{1}{(k+1)^{1+\gamma}}$ for any $\gamma > 0$, which comes at the price of a larger constant in (20).*

4.2 Analysis of Warm-start Strategies for the Inner Loop

In this section, we study the complexity of solving the subproblems with the proposed warm start strategies. The only assumption we make on the optimization method $\mathcal{M}$ is that it enjoys linear convergence when solving a strongly convex problem—meaning, it satisfies either (12) or its randomized variant (13). Then, the following lemma gives us a relation between the accuracy required to solve the sub-problems and the corresponding complexity.

Lemma 11 (Accuracy vs. complexity) *Let us consider a strongly convex objective h and a linearly convergent method $\mathcal{M}$ generating a sequence of iterates $(z_t)_{t \geq 0}$ for minimizing h. Consider the complexity $T(\varepsilon) = \inf\{t \geq 0, h(z_t) - h^* \leq \varepsilon\}$, where $\varepsilon > 0$ is the target accuracy and h^* is the minimum value of h. Then,*

1. *If $\mathcal{M}$ is deterministic and satisfies (12), we have*

$$T(\varepsilon) \leq \frac{1}{\tau_{\mathcal{M}}} \log\left(\frac{C_{\mathcal{M}}(h(z_0) - h^*)}{\varepsilon}\right).$$

2. *If $\mathcal{M}$ is randomized and satisfies (13), we have*

$$\mathbb{E}[T(\varepsilon)] \leq \frac{1}{\tau_{\mathcal{M}}} \log\left(\frac{2C_{\mathcal{M}}(h(z_0) - h^*)}{\tau_{\mathcal{M}}\varepsilon}\right) + 1$$

The proof of the deterministic case is straightforward and the proof of the randomized case is provided in Appendix B.4. From the previous result, a good initialization is essential for fast convergence. More precisely, it suffices to control the initialization $\frac{h(z_0) - h^*}{\varepsilon}$ in order to bound the number of iterations $T(\varepsilon)$. For that purpose, we analyze the quality of various warm-start strategies.

4.2.1 Warm Start Strategies for Criterion (C1)

The next proposition characterizes the quality of initialization for (C1).

Proposition 12 (Warm start for criterion (C1)) *Assume that $\mathcal{M}$ is linearly convergent for strongly convex problems with parameter $\tau_{\mathcal{M}}$ according to (12), or according to (13) in the randomized case. At iteration $k+1$ of Algorithm 2, given the previous iterate x_k in $p^{\varepsilon_k}(y_{k-1})$, we consider the following function*

$$h_{k+1}(z) = f(z) + \frac{\kappa}{2}\|z - y_k\|^2,$$

which we minimize with $\mathcal{M}$, producing a sequence $(z_t)_{t\geq 0}$. Then,

- *when f is smooth, choose $z_0 = x_k + \frac{\kappa}{\kappa+\mu}(y_k - y_{k-1})$;*
- *when $f = f_0 + \psi$ is composite, choose $z_0 = [w_0]_\eta = \text{prox}_{\eta\psi}(w_0 - \eta\nabla h_0(w_0))$ with $w_0 = x_k + \frac{\kappa}{\kappa+\mu}(y_k - y_{k-1})$, $\eta = \frac{1}{L+\kappa}$ and $h_0 = f_0 + \frac{\kappa}{2}\|\cdot - y_k\|^2$.*

We also assume that we choose α_0 and $(\varepsilon_k)_{k\geq 0}$ according to Proposition 5 for $\mu > 0$, or Proposition 6 for $\mu = 0$. Then,

1. *if f is μ-strongly convex, $h_{k+1}(z_0) - h^*_{k+1} \leq C\varepsilon_{k+1}$ where,*

$$C = \frac{L+\kappa}{\kappa+\mu}\left(\frac{2}{1-\rho} + \frac{2592(\kappa+\mu)}{(1-\rho)^2(\sqrt{q}-\rho)^2\mu}\right) \quad \text{if } f \text{ is smooth}, \tag{21}$$

or

$$C = \frac{L+\kappa}{\kappa+\mu}\left(\frac{2}{1-\rho} + \frac{23328(L+\kappa)}{(1-\rho)^2(\sqrt{q}-\rho)^2\mu}\right) \quad \text{if } f \text{ is composite}. \tag{22}$$

2. *if f is convex with bounded level sets, there exists a constant $B > 0$ that only depends on f, x_0 and κ such that*

$$h_{k+1}(z_0) - h^*_{k+1} \leq B. \tag{23}$$

Proof We treat the smooth and composite cases separately.

Smooth and strongly-convex case. When f is smooth, by the gradient Lipschitz assumption,

$$h_{k+1}(z_0) - h^*_{k+1} \leq \frac{(L+\kappa)}{2}\|z_0 - p(y_k)\|^2.$$

Moreover,

$$\begin{aligned}\|z_0 - p(y_k)\|^2 &= \left\|x_k + \frac{\kappa}{\kappa+\mu}(y_k - y_{k-1}) - p(y_k)\right\|^2 \\ &= \left\|x_k - p(y_{k-1}) + \frac{\kappa}{\kappa+\mu}(y_k - y_{k-1}) - (p(y_k) - p(y_{k-1}))\right\|^2 \\ &\leq 2\|x_k - p(y_{k-1})\|^2 + 2\left\|\frac{\kappa}{\kappa+\mu}(y_k - y_{k-1}) - (p(y_k) - p(y_{k-1}))\right\|^2.\end{aligned}$$

Since x_k is in $p^{\varepsilon_k}(y_{k-1})$, we may control the first quadratic term on the right by noting that

$$\|x_k - p(y_{k-1})\|^2 \leq \frac{2}{\kappa+\mu}(h_k(x_k) - h_k^*) \leq \frac{2\varepsilon_k}{\kappa+\mu}.$$

Moreover, by the coerciveness property of the proximal operator,

$$\left\| \frac{\kappa}{\kappa+\mu}(y_k - y_{k-1}) - (p(y_k) - p(y_{k-1})) \right\|^2 \leq \|y_k - y_{k-1}\|^2,$$

see Appendix B.5 for the proof. As a consequence,

$$\begin{aligned} h_{k+1}(z_0) - h_{k+1}^* &\leq \frac{(L+\kappa)}{2}\|z_0 - p(y_k)\|^2 \\ &\leq 2\frac{L+\kappa}{\mu+\kappa}\varepsilon_k + (L+\kappa)\|y_k - y_{k-1}\|^2, \end{aligned} \tag{24}$$

Then, we need to control the term $\|y_k - y_{k-1}\|^2$. Inspired by the proof of accelerated SDCA of Shalev-Shwartz and Zhang (2016),

$$\begin{aligned} \|y_k - y_{k-1}\| &= \|x_k + \beta_k(x_k - x_{k-1}) - x_{k-1} - \beta_{k-1}(x_{k-1} - x_{k-2})\| \\ &\leq (1+\beta_k)\|x_k - x_{k-1}\| + \beta_{k-1}\|x_{k-1} - x_{k-2}\| \\ &\leq 3\max\left\{\|x_k - x_{k-1}\|, \|x_{k-1} - x_{k-2}\|\right\}, \end{aligned}$$

The last inequality was due to the fact that $\beta_k \leq 1$. In fact,

$$\beta_k^2 = \frac{\left(\alpha_{k-1} - \alpha_{k-1}^2\right)^2}{\left(\alpha_{k-1}^2 + \alpha_k\right)^2} = \frac{\alpha_{k-1}^2 + \alpha_{k-1}^4 - 2\alpha_{k-1}^3}{\alpha_k^2 + 2\alpha_k\alpha_{k-1}^2 + \alpha_{k-1}^4} = \frac{\alpha_{k-1}^2 + \alpha_{k-1}^4 - 2\alpha_{k-1}^3}{\alpha_{k-1}^2 + \alpha_{k-1}^4 + q\alpha_k + \alpha_k\alpha_{k-1}^2} \leq 1,$$

where the last equality uses the relation $\alpha_k^2 + \alpha_k\alpha_{k-1}^2 = \alpha_{k-1}^2 + q\alpha_k$ from (10). Then,

$$\|x_k - x_{k-1}\| \leq \|x_k - x^*\| + \|x_{k-1} - x^*\|,$$

and by strong convexity of f

$$\frac{\mu}{2}\|x_k - x^*\|^2 \leq f(x_k) - f^* \leq \frac{36}{(\sqrt{q}-\rho)^2}\varepsilon_{k+1},$$

where the last inequality is obtained from Proposition 5. As a result,

$$\begin{aligned} \|y_k - y_{k-1}\|^2 &\leq 9\max\left\{\|x_k - x_{k-1}\|^2, \|x_{k-1} - x_{k-2}\|^2\right\} \\ &\leq 36\max\left\{\|x_k - x^*\|^2, \|x_{k-1} - x^*\|^2, \|x_{k-2} - x^*\|^2\right\} \\ &\leq \frac{2592\,\varepsilon_{k-1}}{(\sqrt{q}-\rho)^2\mu}. \end{aligned}$$

Since $\varepsilon_{k+1} = (1-\rho)^2\varepsilon_{k-1}$, we may now obtain (21) from (24) and the previous bound.

Smooth and convex case. When $\mu = 0$, Eq. (24) is still valid but we need to control $\|y_k - y_{k-1}\|^2$ in a different way. From Proposition 6, the sequence $(f(x_k))_{k\geq 0}$ is bounded by a constant that only depends on f and x_0; therefore, by the bounded level set assumption, there exists $R > 0$ such that

$$\|x_k - x^*\| \leq R, \quad \text{for all } k \geq 0.$$

Thus, following the same argument as the strongly convex case, we have

$$\|y_k - y_{k-1}\| \leq 36R^2 \quad \text{for all } k \geq 1,$$

and we obtain (23) by combining the previous inequality with (24).

Composite case. By using the notation of gradient mapping introduced in (7), we have $z_0 = [w_0]_\eta$. By following similar steps as in the proof of Lemma 2, the gradient mapping satisfies the following relation

$$h_{k+1}(z_0) - h_{k+1}^* \leq \frac{1}{2(\kappa+\mu)} \left\| \frac{1}{\eta}(w_0 - z_0) \right\|^2,$$

and it is sufficient to bound $\|w_0 - z_0\| = \|w_0 - [w_0]_\eta\|$. For that, we introduce

$$[x_k]_\eta = \text{prox}_{\eta\psi}(x_k - \eta(\nabla f_0(x_k) + \kappa(x_k - y_{k-1}))).$$

Then,

$$\|w_0 - [w_0]_\eta\| \leq \|w_0 - x_k\| + \|x_k - [x_k]_\eta\| + \|[x_k]_\eta - [w_0]_\eta\|, \tag{25}$$

and we will bound each term on the right. By construction

$$\|w_0 - x_k\| = \frac{\kappa}{\kappa+\mu}\|y_k - y_{k-1}\| \leq \|y_k - y_{k-1}\|.$$

Next, it is possible to show that the gradient mapping satisfies the following relation (see Nesterov, 2013),

$$\frac{1}{2\eta}\|x_k - [x_k]_\eta\|^2 \leq h_k(x_k) - h_k^* \leq \varepsilon_k.$$

And then since $[x_k]_\eta = \text{prox}_{\eta\psi}(x_k - \eta(\nabla f_0(x_k) + \kappa(x_k - y_{k-1})))$ and $[w_0]_\eta = \text{prox}_{\eta\psi}(w_0 - \eta(\nabla f_0(w_0) + \kappa(w_0 - y_k)))$. From the non expansiveness of the proximal operator, we have

$$\begin{aligned}
\|[x_k]_\eta - [w_0]_\eta\| &\leq \|x_k - \eta(\nabla f_0(x_k) + \kappa(x_k - y_{k-1})) - (w_0 - \eta(\nabla f_0(w_0) + \kappa(w_0 - y_k)))\,\| \\
&\leq \|x_k - \eta(\nabla f_0(x_k) + \kappa(x_k - y_{k-1})) - (w_0 - \eta(\nabla f_0(w_0) + \kappa(w_0 - y_{k-1})))\,\| \\
&\qquad + \eta\kappa\|y_k - y_{k-1}\| \\
&\leq \|x_k - w_0\| + \eta\kappa\|y_k - y_{k-1}\| \\
&\leq 2\|y_k - y_{k-1}\|.
\end{aligned}$$

We have used the fact that $\|x - \eta\nabla h(x) - (y - \eta\nabla h(y))\| \leq \|x - y\|$. By combining the previous inequalities with (25), we finally have

$$\|w_0 - [w_0]_\eta\| \leq \sqrt{2\eta\varepsilon_k} + 3\|y_k - y_{k-1}\|.$$

Thus, by using the fact that $(a+b)^2 \leq 2a^2 + 2b^2$ for all a, b,

$$h_{k+1}(z_0) - h_{k+1}^* \leq \frac{L+\kappa}{\kappa+\mu}\left(2\varepsilon_k + 9(L+\kappa)\|y_k - y_{k-1}\|^2\right),$$

and we can obtain (22) and (23) by upper-bounding $\|y_k - y_{k-1}\|^2$ in a similar way as in the smooth case, both when $\mu > 0$ and $\mu = 0$.

■

Finally, the complexity of the inner loop can be obtained directly by combining the previous proposition with Lemma 11.

Corollary 13 (Inner-loop Complexity for Criterion (C1)) *Consider the setting of Proposition 12; then, the sequence $(z_t)_{t\geq 0}$ minimizing h_{k+1} is such that the complexity $T_{k+1} = \inf\{t \geq 0, h_{k+1}(z_t) - h_{k+1}^* \leq \varepsilon_{k+1}\}$ satisfies*

$$T_{k+1} \leq \frac{1}{\tau_{\mathcal{M}}} \log\left(C_{\mathcal{M}} C\right) \quad \textit{if } \mu > 0 \quad \Longrightarrow \quad T_{k+1} = \tilde{O}\left(\frac{1}{\tau_{\mathcal{M}}}\right),$$

where C is the constant defined in (21) or in (22) for the composite case; and

$$T_{k+1} \leq \frac{1}{\tau_{\mathcal{M}}} \log\left(\frac{9C_{\mathcal{M}}(k+2)^{4+\eta}B}{2(f(x_0)-f^*)}\right) \quad \textit{if } \mu = 0 \quad \Longrightarrow \quad T_{k+1} = \tilde{O}\left(\frac{\log(k+2)}{\tau_{\mathcal{M}}}\right),$$

where B is the uniform upper bound in (23). Furthermore, when $\mathcal{M}$ is randomized, the expected complexity $\mathbb{E}[T_{k+1}]$ is similar, up to a factor $2/\tau_{\mathcal{M}}$ in the logarithm—see Lemma 11, and we have $\mathbb{E}[T_{k+1}] = \tilde{O}(1/\tau_{\mathcal{M}})$ when $\mu > 0$ and $\mathbb{E}[T_{k+1}] = \tilde{O}(\log(k+2)/\tau_{\mathcal{M}})$. Here, $\tilde{O}(.)$ hides logarithmic dependencies in parameters $\mu, L, \kappa, C_{\mathcal{M}}$, $\tau_{\mathcal{M}}$ and $f(x_0) - f^*$.

4.2.2 Warm Start Strategies for Criterion (C2)

We may now analyze the inner-loop complexity for criterion (C2) leading to upper bounds with smaller constants and simpler proofs. Note also that in the convex case, the bounded level set condition will not be needed, unlike for criterion (C1). To proceed, we start with a simple lemma that gives us a sufficient condition for (C2) to be satisfied.

Lemma 14 (Sufficient condition for criterion (C2)) *If a point z satisfies*

$$h_{k+1}(z) - h_{k+1}^* \leq \frac{\delta_{k+1}\kappa}{8}\|p(y_k) - y_k\|^2,$$

then z is in $g^{\delta_{k+1}}(y_k)$.

Proof

$$\begin{aligned}
h_{k+1}(z) - h_{k+1}^* &\leq \frac{\delta_{k+1}\kappa}{8}\|p(y_k) - y_k\|^2 \\
&\leq \frac{\delta_{k+1}\kappa}{4}\left(\|p(y_k) - z\|^2 + \|z - y_k\|^2\right) \\
&\leq \frac{\delta_{k+1}\kappa}{4}\left(\frac{2}{\mu+\kappa}(h_{k+1}(z) - h_{k+1}^*) + \|z - y_k\|^2\right) \\
&\leq \frac{1}{2}\left(h_{k+1}(z) - h_{k+1}^*\right) + \frac{\delta_{k+1}\kappa}{4}\|z - y_k\|^2.
\end{aligned}$$

Rearranging the terms gives the desired result. ■

With the previous result, we can control the complexity of the inner-loop minimization with Lemma 11 by choosing $\varepsilon = \frac{\delta_{k+1}\kappa}{8}\|p(y_k) - y_k\|^2$. However, to obtain a meaningful upper bound, we need to control the ratio

$$\frac{h_{k+1}(z_0) - h_{k+1}^*}{\varepsilon} = \frac{8(h_{k+1}(z_0) - h_{k+1}^*)}{\delta_{k+1}\kappa\|p(y_k) - y_k\|^2}.$$

Proposition 15 (Warm start for criterion (C2)) *Assume that $\mathcal{M}$ is linearly convergent for strongly convex problems with parameter $\tau_{\mathcal{M}}$ according to (12), or according to (13) in the randomized case. At iteration $k+1$ of Algorithm 2, given the previous iterate x_k in $g^{\delta_k}(y_{k-1})$, we consider the following function*

$$h_{k+1}(z) = f(z) + \frac{\kappa}{2}\|z - y_k\|^2,$$

which we minimize with $\mathcal{M}$, producing a sequence $(z_t)_{t\geq 0}$. Then,

- *when f is smooth, set $z_0 = y_k$;*
- *when $f = f_0 + \psi$ is composite, set $z_0 = [y_k]_\eta = \mathrm{prox}_{\eta\psi}(y_k - \eta\nabla f_0(y_k))$ with $\eta = \frac{1}{L+\kappa}$.*

Then,

$$h_{k+1}(z_0) - h_{k+1}^* \leq \frac{L+\kappa}{2}\|p(y_k) - y_k\|^2. \tag{26}$$

Proof When f is smooth, the optimality conditions of $p(y_k)$ yield $\nabla h_{k+1}(p(y_k)) = \nabla f(p(y_k)) + \kappa(p(y_k) - y_k) = 0$. As a result,

$$\begin{aligned}
h_{k+1}(z_0) - h_{k+1}^* &= f(y_k) - \left(f(p(y_k)) + \frac{\kappa}{2}\|p(y_k) - y_k\|^2\right) \\
&\leq f(p(y_k)) + \langle \nabla f(p(y_k)), y_k - p(y_k)\rangle + \frac{L}{2}\|y_k - p(y_k)\|^2 \\
&\qquad - \left(f(p(y_k)) + \frac{\kappa}{2}\|p(y_k) - y_k\|^2\right) \\
&= \frac{L+\kappa}{2}\|p(y_k) - y_k\|^2.
\end{aligned}$$

When f is composite, we use the inequality in Lemma 2.3 of Beck and Teboulle (2009): for any z,

$$h_{k+1}(z) - h_{k+1}(z_0) \geq \frac{L+\kappa}{2}\|z_0 - y_k\|^2 + (L+\kappa)\langle z_0 - y_k, y_k - z\rangle,$$

Then, we apply this inequality with $z = p(y_k)$, and thus,

$$\begin{aligned} h_{k+1}(z_0) - h^*_{k+1} &\leq -\frac{L+\kappa}{2}\|z_0 - y_k\|^2 - (L+\kappa)\langle z_0 - y_k, y_k - p(y_k)\rangle \\ &\leq \frac{L+\kappa}{2}\|p(y_k) - y_k\|^2. \end{aligned}$$

■

We are now in shape to derive a complexity bound for criterion (C2), which is obtained by combining directly Lemma 11 with the value $\varepsilon = \frac{\delta_{k+1}\kappa}{8}\|p(y_k) - y_k\|^2$, Lemma 14, and the previous proposition.

Corollary 16 (Inner-loop Complexity for Criterion (C2)) *Consider the setting of Proposition 15 when $\mathcal{M}$ is deterministic; assume further that α_0 and $(\delta_k)_{k\geq 0}$ are chosen according to Proposition 8 for $\mu > 0$, or Proposition 9 for $\mu = 0$.*

Then, the sequence $(z_t)_{t\geq 0}$ is such that the complexity $T_{k+1} = \inf\{t \geq 0, z_t \in g^{\delta_{k+1}}(y_k)\}$ satisfies

$$T_{k+1} \leq \frac{1}{\tau_{\mathcal{M}}} \log\left(4C_{\mathcal{M}}\frac{(L+\kappa)}{\kappa}\frac{2-\sqrt{q}}{\sqrt{q}}\right) \quad \textit{when } \mu > 0,$$

and

$$T_{k+1} \leq \frac{1}{\tau_{\mathcal{M}}} \log\left(4C_{\mathcal{M}}\frac{(L+\kappa)}{\kappa}(k+2)^2\right) \quad \textit{when } \mu = 0.$$

When $\mathcal{M}$ is randomized, the expected complexity is similar, up to a factor $2/\tau_{\mathcal{M}}$ in the logarithm—see Lemma 11, and we have $\mathbb{E}[T_{k+1}] = \tilde{O}(1/\tau_{\mathcal{M}})$ when $\mu > 0$ and $\mathbb{E}[T_{k+1}] = \tilde{O}(\log(k+2)/\tau_{\mathcal{M}})$.

The inner-loop complexity is asymptotically similar with criterion (C2) as with criterion (C1), but the constants are significantly better.

4.3 Global Complexity Analysis

In this section, we combine the previous outer-loop and inner-loop convergence results to derive a global complexity bound. We treat here the strongly convex ($\mu > 0$) and convex ($\mu = 0$) cases separately.

4.3.1 Strongly Convex Case

When the problem is strongly convex, we remark that the subproblems are solved in a constant number of iterations $T_k = T = \tilde{O}\left(\frac{1}{\tau_{\mathcal{M}}}\right)$ for both criteria (C1) and (C2). This means that the iterate x_k in Algorithm 2 is obtained after $s = kT$ iterations of the method $\mathcal{M}$. Thus, the true convergence rate of Catalyst applied to $\mathcal{M}$ is of the form

$$f_s - f^* = f\left(x_{\frac{s}{T}}\right) - f^* \leq C'(1-\rho)^{\frac{s}{T}}(f(x_0) - f^*) \leq C'\left(1 - \frac{\rho}{T}\right)^s (f(x_0) - f^*), \qquad (27)$$

where $f_s = f(x_k)$ is the function value after s iterations of $\mathcal{M}$. Then, choosing κ consists of maximizing the rate of convergence (27). In other words, we want to maximize $\sqrt{q}/T = \tilde{O}(\sqrt{q}\tau_{\mathcal{M}})$. Since $q = \frac{\mu}{\mu+\kappa}$, this naturally lead to the maximization of $\tau_{\mathcal{M}}/\sqrt{\mu+\kappa}$. We now state more formally the global convergence result in terms of complexity.

Proposition 17 (Global Complexity for strongly convex objectives) *When f is μ-strongly convex and all parameters are chosen according to Propositions 5 and 12 when using criterion (C1), or Propositions 8 and 15 for (C2), then Algorithm 2 finds a solution $\hat{x}$ such that $f(\hat{x}) - f^* \leq \varepsilon$ in at most $N_{\mathcal{M}}$ iterations of a deterministic method $\mathcal{M}$ with*

1. *when criterion (C1) is used,*
$$N_{\mathcal{M}} \leq \frac{1}{\tau_{\mathcal{M}}\rho}\log\left(C_{\mathcal{M}}C\right)\cdot\log\left(\frac{8(f(x_0)-f^*)}{(\sqrt{q}-\rho)^2\varepsilon}\right) = \tilde{O}\left(\frac{1}{\tau_{\mathcal{M}}\sqrt{q}}\log\left(\frac{1}{\varepsilon}\right)\right),$$
where $\rho = 0.9\sqrt{q}$ and C is the constant defined in (21) or (22) for the composite case;

2. *when criterion (C2) is used,*
$$N_{\mathcal{M}} \leq \frac{2}{\tau_{\mathcal{M}}\sqrt{q}}\log\left(4C_{\mathcal{M}}\frac{L+\kappa}{\kappa}\frac{2-\sqrt{q}}{\sqrt{q}}\right)\cdot\log\left(\frac{2(f(x_0)-f^*)}{\varepsilon}\right) = \tilde{O}\left(\frac{1}{\tau_{\mathcal{M}}\sqrt{q}}\log\left(\frac{1}{\varepsilon}\right)\right).$$

Note that similar results hold in terms of expected number of iterations when the method $\mathcal{M}$ is randomized (see the end of Proposition 12).

Proof Let K be the number of iterations of the outer-loop algorithm required to obtain an ε-accurate solution. From Proposition 5, using (C1) criterion yields
$$K \leq \frac{1}{\rho}\log\left(\frac{8(f(x_0)-f^*)}{(\sqrt{q}-\rho)^2\varepsilon}\right).$$
From Proposition 8, using (C2) criterion yields
$$K \leq \frac{2}{\sqrt{q}}\log\left(\frac{2(f(x_0)-f^*)}{\varepsilon}\right).$$
Then since the number of runs of $\mathcal{M}$ is constant for any inner loop, the total number $N_{\mathcal{M}}$ is given by KT where T is respectively given by Corollaries 13 and 16. ■

4.3.2 Convex, but not Strongly Convex Case

When $\mu = 0$, the number of iterations for solving each subproblems grows logarithmically, which means that the iterate x_k in Algorithm 2 is obtained after $s =\leq kT\log(k+2)$ iterations of the method $\mathcal{M}$, where T is a constant. By using the global iteration counter $s = kT\log(k+2)$, we finally have
$$f_s - f^* \leq C'\frac{\log^2(s)}{s^2}\left(f(x_0) - f^* + \frac{\kappa}{2}\|x_0 - x^*\|^2\right). \tag{28}$$
This rate is *near-optimal*, up to a logarithmic factor, when compared to the optimal rate $O(1/s^2)$. This may be the price to pay for using a generic acceleration scheme. As before, we detail the global complexity bound for convex objectives in the next proposition.

Proposition 18 (Global complexity for convex objectives) *When f is convex and all parameters are chosen according to Propositions 6 and 12 when using criterion (C1), or Propositions 9 and 15 for criterion (C2), then Algorithm 2 finds a solution $\hat{x}$ such that $f(\hat{x}) - f^* \leq \varepsilon$ in at most $N_{\mathcal{M}}$ iterations of a deterministic method $\mathcal{M}$ with*

1. *when criterion (C1) is applied*

$$N_{\mathcal{M}} \leq \frac{1}{\tau_{\mathcal{M}}} K \log\left(\frac{9C_{\mathcal{M}}BK^{4+\gamma}}{2(f(x_0)-f^*)}\right) = \tilde{O}\left(\frac{1}{\tau_{\mathcal{M}}}\sqrt{\frac{\kappa}{\varepsilon}}\log\left(\frac{1}{\varepsilon}\right)\right),$$

where,

$$K_\varepsilon = \sqrt{\frac{8\left(\frac{\kappa}{2}\|x_0 - x^*\|^2 + \frac{4}{\gamma^2}(f(x_0)-f^*)\right)}{\varepsilon}};$$

2. *when criterion (C2) is applied,*

$$\begin{aligned} N_{\mathcal{M}} &\leq \frac{1}{\tau_{\mathcal{M}}}\sqrt{\frac{4\kappa\|x_0 - x^*\|^2}{\varepsilon}}\log\left(\frac{16C_{\mathcal{M}}(L+\kappa)\|x_0-x^*\|^2}{\varepsilon}\right) \\ &= \tilde{O}\left(\frac{1}{\tau_{\mathcal{M}}}\sqrt{\frac{\kappa}{\varepsilon}}\log\left(\frac{1}{\varepsilon}\right)\right). \end{aligned}$$

Note that similar results hold in terms of expected number of iterations when the method $\mathcal{M}$ is randomized (see the end of Proposition 15).

Proof Let K denote the number of outer-loop iterations required to achieve an ε-accurate solution. From Proposition 6, when (C1) is applied, we have

$$K \leq \sqrt{\frac{8\left(\frac{\kappa}{2}\|x_0 - x^*\|^2 + \frac{4}{\gamma^2}(f(x_0)-f^*)\right)}{\varepsilon}}.$$

From Proposition 9, when (C2) is applied, we have

$$K \leq \sqrt{\frac{4\kappa\|x_0 - x^*\|^2}{\varepsilon}}.$$

Since the number of runs in the inner loop is increasing, we have

$$N_{\mathcal{M}} = \sum_{i=1}^{K} T_i \leq K T_K.$$

Respectively apply T_K obtained from Corollary 13 and Corollary 16 gives the result. ■

Theoretical foundations of the choice of κ. The parameter κ plays an important rule in the global complexity result. The linear convergence parameter $\tau_{\mathcal{M}}$ depends typically on κ since it controls the strong convexity parameter of the subproblems. The natural way to choose κ is to minimize the global complexity given by Proposition 17 and Proposition 18, which leads to the following rule

$$\boxed{\textbf{Choose } \boldsymbol{\kappa} \textbf{ to maximize } \frac{\tau_{\mathcal{M}}}{\sqrt{\mu+\kappa}},}$$

where $\mu = 0$ when the problem is convex but not strongly convex. We now illustrate two examples when applying Catalyst to the classical gradient descent method and to the incremental approach SVRG.

Gradient descent. When $\mathcal{M}$ is the gradient descent method, we have

$$\tau_{\mathcal{M}} = \frac{\mu+\kappa}{L+\kappa}.$$

Maximizing the ratio $\frac{\tau_{\mathcal{M}}}{\sqrt{\mu+\kappa}}$ gives

$$\kappa = L - 2\mu, \qquad \text{when } L > 2\mu.$$

Consequently, the complexity in terms of gradient evaluations for minimizing the finite sum (2), where each iteration of $\mathcal{M}$ cost n gradients, is given by

$$N_{\mathcal{M}} = \begin{cases} \tilde{O}\left(n\sqrt{\frac{L}{\mu}}\log\left(\frac{1}{\varepsilon}\right)\right) & \text{when } \mu > 0; \\ \tilde{O}\left(n\sqrt{\frac{L}{\varepsilon}}\log\left(\frac{1}{\varepsilon}\right)\right) & \text{when } \mu = 0. \end{cases}$$

These rates are near-optimal up to logarithmic constants according to the first-order lower bound (Nemirovskii and Yudin, 1983; Nesterov, 2004).

SVRG. For SVRG (Xiao and Zhang, 2014) applied to the same finite-sum objective,

$$\tau_{\mathcal{M}} = \frac{1}{n + \frac{\bar{L}+\kappa}{\mu+\kappa}}.$$

Thus, maximizing the corresponding ratio gives

$$\kappa = \frac{\bar{L}-\mu}{n+1} - \mu, \qquad \text{when } \bar{L} > (n+2)\mu.$$

Consequently, the resulting global complexity, here in terms of expected number of gradient evaluations, is given by

$$\mathbb{E}[N_{\mathcal{M}}] = \begin{cases} \tilde{O}\left(\sqrt{n\frac{\bar{L}}{\mu}}\log\left(\frac{1}{\varepsilon}\right)\right) & \text{when } \mu > 0; \\ \tilde{O}\left(\sqrt{\frac{n\bar{L}}{\varepsilon}}\log\left(\frac{1}{\varepsilon}\right)\right) & \text{when } \mu = 0. \end{cases}$$

Note that we treat here only the case $\bar{L} > (n+2)\mu$ to simplify, see Table 1 for a general results. We also remark that Catalyst can be applied to similar incremental algorithms such as SAG/SAGA (Schmidt et al., 2017; Defazio et al., 2014a) or dual-type algorithm MISO/Finito (Mairal, 2015; Defazio et al., 2014b) or SDCA Shalev-Shwartz and Zhang (2012). Moreover, the resulting convergence rates are near-optimal up to logarithmic constants according to the first-order lower bound (Woodworth and Srebro, 2016; Arjevani and Shamir, 2016).

4.3.3 Practical Aspects of the Theoretical Analysis

So far, we have not discussed the fixed budget criterion mentioned in Section 3. The idea is quite natural and simple to implement: we predefine the number of iterations to run for solving each subproblems and stop worrying about the stopping condition. For example, when $\mu > 0$ and $\mathcal{M}$ is deterministic, we can simply run $T_{\mathcal{M}}$ iterations of $\mathcal{M}$ for each subproblem where $T_{\mathcal{M}}$ is greater than the value given by Corollaries 13 or 16, then the criterions (C1) and (C2) are guaranteed to be satisfied. Unfortunately, the theoretical bound of $T_{\mathcal{M}}$ is relatively poor and does not lead to a practical strategy. On the other hand, using a more aggressive strategy such as $T_{\mathcal{M}} = n$ for incremental algorithms, meaning one pass over the data, seems to provide outstanding results, as shown in the experimental part of this paper.

Finally, one could argue that choosing κ according to a worst-case convergence analysis is not necessarily a good choice. In particular, the convergence rate of the method $\mathcal{M}$, driven by the parameter $\tau_{\mathcal{M}}$ is probably often under estimated in the first place. This suggests that using a smaller value for κ than the one we have advocated earlier is a good thing. In practice, we have observed that indeed Catalyst is often robust to smaller values of κ than the theoretical one, but we have also observed that the theoretical value performs reasonably well, as we shall see in the next section.

5. Experimental Study

In this section, we conduct various experiments to study the effect of the Catalyst acceleration and its different variants, showing in particular how to accelerate SVRG, SAGA, and MISO. In Section 5.1, we describe the data sets and formulations considered for our evaluation, and in Section 5.2, we present the different variants of Catalyst. Then, we study different questions: which variant of Catalyst should we use for incremental approaches? (Section 5.3); how do various incremental methods compare when accelerated with Catalyst? (Section 5.4); what is the effect of Catalyst on the test error when Catalyst is used to minimize a regularized empirical risk? (Section 5.5); is the theoretical value for κ appropriate? (Section 5.6). The code used for all our experiments is available at `https://github.com/hongzhoulin89/Catalyst-QNing/`.

5.1 Data sets, Formulations, and Metric

Data sets. We consider six machine learning data sets with different characteristics in terms of size and dimension to cover a variety of situations.

name	covtype	alpha	real-sim	rcv1	MNIST-CKN	CIFAR-CKN
n	581 012	250 000	72 309	781 265	60 000	50 000
d	54	500	20 958	47 152	2 304	9 216

While the first four data sets are standard ones that were used in previous work about optimization methods for machine learning, the last two are coming from a computer vision application. MNIST and CIFAR-10 are two image classification data sets involving 10 classes. The feature representation of each image was computed using an unsupervised convolutional kernel network Mairal (2016). We focus here on the the task of classifying class #1 vs. the rest of the data set.

Formulations. We consider three common optimization problems in machine learning and signal processing, which admit a particular structure (large finite sum, composite, strong convexity). For each formulation, we also consider a training set $(b_i, a_i)_{i=1}^n$ of n data points, where the b_i's are scalars in $\{-1, +1\}$ and the a_i are feature vectors in $\mathbb{R}^p$. Then, the goal is to fit a linear model x in $\mathbb{R}^p$ such that the scalar b_i can be well predicted by the inner-product $\approx a_i^\top x$, or by its sign. Specifically, the three formulations we consider are listed below.

- **ℓ_2^2-regularized Logistic Regression**:

$$\min_{x \in \mathbb{R}^p} \quad \frac{1}{n} \sum_{i=1}^n \log\left(1 + \exp(-b_i\, a_i^T x)\right) + \frac{\mu}{2} \|x\|^2,$$

  which leads to a μ-strongly convex smooth optimization problem.

- **ℓ_1-regularized Linear Regression (LASSO)**:

$$\min_{x \in \mathbb{R}^p} \quad \frac{1}{2n} \sum_{i=1}^n (b_i - a_i^T x)^2 + \lambda \|x\|_1,$$

  which is non smooth and convex but not strongly convex.

- **$\ell_1 - \ell_2^2$-regularized Linear Regression (Elastic-Net)**:

$$\min_{x \in \mathbb{R}^p} \quad \frac{1}{2n} \sum_{i=1}^n (b_i - a_i^T x)^2 + \lambda \|x\|_1 + \frac{\mu}{2} \|x\|^2,$$

  which is based on the Elastic-Net regularization (Zou and Hastie, 2005) and leading to a strongly-convex optimization problem.

Each feature vector a_i is normalized, and a natural upper-bound on the Lipschitz constant L of the un-regularized objective can be easily obtained with $L_{\text{logistic}} = 1/4$ and $L_{\text{lasso}} = 1$. The regularization parameter μ and λ are choosing in the following way:

- For **Logistic Regression**, we find an optimal regularization parameter μ^* by 10-fold cross validation for each data set on a logarithmic grid $2^i/n$, with $i \in [-12, 3]$. Then, we set $\mu = \mu^*/2^3$ which corresponds to a small value of the regularization parameter and a relatively ill-conditioned problem.

- For **Elastic-Net**, we set $\mu = 0.01/n$ to simulate the ill-conditioned situation and add a small l_1-regularization penalty with $\lambda = 1/n$ that produces sparse solutions.

- For the **Lasso problem**, we consider a logarithmic grid $10^i/n$, with $i = -3, -2, \dots, 3$, and we select the parameter λ that provides a sparse optimal solution closest to 10% non-zero coefficients, which leads to $\lambda = 10/n$ or $100/n$.

Note that for the strongly convex problems, the regularization parameter μ yields a lower bound on the strong convexity parameter of the problem.

Metric used. In this chapter, and following previous work about incremental methods (Schmidt et al., 2017), we plot objective values as a function of the number of gradients evaluated during optimization, which appears to be the computational bottleneck of all previously mentioned algorithms. Since no metric is perfect for comparing algorithms' speed, we shall make the two following remarks, such that the reader can interpret our results and the limitations of our study with no difficulty.

- Ideally, CPU-time is the gold standard but CPU time is implementation-dependent and hardware-dependent.
- We have chosen to count only gradients computed with random data access. Thus, computing n times a gradient f_i by picking each time one function at random counts as "n gradients", whereas we ignore the cost of computing a full gradient $(1/n)\sum_{i=1}^{n}\nabla f_i$ at once, where the f_i's can be accessed in sequential order. Similarly, we ignore the cost of computing the function value $f(x) = (1/n)\sum_{i=1}^{n} f_i(x)$, which is typically performed every pass on the data when computing a duality gap. While this assumption may be inappropriate in some contexts, the cost of random gradient computations was significantly dominating the cost of sequential access in our experiments, where (i) data sets fit into memory; (ii) computing full gradients was done in C++ by calling BLAS2 functions exploiting multiple cores.

5.2 Choice of Hyper-parameters and Variants

Before presenting the numerical results, we discuss the choice of default parameters used in the experiments as well as different variants.

Choice of method $\mathcal{M}$. We consider the acceleration of incremental algorithms which are able to adapt to the problem structure we consider: large sum of functions and possibly non-smooth regularization penalty.

- The proximal SVRG algorithm of Xiao and Zhang (2014) with stepsize $\eta = 1/L$.
- The SAGA algorithm Defazio et al. (2014a) with stepsize $\eta = 1/3L$.
- The proximal MISO algorithm of Lin et al. (2015a).

Choice of regularization parameter κ. As suggested by the theoretical analysis, we take κ to minimize the global complexity, leading to the choice

$$\kappa = \frac{L-\mu}{n+1} - \mu.$$

Stopping criteria for the inner loop. The choice of the accuracies are driven from the theoretical analysis described in paragraph 3. Here, we specify it again for the clarity of presentation:

- **Stopping criterion (C1).** Stop when $h_k(z_t) - h_k^* \leq \varepsilon_k$, where

$$\varepsilon_k = {}^{5}\begin{cases} \frac{1}{2}(1-\rho)^k f(x_0) \ \text{ with } \ \rho = 0.9\sqrt{\frac{\mu}{\mu+\kappa}} & \text{when } \mu > 0; \\ \frac{f(x_0)}{2(k+1)^{4.1}} & \text{when } \mu = 0. \end{cases}$$

The duality gap $h(w_t) - h^*$ can be estimated either by evaluating the Fenchel conjugate function or by computing the squared norm of the gradient.

- **Stopping criterion (C2).** Stop when $h_k(z_t) - h_k^* \leq \delta_k \cdot \frac{\kappa}{2}\|z_t - y_{k-1}\|^2$, where

$$\delta_k = \begin{cases} \frac{\sqrt{q}}{2-\sqrt{q}} \text{ with } q = \frac{\mu}{\mu+\kappa} & \text{when } \mu > 0; \\ \frac{1}{(k+1)^2} & \text{when } \mu = 0. \end{cases}$$

- **Stopping criterion (C3) .** Perform exactly one pass over the data in the inner loop without checking any stopping criteria.[4]

Warm start for the inner loop. This is an important point to achieve acceleration which was not highlighted in the conference paper (Lin et al., 2015a). At iteration $k+1$, we consider the minimization of

$$h_{k+1}(z) = f_0(z) + \frac{\kappa}{2}\|z - y_k\|^2 + \psi(z).$$

We warm start according to the strategy defined in Section 3. Let x_k be the approximate minimizer of h_k, obtained from the last iteration.

- **Initialization for (C1).** Let us define $\eta = \frac{1}{L+\kappa}$, then initialize at

$$z_0^{C1} = \begin{cases} w_0 \triangleq x_k + \frac{\kappa}{\kappa+\mu}(y_k - y_{k-1}) & \text{if } \psi = 0; \\ [w_0]_\eta & \text{otherwise.} \end{cases}$$

  where $[w_0]_\eta = \text{prox}_{\eta\psi}(w_0 - \eta g)$ with $g = \nabla f_0(w_0) + \kappa(w_0 - y_k)$.

- **Initialization for (C2).** Intialize at

$$z_0^{C2} = \begin{cases} y_k & \text{if } \psi = 0; \\ [y_k]_\eta = \text{prox}_{\eta\psi}(y_k - \eta\nabla f_0(y_k)) & \text{otherwise.} \end{cases}$$

- **Initialization for (C3).** Take the best initial point among x_k and z_0^{C1}

$$z_0^{C3} \text{ such that } h_k(z_0^{C3}) = \min\{h_k(x_{k-1}), h_k(z_0^{C1})\}.$$

- **Initialization for (C1*).** Use the strategy (C1) with z_0^{C3}.

The warm start at z_0^{C3} requires to choose the best point between the last iterate x_k and the point z_0^{C1}. The motivation is that since the one-pass strategy is an aggressive heuristic, the solution of the subproblems may not be as accurate as the ones obtained with other criterions. Allowing using the iterate x_k turned out to be significantly more stable in practice. Then, it is also natural to use a similar strategy for criterion (C1), which we call (C1*). Using a similar strategy for (C2) turned out not to provide any benefit in practice and is thus omitted from the list here.

[5] Here we upper bound $f(x_0) - f^*$ by $f(x_0)$ since f is always positive in our models.

[4] This stopping criterion is heuristic since one pass may not be enough to achieve the required accuracy. What we have shown is that with a large enough $T_{\mathcal{M}}$, then the convergence will be guaranteed. Here we take heuristically $T_{\mathcal{M}}$ as one pass.

5.3 Comparison of Stopping Criteria and Warm-start Strategies

First, we evaluate the performance of the previous strategies when applying Catalyst to SVRG, SAGA and MISO. The results are presented in Figures 1, 2, and 3, respectively.

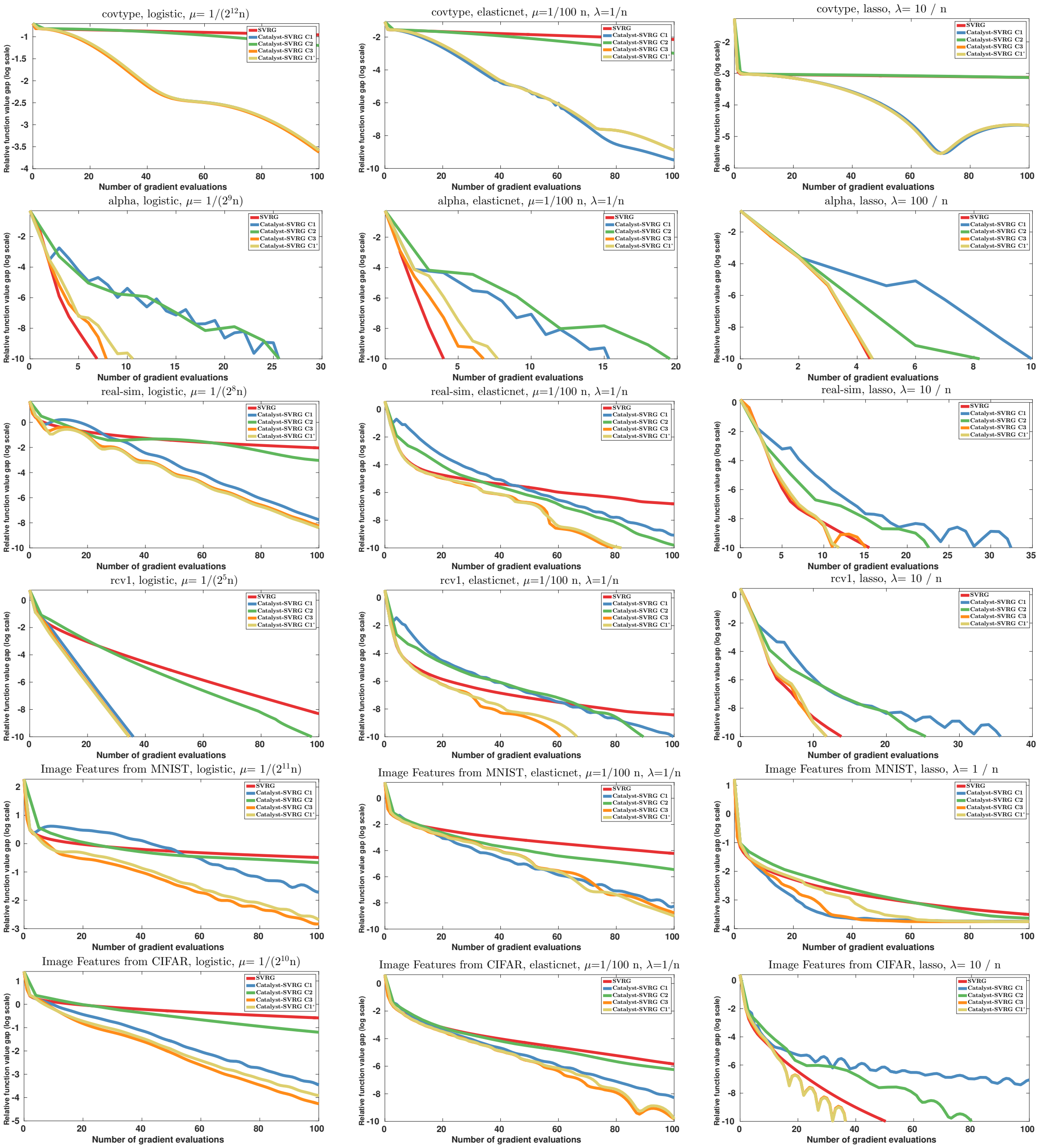


Figure 1: Experimental study of different stopping criterions for Catalyst-SVRG. We plot the value $f(x_k)/f^* - 1$ as a function of the number of gradient evaluations, on a logarithmic scale; the optimal value f^* is estimated with a duality gap.

Observations for Catalyst-SVRG. We remark that in most of the cases, the curve of (C3) and (C1*)are superimposed, meaning that one pass through the data is enough for solving the subproblem up to the required accuracy. Moreover, they give the best performance among all criterions. Regarding the logistic regression problem, the acceleration is significant (even huge for the covtype data set) except for alpha, where only (C3) and (C1*)do not degrade significantly the performance. For sparse problems, the effect of acceleration is more mitigated, with 7 cases out of 12 exhibiting important acceleration and 5 cases no acceleration. As before, (C3) and (C1*)are the only strategies that never degrade performance.

One reason explaining why acceleration is not systematic may be the ability of incremental methods to adapt to the unknown strong convexity parameter $\mu' \geq \mu$ hidden in the objective's loss, or local strong convexity near the solution. When $\mu'/L \geq 1/n$, we indeed obtain a well-conditioned regime where acceleration should not occur theoretically. In fact the complexity $O(n \log(1/\varepsilon))$ is already optimal in this regime, see Arjevani and Shamir (2016); Woodworth and Srebro (2016). For sparse problems, conditioning of the problem with respect to the linear subspace where the solution lies might also play a role, even though our analysis does not study this aspect. Therefore, this experiment suggests that adaptivity to unknown strong convexity is of high interest for incremental optimization.

Observations for Catalyst-SAGA. Our conclusions with SAGA are almost the same as with SVRG. However, in a few cases, we also notice that criterion C1 lacks stability, or at least exhibits some oscillations, which may suggest that SAGA has a larger variance compared to SVRG. The difference in the performance of (C1) and (C1*)can be huge, while they differ from each other only by the warm start strategy. Thus, *choosing a good initial point for solving the sub-problems is a key for obtaining acceleration in practice.*

Observations for Catalyst-MISO. The warm-start strategy of MISO is different from primal algorithms because parameters for the dual function need to be specified. The most natural way for warm starting the dual functions is to set

$$d_{k+1}(x) = d_k(x) + \frac{\kappa}{2}\|x - y_k\|^2 - \frac{\kappa}{2}\|x - y_{k-1}\|^2,$$

where d_k is the last dual function of the previous subproblem h_k. This gives the warm start

$$z_0 = \text{prox}\left(x_k + \frac{\kappa}{\kappa + \mu}(y_k - y_{k-1})\right).$$

For other choices of z_0, the dual function needs to be recomputed from scratch, which is computationally expensive and unstable for ill-conditioned problems. Thus, we only present the experimental results with respect to criterion (C1) and the one-pass heuristic (C3) . As we observe, a huge acceleration is obtained in logistic regression and Elastic-net formulations. For Lasso problem, the original Prox-MISO is not defined since the problem is not strongly convex. Thus, in order to make a comparison, we compare with Catalyst-SVRG which shows that the acceleration achieves a similar performance. This aligns with the theoretical result stating that Catalyst applied to incremental algorithms yields a similar convergence rate. Notice also that the original MISO algorithm suffers from numerical stability in this ill-conditioned regime chosen for our experiments. Catalyst not only accelerates MISO, but it also stabilizes it.

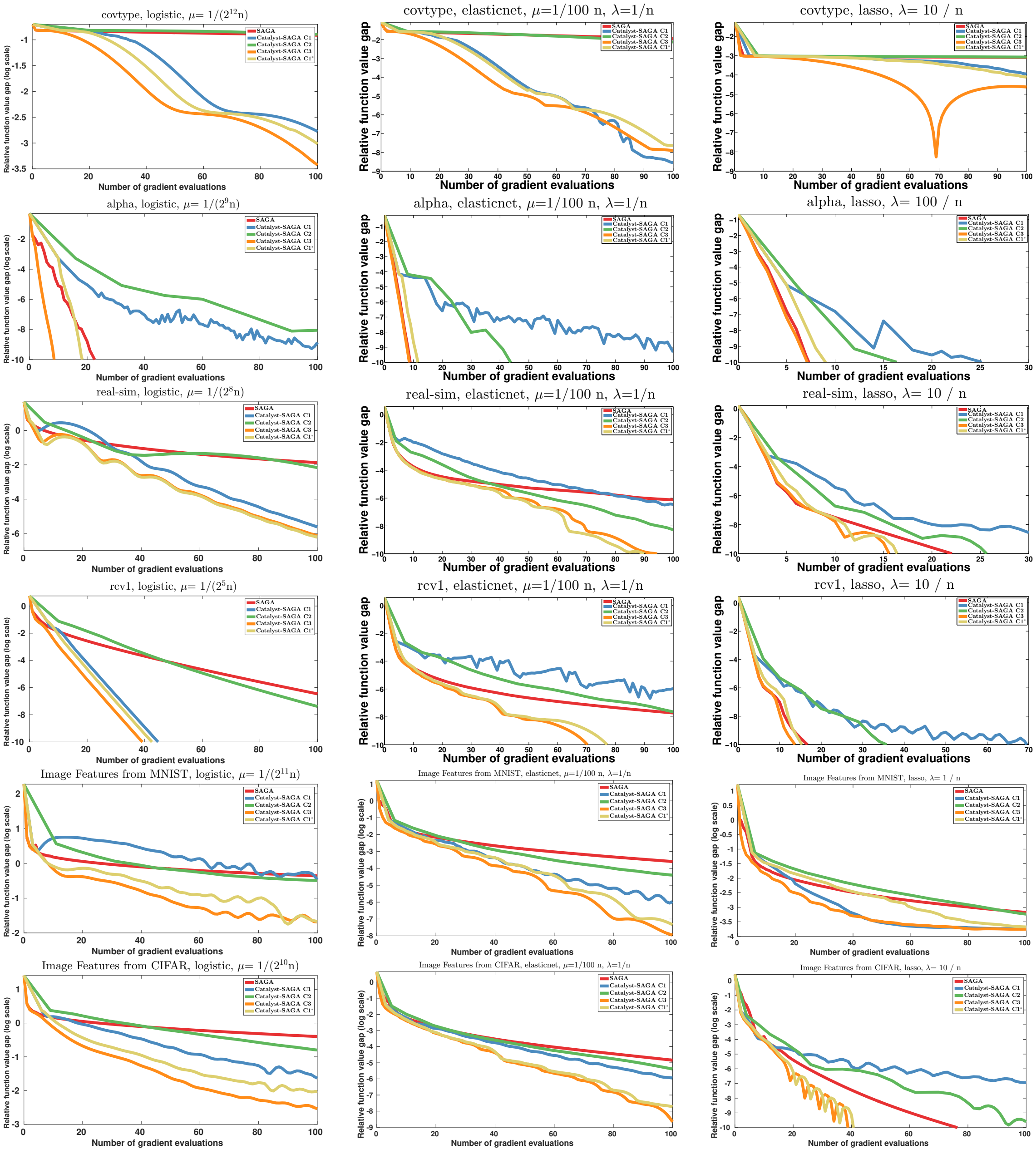


Figure 2: Experimental study of different stopping criterions for Catalyst-SAGA, with a similar setting as in Figure 1.

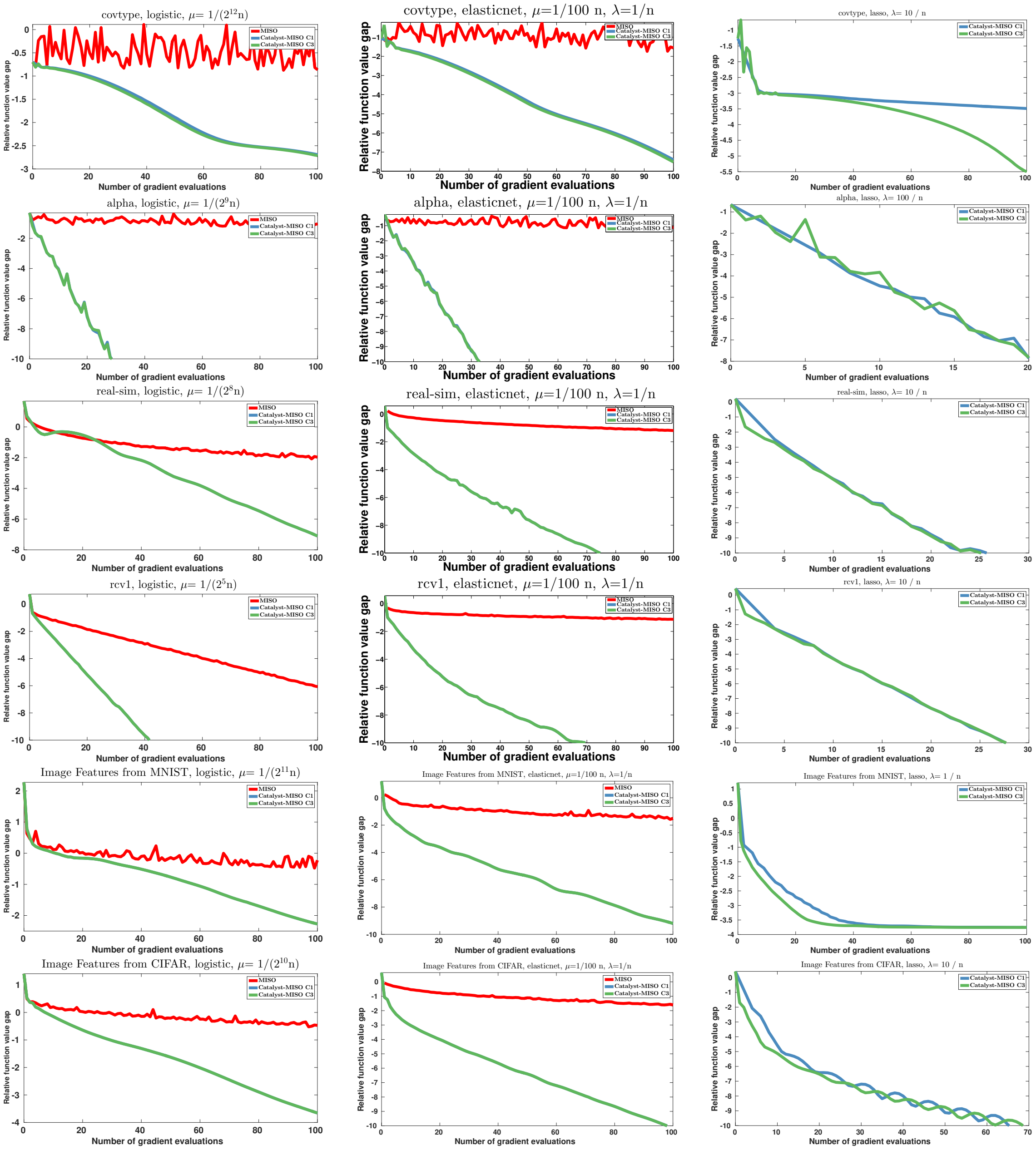


Figure 3: Experimental study of different stopping criterions for Catalyst-MISO, with a similar setting as in Figure 1

5.4 Acceleration of Existing Methods

Then, we put the previous curves into perspective and make a comparison of the performance before and after applying Catalyst across methods. We show the best performance among the three developed stopping criteria, which corresponds to be (C3) .

Observations. In Figure 4, we observe that by applying Catalyst, we accelerate the original algorithms up to the limitations discussed above (comparing the dashed line and the solid line of the same color). In three data sets (covtype, real-sim and rcv1), significant improvements are achieved as expected by the theory for the ill-conditioned problems in logistic regression and Elastic-net. For data set alpha, we remark that an relative accuracy in the order 10^{-10} is attained in less than 10 iterations. This suggests that the problems is in fact well-conditioned and there is some hidden strong convexity for this data set. Thus, the incremental algorithms like SVRG or SAGA are already optimal under this situation and no further improvement can be obtained by applying Catalyst.

5.5 Empirical Effect on the Generalization Error

A natural question that applies to all incremental methods is whether or not the acceleration that we may see for minimizing an empirical risk on *training* data affects the objective function and the test accuracy on new unseen *test* data. To answer this question, we consider the logistic regression formulation with the regularization parameter μ^* obtained by cross-validation. Then, we cut each data set into 80% of training data and set aside 20% of the data point as test data.

Observations on the test loss and test accuracy. The left column of Figure 5 shows the loss function on the training set, where acceleration is significant in 5 cases out of 6. The middle column shows the loss function evaluated on the test set, but on a non-logarithmic scale since the optimal value of the test loss is unknown. Acceleration appears in 4 cases out of 6. Regarding the test accuracy, an important acceleration is obtained in 2 cases, whereas it is less significant or negligible in the other cases.

5.6 Study of the Parameter κ

Finally, we evaluate the performance for different values of κ.

Observations for different choices of κ. We consider a logarithmic grid $\kappa = 10^i\kappa_0$ with $i = -2, -1, \cdots, 2$ and κ_0 is the optimal κ given by the theory. We observe that for ill-conditioned problems, using optimal choice κ_0 provides much better performance than other choices, which confirms the theoretical result. For the data set of alpha or Lasso problems, we observe that the best choice is given by the smallest $\kappa = 0.01\kappa_0$. This suggests that, as discussed before, there is a certain degree of strong convexity present in the objective even without any regularization.

6. Conclusion

We have introduced a generic algorithm called Catalyst that allows us to extend Nesterov's acceleration to a large class of first-order methods. We have shown that it can be effective

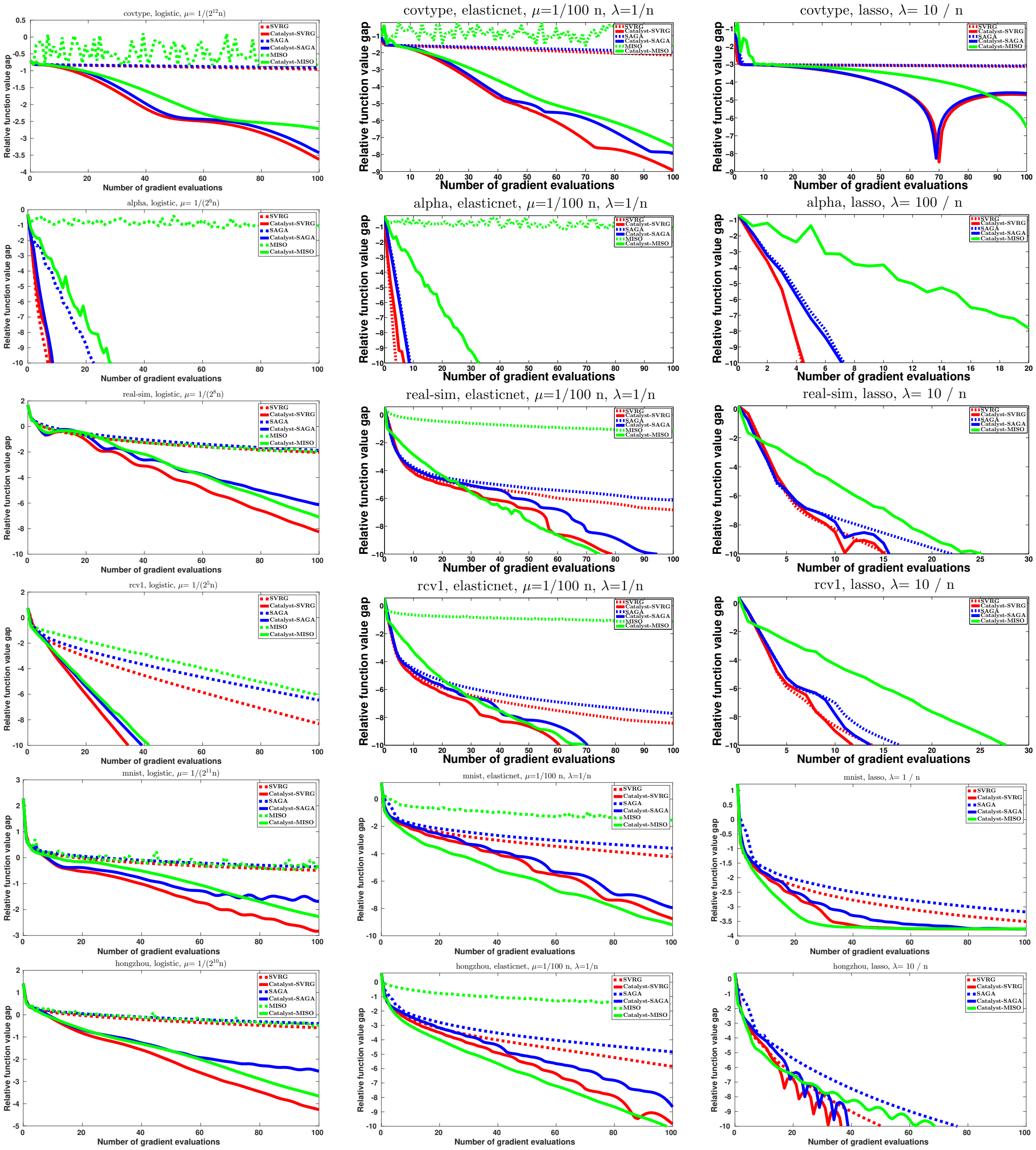


Figure 4: Experimental study of the performance of Catalyst applying to SVRG, SAGA and MISO. The dashed lines correspond to the original algorithms and the solid lines correspond to accelerated algorithms by applying Catalyst. We plot the relative function value gap $(f(x_k) - f^*)/f^*$ in the number of gradient evaluations, on a logarithmic scale.

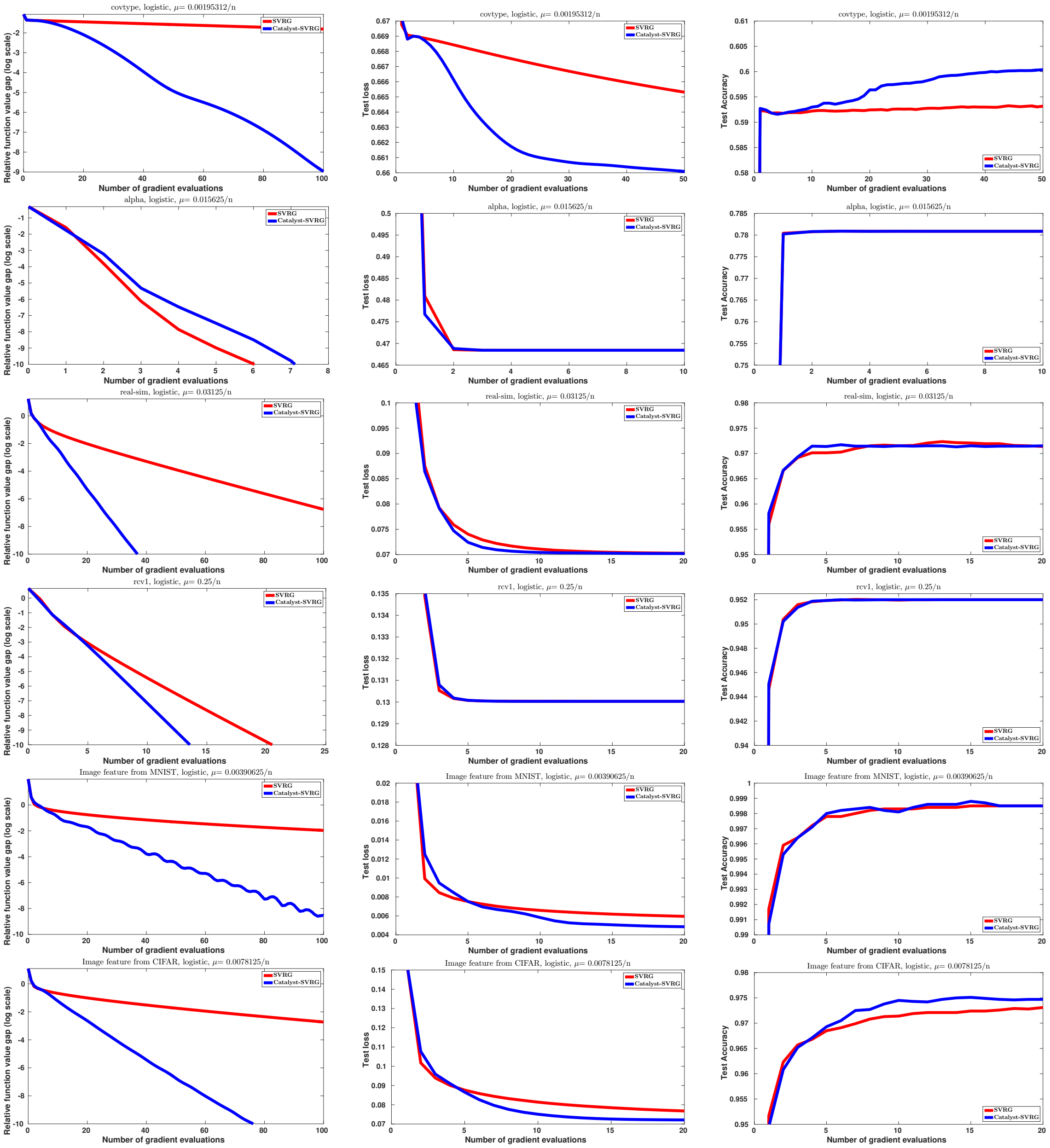
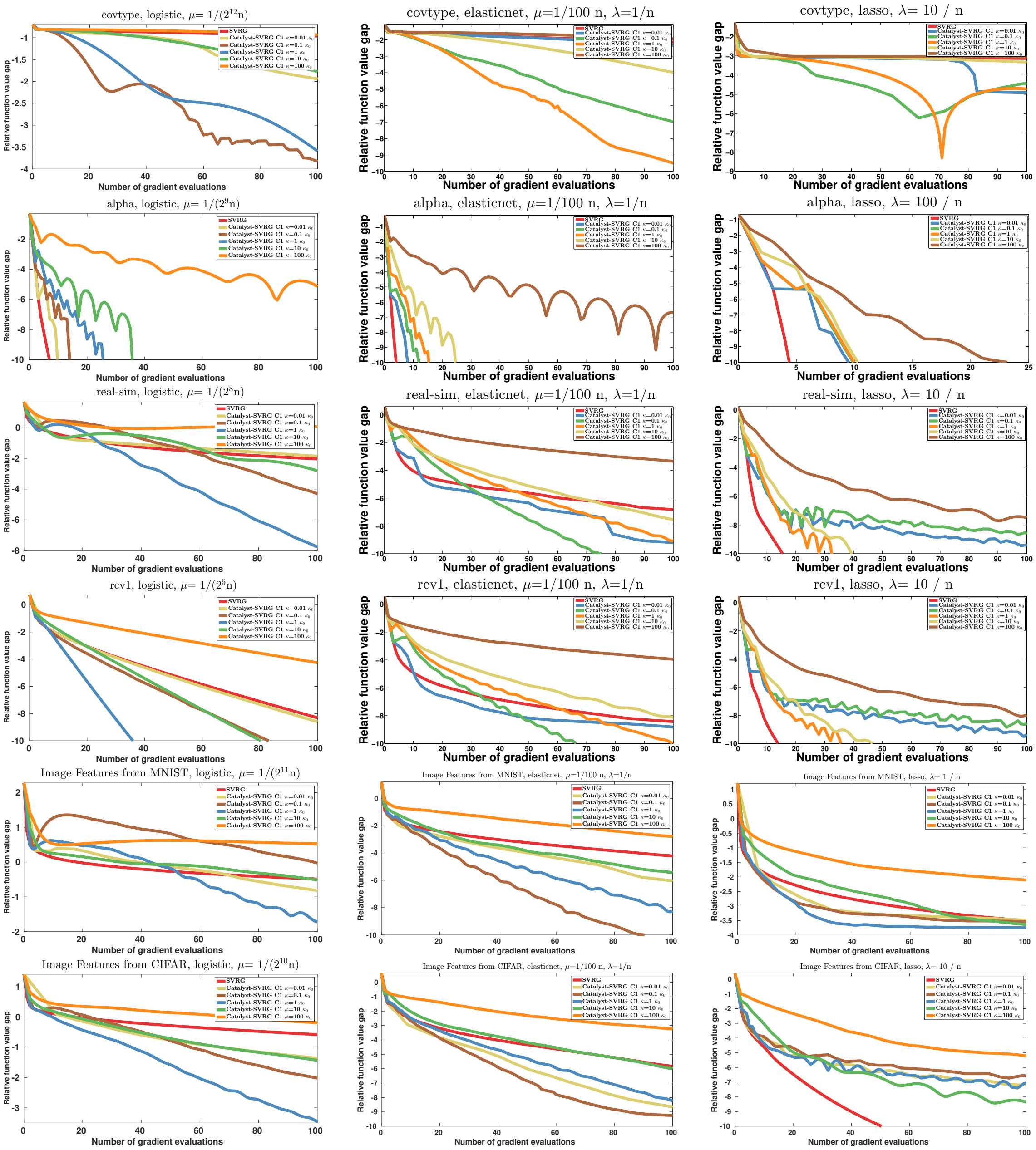


Figure 5: Empirical effect on the generalization error. For a logistic regression experiment, we report the value of the objective function evaluated on the training data on the left column, the value of the loss evaluated on a test set on the middle column, and the classification error evaluated on the test set on the right.



covtype, logistic, $\mu = 1/(2^{12}\text{n})$

covtype, elasticnet, μ=1/100 n, λ=1/n

covtype, lasso, λ= 10 / n

alpha, logistic, $\mu = 1/(2^{9}\text{n})$

alpha, elasticnet, μ=1/100 n, λ=1/n

alpha, lasso, λ= 100 / n

real-sim, logistic, $\mu = 1/(2^{8}\text{n})$

real-sim, elasticnet, μ=1/100 n, λ=1/n

real-sim, lasso, λ= 10 / n

rcv1, logistic, $\mu = 1/(2^{5}\text{n})$

rcv1, elasticnet, μ=1/100 n, λ=1/n

rcv1, lasso, λ= 10 / n

Image Features from MNIST, logistic, $\mu = 1/(2^{11}\text{n})$

Image Features from MNIST, elasticnet, μ=1/100 n, λ=1/n

Image Features from MNIST, lasso, λ= 1 / n

Image Features from CIFAR, logistic, $\mu = 1/(2^{10}\text{n})$

Image Features from CIFAR, elasticnet, μ=1/100 n, λ=1/n

Image Features from CIFAR, lasso, λ= 10 / n

Relative function value gap
Number of gradient evaluations
SVRG
Catalyst-SVRG C1 κ=0.01 κ0
Catalyst-SVRG C1 κ=0.1 κ0
Catalyst-SVRG C1 κ=1 κ0
Catalyst-SVRG C1 κ=10 κ0
Catalyst-SVRG C1 κ=100 κ0

Figure 6: Evaluations of Catalyst-SVRG for different κ using stopping criterion $\mathsf{C1}$, where κ_0 is the theoretical choice given by the complexity analysis.

in practice for ill-conditioned problems. Besides acceleration, Catalyst also improves the numerical stability of a given algorithm, by applying it to auxiliary problems that are better conditioned than the original objective. For this reason, it also provides support to convex, but not strongly convex objectives, to algorithms that originally require strong convexity. We have also studied experimentally many variants to identify the ones that are the most effective and the simplest to use in practice. For incremental methods, we showed that the "almost-parameter-free" variant, consisting in performing a single pass over the data at every outer-loop iteration, was the most effective one in practice.

Even though we have illustrated Catalyst in the context of finite-sum optimization problems, the main feature of our approach is its versatility. Catalyst could also be applied to other algorithms that have not been considered so far and give rise to new accelerated algorithms.

Acknowledgments

The authors would like to thank Dmitriy Drusvyatskiy, Anatoli Juditsky, Sham Kakade, Arkadi Nemirovski, Courtney Paquette, and Vincent Roulet for fruitful discussions. Hongzhou Lin and Julien Mairal were supported by the ERC grant SOLARIS (# 714381) and a grant from ANR (MACARON project ANR-14-CE23-0003-01). Zaid Harchaoui was supported by NSF Award CCF-1740551 and the program "Learning in Machines and Brains" of CIFAR. This work was performed while Hongzhou Lin was at Inria and Univ. Grenoble Alpes.

Appendix A. Useful Lemmas

Lemma 19 (Simple lemma on quadratic functions) *For all vectors x, y, z in $\mathbb{R}^p$ and $\theta > 0$,*

$$\|x - y\|^2 \geq (1 - \theta)\|x - z\|^2 + \left(1 - \frac{1}{\theta}\right)\|z - y\|^2.$$

Proof

$$\begin{aligned}
\|x - y\|^2 &= \|x - z + z - y\|^2 \\
&= \|x - z\|^2 + \|z - y\|^2 + 2\langle x - z, z - y\rangle \\
&= \|x - z\|^2 + \|z - y\|^2 + \left\|\sqrt{\theta}(x - z) + \frac{1}{\sqrt{\theta}}(z - y)\right\|^2 - \theta\|x - z\|^2 - \frac{1}{\theta}\|z - y\|^2 \\
&\geq (1 - \theta)\|x - z\|^2 + \left(1 - \frac{1}{\theta}\right)\|z - y\|^2.
\end{aligned}$$

■

Lemma 20 (Simple lemma on non-negative sequences) *Consider a increasing sequence $(S_k)_{k\geq 0}$ and two non-negative sequences $(a_k)_{k\geq 0}$ and $(u_k)_{k\geq 0}$ such that for all k,*

$$u_k^2 \leq S_k + \sum_{i=1}^{k} a_i u_i. \tag{29}$$

Then,

$$S_k + \sum_{i=1}^{k} a_i u_i \leq \left(\sqrt{S_k} + \sum_{i=1}^{k} a_i \right)^2. \tag{30}$$

Proof This lemma is identical to the Lemma A.10 in the original Catalyst paper (Lin et al., 2015a), inspired by a lemma of Schmidt et al. (2011) for controlling errors of inexact proximal gradient methods.

We give here an elementary proof for completeness based on induction. The relation (30) is obviously true for $k = 0$. Then, we assume it is true for $k - 1$ and prove the relation for k. We remark that from (29),

$$\left(u_k - \frac{a_k}{2}\right)^2 \leq S_k + \sum_{i=1}^{k-1} a_i u_i + \frac{a_k^2}{4},$$

and then

$$u_k \leq \sqrt{S_k + \sum_{i=1}^{k-1} a_i u_i + \frac{a_k^2}{4}} + \frac{a_k}{2}.$$

We may now prove the relation (30) by induction,

$$\begin{aligned}
S_k + \sum_{i=1}^{k} a_i u_i &\leq S_k + \sum_{i=1}^{k-1} a_i u_i + a_k \left(\frac{a_k}{2} + \sqrt{S_k + \sum_{i=1}^{k-1} a_i u_i + \frac{a_k^2}{4}} \right) \\
&\leq S_k + \sum_{i=1}^{k-1} a_i u_i + a_k \left(a_k + \sqrt{S_k + \sum_{i=1}^{k-1} a_i u_i} \right) \\
&\leq \left(\sqrt{S_k + \sum_{i=1}^{k-1} a_i u_i} + a_k \right)^2 \\
&= \left(\sqrt{(S_k - S_{k-1}) + (S_{k-1} + \sum_{i=1}^{k-1} a_i u_i)} + a_k \right)^2 \\
&\leq \left(\sqrt{(S_k - S_{k-1}) + \left(\sqrt{S_{k-1}} + \sum_{i=1}^{k-1} a_i \right)^2} + a_k \right)^2 \qquad \text{(by induction)} \\
&\leq \left(\sqrt{S_k} + \sum_{i=1}^{k} a_i \right)^2.
\end{aligned}$$

The last inequality is obtained by developing the square $\left(\sqrt{S_{k-1}} + \sum_{i=1}^{k-1} a_i\right)^2$ and use the increasing assumption $S_{k-1} \leq S_k$. ■

Lemma 21 (Growth of the sequence $(A_k)_{k\geq 0}$)
Let $(A_k)_{k\geq 0}$ be the sequence defined in (14) where $(\alpha_k)_{k\geq 0}$ is produced by (10) with $\alpha_0 = 1$ and $\mu = 0$. Then, we have the following bounds for all $k \geq 0$,

$$\frac{2}{(k+2)^2} \leq A_k \leq \frac{4}{(k+2)^2}.$$

Proof The righthand side is directly obtained from Lemma 4 by noticing that $\gamma_0 = \kappa$ with the choice of α_0. Using the recurrence of α_k, we have for all $k \geq 1$,

$$\alpha_k^2 = (1-\alpha_k)\alpha_{k-1}^2 = \prod_{i=1}^{k}(1-\alpha_i)\alpha_0^2 = A_k \leq \frac{4}{(k+2)^2}.$$

Thus, $\alpha_k \leq \frac{2}{k+2}$ for all $k \geq 1$ (it is also true for $k = 0$). We now have all we need to conclude the lemma:

$$A_k = \prod_{i=1}^{k}(1-\alpha_i) \geq \prod_{i=1}^{k}\left(1 - \frac{2}{i+2}\right) = \frac{2}{(k+2)(k+1)} \geq \frac{2}{(k+2)^2}.$$

■

Appendix B. Proofs of Auxiliary Results

B.1 Proof of Lemma 2

Proof Let us introduce the notation $h'(z) \triangleq \frac{1}{\eta}(z - [z]_\eta)$ for the gradient mapping at z. The first order conditions of the convex problem defining $[z]_\eta$ give

$$h'(z) - \nabla h_0(z) \in \partial\psi([z]_\eta).$$

Then, we may define

$$\begin{aligned} u &\triangleq \frac{1}{\eta}(z - [z]_\eta) - (\nabla h_0(z) - \nabla h_0([z]_\eta)), \\ &= h'(z) - \nabla h_0(z) + \nabla h_0([z]_\eta) \in \partial h([z]_\eta). \end{aligned}$$

Then, by strong convexity,

$$\begin{aligned} h^* &\geq h([z]_\eta) + u^\top(p(x) - [z]_\eta) + \frac{\kappa+\mu}{2}\|p(x) - [z]_\eta\|^2 \\ &\geq h([z]_\eta) - \frac{1}{2(\kappa+\mu)}\|u\|^2. \end{aligned}$$

Moreover,

$$\begin{aligned} \|u\|^2 &= \left\|\frac{1}{\eta}(z - [z]_\eta)\right\|^2 - \frac{2}{\eta}\langle z - [z]_\eta, \nabla h_0(z) - \nabla h_0([z]_\eta)\rangle + \|\nabla h_0(z) - \nabla h_0([z]_\eta)\|^2 \\ &\leq \left\|h'(z)\right\|^2 - \|\nabla h_0(z) - \nabla h_0([z]_\eta)\|^2 \\ &\leq \left\|h'(z)\right\|^2, \end{aligned}$$

where the first inequality comes from the relation (Nesterov, 2004, Theorem 2.1.5) using the fact h_0 is $(1/\eta)$-smooth

$$\|\nabla h_0(z) - \nabla h_0([z]_\eta)\|^2 \leq \frac{1}{\eta}\langle z - [z]_\eta, \nabla h_0(z) - \nabla h_0([z]_\eta)\rangle.$$

Thus,

$$h([z]_\eta) - h^* \leq \frac{1}{2(\kappa+\mu)}\|u\|^2 \leq \frac{1}{2(\kappa+\mu)}\|h'(z)\|^2.$$

As a result,

$$\|h'(z)\| \leq \sqrt{2\kappa\varepsilon} \quad \Rightarrow \quad h([z]_\eta) - h^* \leq \varepsilon.$$

■

B.2 Proof of Proposition 5

Proof We simply use Theorem 3 and specialize it to the choice of parameters. The initialization $\alpha_0 = \sqrt{q}$ leads to a particularly simple form of the algorithm, where $\alpha_k = \sqrt{q}$ for all $k \geq 0$. Therefore, the sequence $(A_k)_{k\geq 0}$ from Theorem 3 is also simple since we indeed have $A_k = (1-\sqrt{q})^k$. Then, we remark that $\gamma_0 = \mu(1-\sqrt{q})$ and thus, by strong convexity of f,

$$S_0 = (1-\sqrt{q})\left(f(x_0) - f^* + \frac{\mu}{2}\|x_0 - x^*\|^2\right) \leq 2(1-\sqrt{q})(f(x_0) - f^*).$$

Therefore,

$$\begin{aligned}
\sqrt{S_0} + 3\sum_{j=1}^{k}\sqrt{\frac{\varepsilon_j}{A_{j-1}}} &\leq \sqrt{2(1-\sqrt{q})(f(x_0)-f^*)} + 3\sum_{j=1}^{k}\sqrt{\frac{\varepsilon_j}{A_{j-1}}} \\
&= \sqrt{2(1-\sqrt{q})(f(x_0)-f^*)}\left[1 + \sum_{j=1}^{k}\underbrace{\left(\sqrt{\frac{1-\rho}{1-\sqrt{q}}}\right)}_{\eta}^{j}\right] \\
&= \sqrt{2(1-\sqrt{q})(f(x_0)-f^*)}\,\frac{\eta^{k+1}-1}{\eta-1} \\
&\leq \sqrt{2(1-\sqrt{q})(f(x_0)-f^*)}\,\frac{\eta^{k+1}}{\eta-1}.
\end{aligned}$$

Therefore, Theorem 3 combined with the previous inequality gives us

$$\begin{aligned}
f(x_k) - f^* &\leq 2A_{k-1}(1-\sqrt{q})(f(x_0)-f^*)\ \left(\frac{\eta^{k+1}}{\eta-1}\right)^2 \\
&= 2\left(\frac{\eta}{\eta-1}\right)^2 (1-\rho)^k (f(x_0)-f^*) \\
&= 2\left(\frac{\sqrt{1-\rho}}{\sqrt{1-\rho}-\sqrt{1-\sqrt{q}}}\right)^2 (1-\rho)^k (f(x_0)-f^*) \\
&= 2\left(\frac{1}{\sqrt{1-\rho}-\sqrt{1-\sqrt{q}}}\right)^2 (1-\rho)^{k+1} (f(x_0)-f^*).
\end{aligned}$$

Since $\sqrt{1-x}+\frac{x}{2}$ is decreasing in $[0,1]$, we have $\sqrt{1-\rho}+\frac{\rho}{2} \geq \sqrt{1-\sqrt{q}}+\frac{\sqrt{q}}{2}$. Consequently,

$$f(x_k) - f^* \leq \frac{8}{(\sqrt{q}-\rho)^2}(1-\rho)^{k+1}(f(x_0)-f^*).$$

■

B.3 Proof of Proposition 6

Proof The initialization $\alpha_0 = 1$ leads to $\gamma_0 = \kappa$ and $S_0 = \frac{\kappa}{2}\|x^* - x_0\|^2$. Then,

$$\begin{aligned}
\sqrt{\frac{\gamma_0}{2}\|x_0 - x^*\|^2} + 3\sum_{j=1}^{k}\sqrt{\frac{\varepsilon_j}{A_{j-1}}} &\leq \sqrt{\frac{\kappa}{2}\|x_0 - x^*\|^2} + 3\sum_{j=1}^{k}\sqrt{\frac{(j+1)^2\varepsilon_j}{2}} \qquad \text{(from Lemma 21)} \\
&\leq \sqrt{\frac{\kappa}{2}\|x_0 - x^*\|^2} + \sqrt{f(x_0)-f^*}\left(\sum_{j=1}^{k}\frac{1}{(j+1)^{1+\gamma/2}}\right),
\end{aligned}$$

where the last inequality uses Lemma 21 to upper-bound the ratio ε_j/A_j. Moreover,

$$\sum_{j=1}^{k}\frac{1}{(j+1)^{1+\gamma/2}} \leq \sum_{j=2}^{\infty}\frac{1}{j^{1+\gamma/2}} \leq \int_1^{\infty}\frac{1}{x^{1+\gamma/2}}\,\mathrm{d}x = \frac{2}{\gamma}.$$

Then applying Theorem 3 yields

$$\begin{aligned}
f(x_k) - f^* &\leq A_{k-1}\left(\sqrt{\frac{\kappa}{2}\|x_0 - x^*\|^2} + \frac{2}{\gamma}\sqrt{f(x_0)-f^*}\right)^2 \\
&\leq \frac{8}{(k+1)^2}\left(\frac{\kappa}{2}\|x_0 - x^*\|^2 + \frac{4}{\gamma^2}(f(x_0)-f^*)\right).
\end{aligned}$$

The last inequality uses $(a+b)^2 \leq 2(a^2+b^2)$. ■

B.4 Proof of Lemma 11

Proof We abbreviate $\tau_{\mathcal{M}}$ by τ and $C = C_{\mathcal{M}}(h(z_0) - h^*)$ to simplify the notation. Set

$$T_0 = \frac{1}{\tau} \log \left(\frac{1}{1 - e^{-\tau}} \frac{C}{\varepsilon} \right).$$

For any $t \geq 0$, we have

$$\mathbb{E}[h(z_t) - h^*] \leq C(1 - \tau)^t \leq C\, e^{-t\tau}.$$

By Markov's inequality,

$$\mathbb{P}[h(z_t) - h^* > \varepsilon] = \mathbb{P}[T(\varepsilon) > t] \leq \frac{\mathbb{E}[h(z_t) - h^*]}{\varepsilon} \leq \frac{C\, e^{-t\tau}}{\varepsilon}.$$

Together with the fact $\mathbb{P} \leq 1$ and $t \geq 0$. We have

$$\mathbb{P}[T(\varepsilon) \geq t + 1] \leq \min \left\{ \frac{C}{\varepsilon} e^{-t\tau}, 1 \right\}.$$

Therefore,

$$\begin{aligned} \mathbb{E}[T(\varepsilon)] &= \sum_{t=1}^{\infty} \mathbb{P}[T(\varepsilon) \geq t] = \sum_{t=1}^{T_0} \mathbb{P}[T(\varepsilon) \geq t] + \sum_{t=T_0+1}^{\infty} \mathbb{P}[T(\varepsilon) \geq t] \\ &\leq T_0 + \sum_{t=T_0}^{\infty} \frac{C}{\varepsilon} e^{-t\tau} = T_0 + \frac{C}{\varepsilon} e^{-T_0 \tau} \sum_{t=0}^{\infty} e^{-t\tau} \\ &= T_0 + \frac{C}{\varepsilon} \frac{e^{-\tau T_0}}{1 - e^{-\tau}} = T_0 + 1. \end{aligned}$$

A simple calculation shows that for any $\tau \in (0, 1)$, $\frac{\tau}{2} \leq 1 - e^{-\tau}$ and then

$$\mathbb{E}[T(\varepsilon)] \leq T_0 + 1 = \frac{1}{\tau} \log \left(\frac{1}{1 - e^{-\tau}} \frac{C}{\varepsilon} \right) + 1 \leq \frac{1}{\tau} \log \left(\frac{2C}{\tau \varepsilon} \right) + 1.$$

■

B.5 Proof of coerciveness property of the proximal operator

Lemma 22 *Given a μ-strongly convex function $f : \mathbb{R}^p \to \mathbb{R}$ and a positive parameter $\kappa > 0$. For any x, $y \in \mathbb{R}^p$, the following inequality holds,*

$$\frac{\kappa}{\kappa + \mu} \langle y - x, p(y) - p(x) \rangle \geq \|p(y) - p(x)\|^2,$$

where $p(x) = \operatorname*{arg\,min}_{z \in \mathbb{R}^p} \left\{ f(z) + \frac{\kappa}{2} \|z - x\|^2 \right\}$.

Proof By the definition of $p(x)$, we have $0 \in \partial f(p(x)) + \kappa(p(x) - x)$, meaning that $\kappa(x - p(x)) \in \partial f(p(x))$. By strong convexity of f,

$$\langle \kappa(y - p(y)) - \kappa(x - p(x)), p(y) - p(x) \rangle \geq \mu \|p(y) - p(x)\|^2.$$

Rearranging the terms yields the desired inequality. ■

As a consequence,

$$\begin{aligned}
&\left\| \frac{\kappa}{\kappa+\mu}(y_k - y_{k-1}) - (p(y_k) - p(y_{k-1})) \right\|^2 \\
&\qquad = \left\| \frac{\kappa}{\kappa+\mu}(y_k - y_{k-1}) \right\|^2 - 2\frac{\kappa}{\kappa+\mu}\langle y_k - y_{k-1}, p(y_k) - p(y_{k-1}) \rangle + \|p(y_k) - p(y_{k-1})\|^2 \\
&\qquad \leq \left\| \frac{\kappa}{\kappa+\mu}(y_k - y_{k-1}) \right\|^2 \\
&\qquad \leq \|y_k - y_{k-1}\|^2 .
\end{aligned}$$

Appendix C. Catalyst for MISO/Finito/SDCA

In this section, we present the application of Catalyst to MISO/Finito (Mairal, 2015; Defazio et al., 2014b), which may be seen as a variant of SDCA (Shalev-Shwartz and Zhang, 2016). The reason why these algorithms require a specific treatment is due to the fact that their linear convergence rates are given in a different form than (12); specifically, Theorem 4.1 of Lin et al. (2015a) tells us that MISO produces a sequence of iterates $(z_t)_{t\geq 0}$ for minimizing the auxiliary objective $h(z) = f(z) + \frac{\kappa}{2}\|z - y\|^2$ such that

$$\mathbb{E}[h(z_t)] - h^* \leq C_{\mathcal{M}}(1 - \tau_{\mathcal{M}})^{t+1}(h^* - d_0(z_0)),$$

where d_0 is a lower-bound of h defined as the sum of a simple quadratic function and the composite regularization ψ. More precisely, these algorithms produce a sequence $(d_t)_{t\geq 0}$ of such lower-bounds, and the iterate z_t is obtained by minimizing d_t in closed form. In particular, z_t is obtained from taking a proximal step at a well chosen point w_t, providing the following expression,

$$z_t = \text{prox}_{\psi/(\kappa+\mu)}(w_t).$$

Then, linear convergence is achieved for the duality gap

$$\mathbb{E}[h(z_t) - h^*] \leq \mathbb{E}[h(z_t) - d_t(z_t)] \leq C_{\mathcal{M}}(1 - \tau_{\mathcal{M}})^t(h^* - d_0(z_0)).$$

Indeed, the quantity $h(z_t) - d_t(z_t)$ is a natural upper-bound on $h(z_t) - h^*$, which is simple to compute, and which can be naturally used for checking the criterions (C1) and (C2). Consequently, the expected complexity of solving a given problem is slightly different compared to Lemma 11.

Lemma 23 (Accuracy vs. complexity) *Let us consider a strongly convex objective h and denote $(z_t)_{t\geq 0}$ the sequence of iterates generated by MISO/Finito/SDCA. Consider the* *complexity* $T(\varepsilon) = \inf\{t \geq 0, h(z_t) - d_t(z_t) \leq \varepsilon\}$*, where* $\varepsilon > 0$ *is the target accuracy and* h^* *is the minimum value of* h*. Then,*

$$\mathbb{E}[T(\varepsilon)] \leq \frac{1}{\tau_{\mathcal{M}}} \log \left(\frac{2C_{\mathcal{M}}(h^* - d_0(z_0))}{\tau_{\mathcal{M}}\varepsilon} \right) + 1,$$

where d_0 *is a lower bound of* f *built by the algorithm.*

For the convergence analysis, the outer-loop complexity does not change as long as the algorithm finds approximate proximal points satisfying criterions (C1) and (C2). It is then sufficient to control the inner loop complexity. As we can see, we now need to bound the dual gap $h^* - d_0(z_0)$ instead of the primal gap $h(z_0) - h^*$, leading to slightly different warm start strategies. Here, we show how to restart MISO/Finito.

Proposition 24 (Warm start for criterion (C1)) *Consider applying Catalyst with the same parameter choices as in Proposition 12 to MISO/Finito. At iteration* $k+1$ *of Algorithm 2, assume that we are given the previous iterate* x_k *in* $p^{\varepsilon_k}(y_{k-1})$*, the corresponding dual function* $d(x)$ *and its prox-center* w_k *satisfying* $x_k = \text{prox}_{\psi/(\kappa+\mu)}(w_k)$*. Then, initialize the sequence* $(z_t)_{t \geq 0}$ *for minimizing* $h_{k+1} = f + \frac{\kappa}{2}\| \cdot - y_k\|^2$ *with,*

$$z_0 = \text{prox}_{\psi/(\kappa+\mu)} \left(w_k + \frac{\kappa}{\kappa + \mu}(y_k - y_{k-1}) \right),$$

and initialize the dual function as

$$d_0(x) = d(x) + \frac{\kappa}{2}\|x - y_k\|^2 - \frac{\kappa}{2}\|x - y_{k-1}\|^2.$$

Then,

1. *when* f *is* μ*-strongly convex, we have* $h_{k+1}^* - d_0(z_0) \leq C\varepsilon_{k+1}$ *with the same constant as in (21) and (22), where* d_0 *is the dual function corresponding to* z_0*;*

2. *when* f *is convex with bounded level sets, there exists a constant* $B > 0$ *identical to the one of (23) such that*
$$h_{k+1}^* - d_0(z_0) \leq B.$$

Proof The proof is given in Lemma D.5 of Lin et al. (2015a), which gives

$$h_{k+1}^* - d_0(z_0) \leq \varepsilon_k + \frac{\kappa^2}{2(\kappa + \mu)}\|y_k - y_{k-1}\|^2.$$

This term is smaller than the quantity derived from (24), leading to the same upper bound. ■

Proposition 25 (Warm start for criterion (C2)) *Consider applying Catalyst with the same parameter choices as in Proposition 15 to MISO/Finito. At iteration* $k+1$ *of Algorithm 2, we assume that we are given the previous iterate* x_k *in* $g^{\delta_k}(y_{k-1})$ *and the* *corresponding dual function* $d(x)$*. Then, initialize the sequence* $(z_t)_{t\geq 0}$ *for minimizing* $h_{k+1} = f + \frac{\kappa}{2}\|\cdot - y_k\|^2$ *by*

$$z_0 = \text{prox}_{\psi/(\kappa+\mu)}\left(y_k - \frac{1}{\kappa+\mu}\nabla f_0(y_k)\right),$$

where $f = f_0 + \psi$ *and* f_0 *is the smooth part of* f*, and set the dual function* d_0 *by*

$$d_0(x) = f_0(y_k) + \langle \nabla f_0(y_k), x - y_k\rangle + \frac{\kappa+\mu}{2}\|x - y_k\|^2 + \psi(x).$$

Then,

$$h^*_{k+1} - d_0(z_0) \leq \frac{(L+\kappa)^2}{2(\mu+\kappa)}\|p(y_k) - y_k\|^2. \tag{31}$$

Proof Since $p(y_k)$ is the minimum of h_{k+1}, the optimality condition provides

$$-\nabla f_0(p(y_k)) - \kappa(p(y_k) - y_k) \in \partial\psi(p(y_k)).$$

Thus, by convexity,

$$\begin{aligned}
&\psi(p(y_k)) + \langle -\nabla f_0(p(y_k)) - \kappa(p(y_k) - y_k), z_0 - p(y_k)\rangle \leq \psi(z_0),\\
&f_0(p(y_k)) + \frac{\kappa}{2}\|p(y_k) - y_k\|^2 + \langle \nabla f_0(p(y_k)) + \kappa(p(y_k) - y_k), y_k - p(y_k)\rangle \leq f_0(y_k).
\end{aligned}$$

Summing up gives

$$h^*_{k+1} \leq f_0(y_k) + \psi(z_0) + \langle \nabla f_0(p(y_k)) + \kappa(p(y_k) - y_k), z_0 - y_k\rangle.$$

As a result,

$$\begin{aligned}
h^*_{k+1} - d_0(z_0) &\leq f_0(y_k) + \psi(z_0) + \langle \nabla f_0(p(y_k)) + \kappa(p(y_k) - y_k), z_0 - y_k\rangle - d_0(z_0)\\
&= \langle \nabla f_0(p(y_k)) + \kappa(p(y_k) - y_k) - \nabla f_0(y_k), z_0 - y_k\rangle - \frac{\kappa+\mu}{2}\|z_0 - y_k\|^2\\
&\leq \frac{1}{2(\kappa+\mu)}\|\nabla \underbrace{f_0(p(y_k)) - \nabla f_0(y_k)}_{\|\cdot\|\leq L\|p(y_k)-y_k\|} + \kappa(p(y_k) - y_k)\|^2\\
&\leq \frac{(L+\kappa)^2}{2(\mu+\kappa)}\|p(y_k) - y_k\|^2.
\end{aligned}$$

■

The bound obtained from (31) is similar to the one form Proposition 15, and differs only in the constant factor. Thus, the inner loop complexity in Section 4.2.2 still holds for MISO/Finito up to a constant factor. As a consequence, the global complexity of MISO/Finito applied to Catalyst is similar to one obtained by SVRG, yielding an acceleration for ill-conditioned problems.

References

A. Agarwal and L. Bottou. A lower bound for the optimization of finite sums. In *Proceedings of the International Conferences on Machine Learning (ICML)*, 2015.

Z. Allen-Zhu. Katyusha: The first direct acceleration of stochastic gradient methods. In *Proceedings of Symposium on Theory of Computing (STOC)*, 2017.

Y. Arjevani and O. Shamir. Dimension-free iteration complexity of finite sum optimization problems. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

A. Auslender. Numerical methods for nondifferentiable convex optimization. In *Nonlinear Analysis and Optimization*, volume 30, pages 102–126. Springer, 1987.

F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 4(1):1–106, 2012.

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.

D. P. Bertsekas. *Convex Optimization Algorithms.* Athena Scientific, 2015.

A. Chambolle and T. Pock. A remark on accelerated block coordinate descent for computing the proximity operators of a sum of convex functions. *SMAI Journal of Computational Mathematics*, 1:29–54, 2015.

R. Correa and C. Lemaréchal. Convergence of some algorithms for convex minimization. *Mathematical Programming*, 62(1):261–275, 1993.

A. Defazio. A simple practical accelerated method for finite sums. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

A. Defazio, F. Bach, and S. Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems (NIPS)*, 2014a.

A. Defazio, J. Domke, and T. S. Caetano. Finito: A faster, permutable incremental gradient method for big data problems. In *Proceedings of the International Conferences on Machine Learning (ICML)*, 2014b.

O. Devolder, F. Glineur, and Y. Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Mathematical Programming*, 146(1-2):37–75, 2014.

R. Frostig, R. Ge, S. M. Kakade, and A. Sidford. Un-regularizing: approximate proximal point and faster stochastic algorithms for empirical risk minimization. In *Proceedings of the International Conferences on Machine Learning (ICML)*, 2015.

M. Fuentes, J. Malick, and C. Lemaréchal. Descentwise inexact proximal algorithms for smooth optimization. *Computational Optimization and Applications*, 53(3):755–769, 2012.

P. Giselsson and M. Fält. Nonsmooth minimization using smooth envelope functions. *arXiv:1606.01327*, 2016.

O. Güler. On the convergence of the proximal point algorithm for convex minimization. *SIAM Journal on Control and Optimization*, 29(2):403–419, 1991.

O. Güler. New proximal point algorithms for convex minimization. *SIAM Journal on Optimization*, 2(4):649–664, 1992.

B. He and X. Yuan. An accelerated inexact proximal point algorithm for convex minimization. *Journal of Optimization Theory and Applications*, 154(2):536–548, 2012.

R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems (NIPS)*, 2013.

G. Lan and Y. Zhou. An optimal randomized incremental gradient method. *Mathematical Programming*, 2017.

C. Lemaréchal and C. Sagastizábal. Practical aspects of the Moreau–Yosida regularization: Theoretical preliminaries. *SIAM Journal on Optimization*, 7(2):367–385, 1997.

H. Lin, J. Mairal, and Z. Harchaoui. A universal catalyst for first-order optimization. In *Advances in Neural Information Processing Systems (NIPS)*, 2015a.

Q. Lin, Z. Lu, and L. Xiao. An accelerated randomized proximal coordinate gradient method and its application to regularized empirical risk minimization. *SIAM Journal on Optimization*, 25(4):2244–2273, 2015b.

J. Mairal. Incremental majorization-minimization optimization with application to large-scale machine learning. *SIAM Journal on Optimization*, 25(2):829–855, 2015.

J. Mairal. End-to-end kernel learning with supervised convolutional kernel networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

B. Martinet. Brève communication. Régularisation d'inéquations variationnelles par approximations successives. *Revue française d'informatique et de recherche opérationnelle, série rouge*, 4(3):154–158, 1970.

J.-J. Moreau. Fonctions convexes duales et points proximaux dans un espace hilbertien. *CR Acad. Sci. Paris Sér. A Math*, 255:2897–2899, 1962.

A. Nemirovskii and D. B. Yudin. *Problem complexity and method efficiency in optimization.* John Wiley & Sons, 1983.

Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course.* Springer, 2004.

Y. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

Y. Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.

C. Paquette, H. Lin, D. Drusvyatskiy, J. Mairal, and Z. Harchaoui. Catalyst acceleration for gradient-based non-convex optimization. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2018.

N. Parikh and S. P. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):123–231, 2014.

P. Richtárik and M. Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1-2):1–38, 2014.

R. T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14(5):877–898, 1976.

S. Salzo and S. Villa. Inexact and accelerated proximal point algorithms. *Journal of Convex Analysis*, 19(4):1167–1192, 2012.

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

M. Schmidt, N. Le Roux, and F. Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 160(1):83–112, 2017.

D. Scieur, A. d' Aspremont, and F. Bach. Regularized nonlinear acceleration. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

S. Shalev-Shwartz and T. Zhang. Proximal stochastic dual coordinate ascent. *preprint arXiv:1211.2717*, 2012.

S. Shalev-Shwartz and T. Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. *Mathematical Programming*, 155(1):105–145, 2016.

A. Sidi. *Vector Extrapolation Methods with Applications.* Society for Industrial and Applied Mathematics, 2017.

M. V. Solodov and B. F. Svaiter. A unified framework for some inexact proximal point algorithms. *Numerical Functional Analysis and Optimization*, 22(7-8):1013–1035, 2001.

A. Themelis, L. Stella, and P. Patrinos. Forward-backward envelope for the sum of two nonconvex functions: Further properties and nonmonotone line-search algorithms. *arXiv:1606.06256*, 2016.

B. E. Woodworth and N. Srebro. Tight complexity bounds for optimizing composite objectives. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

L. Xiao and T. Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

K. Yosida. Functional analysis. *Berlin-Heidelberg*, 1980.

Y. Zhang and L. Xiao. Stochastic primal-dual coordinate method for regularized empirical risk minimization. In *Proceedings of the International Conferences on Machine Learning (ICML)*, 2015.

H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.