Automatic Differentiation of Some First-Order Methods in Parametric Optimization

Sheheryar Mehmood
Deparment of Mathematics and Computer Science, Saarland University, Germany

Peter Ochs
Deparment of Mathematics and Computer Science, Saarland University, Germany

Abstract

We aim at computing the derivative of the solution to a parametric optimization problem with respect to the involved parameters. For a class broader than that of strongly convex functions, this can be achieved by automatic differentiation of iterative minimization algorithms. If the iterative algorithm converges pointwise, then we prove that the derivative sequence also converges pointwise to the derivative of the minimizer with respect to the parameters. Moreover, we provide convergence rates for both sequences. In particular, we prove that the accelerated convergence rate of the Heavy-ball method compared to Gradient Descent also accelerates the derivative computation. An experiment with L2-Regularized Logistic Regression validates the theoretical results.

1 Introduction

For a sufficiently smooth function $f : \mathbb{R}^N \times \mathbb{R}^P \to \mathbb{R}$, with $N, P \in \mathbb{N}$, we consider the parametric optimization problem:

$$\min_{\boldsymbol{x} \in \mathbb{R}^N} f(\boldsymbol{x}, \boldsymbol{u}) \,, \tag{$\mathcal{P}$}$$

for parameters $\boldsymbol{u} \in \mathbb{R}^P$. We assume that, for any $\boldsymbol{u}$, this problem has a unique solution, which defines the solution function $\boldsymbol{u} \mapsto \boldsymbol{x}^*(\boldsymbol{u})$, mapping a parameter $\boldsymbol{u}$ onto the solution $\boldsymbol{x}^*(\boldsymbol{u})$ of ($\mathcal{P}$). In this paper, we seek fast convergent iterative approximations of the derivative $D_{\boldsymbol{u}}\boldsymbol{x}^*$ of the solution function.

Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

Problems of the form ($\mathcal{P}$) are frequently encountered as lower (or inner) level problems in bilevel optimization (Dempe et al., 2015). The complementing upper (or outer) level problem often minimizes a loss function with respect to the parameter and the solution of the lower level problem. If both levels are sufficiently smooth, gradient based schemes can be used to solve the bilevel problem, which eventually requires to compute the derivative of the minimizer $D_{\boldsymbol{u}}\boldsymbol{x}^*$ (of the lower level) with respect to the parameter. This strategy is used in image denoising (Giryes et al., 2008; Domke, 2012; Kunisch and Pock, 2013), deblurring (Giryes et al., 2011) and segmentation (Ochs et al., 2016); data cleaning (Franceschi et al., 2017) and various other applications (Deledalle et al., 2014; Maclaurin et al., 2015; Pedregosa, 2016) for parameter learning, otherwise known as hyperparameter optimization in machine learning literature. Maclaurin et al. (2015) and Pedregosa (2016) were able to optimize thousands of hyperparamters using the so-called gradient based methods.
Another application is in grid search methods (Bergstra and Bengio, 2012), for which the derivative value allows for adaptable grid spacing.

In practice, at any $\boldsymbol{u}$, the solution $\boldsymbol{x}^*$ in ($\mathcal{P}$) is approximated by a sequence $(\boldsymbol{x}^{(k)})_{k\in\mathbb{N}}$ generated by an iterative optimization algorithm that converges to $\boldsymbol{x}^*$, for example, by Gradient Descent:

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} - \alpha \nabla_{\boldsymbol{x}} f(\boldsymbol{x}^{(k)}, \boldsymbol{u}) \,,$$

for $k \in \mathbb{N}$. Here we start with $\boldsymbol{x}^{(0)} \in \mathbb{R}^N$ and assume a constant step size $\alpha > 0$. The above update rule suggests that the iterates are dependent on $\boldsymbol{u}$ and under suitable conditions, the convergence of the sequence $(\boldsymbol{x}^{(k)}(\boldsymbol{u}))_{k\in\mathbb{N}}$ is guaranteed for a given $\boldsymbol{u}$. Since the algorithm relies only on the gradient information, it is therefore called a first order method. Another example of first order algorithms is the Heavy-ball method (Polyak, 1964), also known as gradient descent with

momentum or inertial gradient descent. This algorithm often accelerates the convergence of Gradient Descent and is known to be a so-called optimal algorithm for strongly convex functions (Nemirovsky and Yudin, 1983; Nesterov, 2004). As we are mainly interested in large scale problems (e.g. deep learning), the high dimensionality prohibits the usage of second order algorithms such as Newton's method (LeCun et al., 1998).

Since the minimizing sequence depends on $\boldsymbol{u}$, we consider the derivative sequence $(D_{\boldsymbol{u}}\boldsymbol{x}^{(k)}(\boldsymbol{u}))_{k\in\mathbb{N}}$ for approximating $D_{\boldsymbol{u}}\boldsymbol{x}^*$. In particular, our contribution is the following:

- For a sequence $(\boldsymbol{x}^{(k)}(\boldsymbol{u}))_{k\in\mathbb{N}}$ generated by Gradient Descent, we prove pointwise convergence and a convergence rate of the derivative sequence $(D_{\boldsymbol{u}}\boldsymbol{x}^{(k)}(\boldsymbol{u}))_{k\in\mathbb{N}}$ to $D_{\boldsymbol{u}}\boldsymbol{x}^*(\boldsymbol{u})$.
- For the Heavy-ball method, the optimal rate of convergence for $(\boldsymbol{x}^{(k)}(\boldsymbol{u}))_{k\in\mathbb{N}}$ is also proved for the derivative sequence.
- We study memory efficient variants, which turn out to yield an additional speed ups.

1.1 Related Work

One of the first works on differentiating iterative algorithms for parametric minimization is by Fischer (1991), who studied a parametric linear system of equations. For the discussed Jacobi method, the derivative sequence is shown to converge under the same conditions as the original sequence. Gilbert (1992) did the first comprehensive study of the problem. He considered a parametric iterative process that approaches a fixed point, and concluded convergence of the derivative sequence to the derivative of the fixed point. As an example, he showed that these results hold for Newton's method. He also suggested a technique to improve the convergence speed of the derivative sequence for forward mode case. This was further studied in detail by Christianson (1994) who proposed an efficient method for computing the derivative using the reverse mode automatic differentiation (AD).

Azmy (1997) performed numerical experiments by using Gilbert's efficient strategy for forward mode AD and found significant improvement in the accuracy of the derivative with same number of iterations as well as in computational power used in each iteration. Bartholomew–Biggs (1998) also used this strategy to speed up the convergence process. He performed numerical experiments and applied the results to various practical applications. Schlenkrich et al. (2008) integrated the reverse accumulation technique in ADOL-C (Griewank et al., 1996) for computing the derivatives of fixed-point iterations and used the package for analysis of a problem in Fluid Dynamics.

A question that remained unsolved in (Gilbert, 1992) was as to how to apply his results to a generalized fixed-point iterations, for instance, the quasi-Newton methods. Rosemblun (1993) performed successful experiments on the Broyden's method. Beck (1994) studied these iterations and provided theoretical results for convergence of the derivative sequences for such iterations. The conditions that he imposed on the iterations were similar to those by Gilbert. Griewank et al. (1993) provided the convergence guarantees for quasi-Newton methods like Broyden and DFP update rules. They pointed out that the rate and order of convergence of derivative sequences at best matches that of original sequences.

Christianson (1998) investigated this problem in a more general setting. He used reverse accumulation to compute the derivative of an implicit function when any eversion process is used to compute the value of the dependent variable (not just the fixed-point iterations). Griewank and Faure (2002) studied a similar problem in the context of a dynamic system where the state vector is given as an implicit function of the input vector and the derivative of the output vector which is provided as a function of input and state vector, is required. Bell and Burke (2008) studied the problem of computing gradient and Hessian of optimal value of a parametric objective function which is useful in saddle point problems or multilevel optimization.

We study AD for a more specialized setting of sequences that are derived from a minimization problem. We explore this additional information and prove that the derivative sequence generated by a so-called optimal algorithm, in the sense of (Nemirovsky and Yudin, 1983) and (Nesterov, 2004), has the same accelerated convergence rate as the original sequence.

2 Problem Setting

Given an open, non-empty and bounded set $\mathcal{U} \subset \mathbb{R}^P$, we consider ($\mathcal{P}$), where f is twice continuously differentiable on $\mathbb{R}^N \times \mathcal{U}$. We further assume that for all $\boldsymbol{u} \in \mathcal{U}$, the function $f(\cdot, \boldsymbol{u})$ is convex and a unique solution to ($\mathcal{P}$) exists. This allows us to define a map $\boldsymbol{x}^* : \mathcal{U} \to \mathbb{R}^N$ as $\boldsymbol{x}^*(\boldsymbol{u}) = \arg\min_{\boldsymbol{x}\in\mathbb{R}^N} f(\boldsymbol{x}, \boldsymbol{u})$ which is equivalently characterized by its optimality condition:

$$\nabla_{\boldsymbol{x}} f(\boldsymbol{x}^*(\boldsymbol{u}), \boldsymbol{u}) = 0\,.$$

For differentiation of the left hand side, we require the following assumption:

A1. *For all $\boldsymbol{u} \in \mathcal{U}$, the matrix $\nabla^2_{\boldsymbol{x}} f(\boldsymbol{x}^*(\boldsymbol{u}), \boldsymbol{u})$ is positive definite, and hence invertible.*

Example 1. *If $f(\cdot, \boldsymbol{u})$ is strongly convex for all $\boldsymbol{u} \in \mathcal{U}$, then Assumption A1 is satisfied. Therefore, our setting is more general.*

Remark 2. *The set $\mathcal{U}$ can be thought of as a neighborhood of a point for which we want to compute the derivative.*

From Assumption A1, we conclude that, for all $\boldsymbol{u} \in \mathcal{U}$, the function $f(\cdot, \boldsymbol{u})$ is $m(\boldsymbol{u})$-strongly convex on a closed $\varepsilon(\boldsymbol{u})$-neighborhood $B_{\varepsilon(\boldsymbol{u})}(\boldsymbol{x}^*(\boldsymbol{u}))$ of $\boldsymbol{x}^*(\boldsymbol{u})$ with $m(\boldsymbol{u}) > 0$. This implies that $\nabla^2_{\boldsymbol{x}} f(\boldsymbol{x}, \boldsymbol{u})$ is invertible on $\mathcal{Y} := \{(\boldsymbol{x}, \boldsymbol{u}) \in \mathbb{R}^N \times \mathcal{U} : \boldsymbol{x} \in B_{\varepsilon(\boldsymbol{u})}(\boldsymbol{x}^*(\boldsymbol{u}))\}$ and $\left\|\nabla^2_{\boldsymbol{x}} f(\boldsymbol{x}, \boldsymbol{u})^{-1}\right\| \leq 1/m(\boldsymbol{u})$ holds for all $(\boldsymbol{x}, \boldsymbol{u}) \in \mathcal{Y}$. Moreover, $f : \mathbb{R}^N \times \mathcal{U} \to \mathbb{R}$ is level bounded in $\boldsymbol{x}$ locally uniformly in $\boldsymbol{u}$ (Rockafellar and Wets, 1998, Definition 1.16). That is, for all $\boldsymbol{u} \in \mathcal{U}$, the function $f(\cdot, \boldsymbol{u})$ is lower-level bounded so that for some fixed $\boldsymbol{a} \in \mathbb{R}^N$, the set $\mathcal{X}(\boldsymbol{u}) := \operatorname{lev}_{\leq f(\boldsymbol{a}, \boldsymbol{u})} f(\cdot, \boldsymbol{u}) \supseteq B_{\varepsilon(\boldsymbol{u})}(\boldsymbol{x}^*(\boldsymbol{u}))$ is bounded and the set $\mathcal{Z} := \{(\boldsymbol{x}, \boldsymbol{u}) \in \mathbb{R}^N \times \mathcal{U} : \boldsymbol{x} \in \mathcal{X}(\boldsymbol{u})\}$ is bounded. Also, from extreme value theorem, for any $\boldsymbol{u} \in \mathcal{U}$, there exists an upper bound, $L(\boldsymbol{u}) > 0$, on the maximum eigenvalue of $\nabla^2_{\boldsymbol{x}} f(\cdot, \boldsymbol{u})$ on $\mathcal{X}(\boldsymbol{u})$. In other words, $\nabla_{\boldsymbol{x}} f(\cdot, \boldsymbol{u})$ is locally $L(\boldsymbol{u})$-Lipschitz continuous for every $\boldsymbol{u} \in \mathcal{U}$ and we have:

$$m(\boldsymbol{u}) I \preceq \nabla^2_{\boldsymbol{x}} f(\boldsymbol{x}, \boldsymbol{u}) \preceq L(\boldsymbol{u}) I\,, \tag{1}$$

for every $(\boldsymbol{x}, \boldsymbol{u}) \in \mathcal{Y}$. Similarly, $\|\nabla_{\boldsymbol{x}\boldsymbol{u}} f\|$ is bounded on $\mathcal{Z}$ by some $\kappa > 0$.

We state our second assumption for f which is motivated from the previous papers (Gilbert, 1992; Griewank et al., 1993).

A2. *The derivative map $D(\nabla_{\boldsymbol{x}} f)$ of the gradient of f with respect to $\boldsymbol{x}$ is Lipschitz continuous on $\mathcal{Z}$ with constant $C \geq 0$.*

We state following results for the solution map $\boldsymbol{x}^*$ and its derivative.

Lemma 3. *Under Assumptions A1 and A2, the function φ given by:*

$$\varphi(\boldsymbol{x}, \boldsymbol{u}) = -\nabla^2_{\boldsymbol{x}} f(\boldsymbol{x}, \boldsymbol{u})^{-1} \nabla_{\boldsymbol{x}\boldsymbol{u}} f(\boldsymbol{x}, \boldsymbol{u})\,, \tag{2}$$

is well-defined for all $(\boldsymbol{x}, \boldsymbol{u}) \in \mathcal{Y}$. It is bounded by $\kappa/m(\boldsymbol{u})$ and is $C(\kappa + m(\boldsymbol{u}))/m(\boldsymbol{u})^2$-Lipschitz Continuous on $\mathcal{Y}$. The function $\boldsymbol{x}^* : \mathcal{U} \to \mathcal{X}$ is continuously differentiable with $C(\kappa + m(\boldsymbol{u}))^2/m(\boldsymbol{u})^3$-Lipschitz Continuous derivative $D_{\boldsymbol{u}} \boldsymbol{x}^*(\boldsymbol{u}) = \varphi(\boldsymbol{x}^*(\boldsymbol{u}), \boldsymbol{u})$ on $\mathcal{U}$.

The above lemma is essentially a restatement of (Christianson, 1994, Theorem 2.2) with $\Phi(\cdot, \boldsymbol{u}) = \mathrm{Id} - \alpha \nabla_{\boldsymbol{x}} f(\cdot, \boldsymbol{u})$. An important consequence of this lemma is the following result which will be useful later.

Corollary 4. *Under the conditions of Lemma 3, for all $\boldsymbol{u} \in \mathcal{U}$, if a sequence $(\boldsymbol{x}^{(k)})_{k \in \mathbb{N}}$ lies in $\mathcal{X}(\boldsymbol{u})$ and converges to $\boldsymbol{x}^*(\boldsymbol{u})$ at a linear rate, then the sequence $(\varphi(\boldsymbol{x}^{(k)}, \boldsymbol{u}))_{k \in \mathbb{N}}$ converges to $D_{\boldsymbol{u}} \boldsymbol{x}^*(\boldsymbol{u})$ with the same rate.*

The proof is in Section A.1.

As discussed in the introduction, the objective of this paper is to estimate the derivative of the minimizer $D_{\boldsymbol{u}} \boldsymbol{x}^*$. In practice, however, direct computation of $D_{\boldsymbol{u}} \boldsymbol{x}^*$ is usually not possible and we have to content ourselves with approximations. A successful strategy is provided by automatic differentiation or AD, which we briefly recap in the following subsection.

2.1 Recap of AD

AD is an algorithmic way of differentiating a function given by a computer program at a given value of the input variable. It comprises two modes, namely forward and reverse mode, which we demonstrate in the context of our problem. We refer the reader to (Griewank and Walther, 2008) for a detailed account on AD and to (Gilbert, 1992; Beck, 1994) for AD applied to an iteration mapping.

Let $\boldsymbol{u} \in \mathcal{U}$, we approximate $\boldsymbol{x}^*(\boldsymbol{u})$ using the following parametrized, continuously differentiable iteration mapping $g : \mathbb{R}^N \times \mathbb{R}^P \to \mathbb{R}^N$:

$$\boldsymbol{x}^{(k+1)} := g(\boldsymbol{x}^{(k)}, \boldsymbol{u})\,, \tag{IM}$$

where $\boldsymbol{x}^{(0)} \in \mathbb{R}^N$ and $k \in \mathbb{N}$ denotes the iteration counter. We assume that the sequence $(\boldsymbol{x}^{(k)}(\boldsymbol{u}))_{k \in \mathbb{N}}$ generated by (IM) converges to $\boldsymbol{x}^*(\boldsymbol{u})$. We break the algorithm after a fixed number of $K \in \mathbb{N}$ iterations to obtain $\boldsymbol{x}^{(K)}(\boldsymbol{u})$, the suboptimal solution. Assuming $\boldsymbol{x}^{(0)}$ is independent of $\boldsymbol{u}$, the map $\boldsymbol{x}^{(K)} : \mathcal{U} \to \mathbb{R}^N$ is differentiable. We compute its derivative using the two modes of AD (forward and reverse mode) and use standard dotted and barred variable notation for these modes respectively. Also, following AD literature, if the original variables lie in a space (e.g. $\mathbb{R}^N$), then the dotted variables lie in the same space $\mathbb{R}^N$ whereas the barred variables lie in the dual space $\mathcal{L}(\mathbb{R}^N, \mathbb{R})$ (of linear mappings on $\mathbb{R}^N$).

The forward mode is straightforward. We start with $\dot{\boldsymbol{u}} := \boldsymbol{s}$ for some $\boldsymbol{s} \in \mathbb{R}^P$ and perform the following iterations for $k = 0, \ldots, K-1$:

$$\dot{\boldsymbol{x}}^{(k+1)} := D_{\boldsymbol{x}} g(\boldsymbol{x}^{(k)}, \boldsymbol{u}) \dot{\boldsymbol{x}}^{(k)} + D_{\boldsymbol{u}} g(\boldsymbol{x}^{(k)}, \boldsymbol{u}) \dot{\boldsymbol{u}}\,, \tag{IM-F}$$

to obtain the sequence $(\dot{\boldsymbol{x}}^{(k)})_{k\in[K]}$ where $[K] := \{0,\dots,K\}$ with $\dot{\boldsymbol{x}}^{(0)} = 0$, because $D_{\boldsymbol{u}}\boldsymbol{x}^{(0)} = 0$. In forward mode, the original iterates are computed alongside the derivative iterates without any overhead of memory.

The reverse mode, although a bit more complicated than the forward mode, proves to be relatively computationally efficient when P is significantly larger than N, for example, in deep learning where it is known as back-propagation (Rumelhart et al., 1986). In this mode, we start with $\bar{\boldsymbol{u}}_K^{(0)} = 0$ and $\bar{\boldsymbol{x}}^{(K)} := \boldsymbol{r}^T$ for some $\boldsymbol{r} \in \mathbb{R}^N$ and perform the following iterations for $n = 0,\dots,K-1$:

$$\begin{aligned}\bar{\boldsymbol{u}}_K^{(n+1)} &:= \bar{\boldsymbol{u}}_K^{(n)} + \bar{\boldsymbol{x}}^{(K-n)} D_{\boldsymbol{u}} g(\boldsymbol{x}^{(K-n-1)}, \boldsymbol{u}) \\ \bar{\boldsymbol{x}}^{(K-n-1)} &:= \bar{\boldsymbol{x}}^{(K-n)} D_{\boldsymbol{x}} g(\boldsymbol{x}^{(K-n-1)}, \boldsymbol{u})\,,\end{aligned} \tag{IM-R}$$

to obtain the sequence $(\bar{\boldsymbol{u}}_K^{(n)})_{n\in[K]}$. In reverse mode, we perform the original iterations and store the finite sequence $(\boldsymbol{x}^{(k)})_{k\in[K]}$ before going to derivative computation. Therefore memorywise, it is less efficient than forward mode. Notice that, we use a different index to denote the derivative sequence in reverse mode because we move in the opposite direction (backwards) to compute the derivative. The derivative information for forward and reverse mode is contained within the terms $\dot{\boldsymbol{x}}^{(K)} = D_{\boldsymbol{u}}\boldsymbol{x}^{(K)}\boldsymbol{s}$ and $\bar{\boldsymbol{u}}^{(K)} := \bar{\boldsymbol{u}}_K^{(K)} = \boldsymbol{r}^T D_{\boldsymbol{u}}\boldsymbol{x}^{(K)}$ respectively.

Gilbert (1992) showed that for all $\boldsymbol{u} \in \mathcal{U}$, if the sequence $(\boldsymbol{x}^{(k)}(\boldsymbol{u}))_{k\in\mathbb{N}}$ lies in $\mathcal{X}(\boldsymbol{u})$, the map Dg is Lipschitz on $\mathcal{Z}$ and the spectral radius $\rho(D_{\boldsymbol{x}} g(\boldsymbol{x}^*(\boldsymbol{u}), \boldsymbol{u})) < \tau$ for some $\tau \in [0,1)$, then $(\boldsymbol{x}^{(k)})_{k\in\mathbb{N}}$ converges like $\mathcal{O}(\tau^k)$ to $\boldsymbol{x}^*(\boldsymbol{u})$ and $(\dot{\boldsymbol{x}}^{(k)})_{k\in\mathbb{N}}$ converges like $\mathcal{O}(k\tau^k)$ to $\dot{\boldsymbol{x}}^*(\boldsymbol{u}) = D_{\boldsymbol{u}}\boldsymbol{x}^*(\boldsymbol{u})\boldsymbol{s}$.

Similar result holds for reverse mode because of the equivalence of two modes. Thus, the convergence of the derivative sequences is slightly slower as compared to that of original sequences. Gilbert (for forward mode) and later Christianson (1994) (for reverse mode) suggested ways to get past this problem by performing AD of $\boldsymbol{x}^{(K)}$ in an inexact manner. We briefly discuss this approach in the following subsection.

2.2 Inexact AD

Consider again, the update rules for forward (IM-F) and reverse (IM-R) mode AD of our iteration mapping given by (IM). The idea is to replace the intermediate iterates $\boldsymbol{x}^{(k)}$ (resp. $\boldsymbol{x}^{(K-n-1)}$) on the right side by the last iterate $\boldsymbol{x}^{(K)}$ for forward (resp. reverse) mode case for all $k \in [K]$ (resp. $n \in [K]$). Since this approach is different from exact AD, we alter our notation slightly. That is, we denote forward mode derivatives by hatted variables and reverse mode derivatives by tilde'ed variables for this approach. Therefore, the modified update rule for forward mode is given by:

$$\hat{\boldsymbol{x}}_K^{(k+1)} := D_{\boldsymbol{x}} g(\boldsymbol{x}^{(K)}, \boldsymbol{u})\hat{\boldsymbol{x}}_K^{(k)} + D_{\boldsymbol{u}} g(\boldsymbol{x}^{(K)}, \boldsymbol{u})\hat{\boldsymbol{u}} \tag{IM-FI}$$

and for reverse mode, by:

$$\begin{aligned}\tilde{\boldsymbol{u}}_K^{(n+1)} &:= \tilde{\boldsymbol{u}}_K^{(n)} + \tilde{\boldsymbol{x}}^{(K-n)} D_{\boldsymbol{u}} g(\boldsymbol{x}^{(K)}, \boldsymbol{u}) \\ \tilde{\boldsymbol{x}}^{(K-n-1)} &:= \tilde{\boldsymbol{x}}^{(K-n)} D_{\boldsymbol{x}} g(\boldsymbol{x}^{(K)}, \boldsymbol{u})\,,\end{aligned} \tag{IM-RI}$$

where we similarly set $\hat{\boldsymbol{u}} := \boldsymbol{s}$ and $\hat{\boldsymbol{x}}_K^{(0)} := 0$ for forward mode and $\tilde{\boldsymbol{x}}^{(K)} := \boldsymbol{r}^T$ and $\tilde{\boldsymbol{u}}_K^{(0)} := 0$ for reverse mode. These initializations are important and will be retained when we move to gradient descent and the Heavy-ball method in Sections 3 and 4. Note that, it is possible to perform (IM-FI) and (IM-RI) for $k, n \geq K$, even though we only performed a fixed K iterations of (IM). This is in contrast with (IM-F) and (IM-R).

Gilbert (1992) argued that under his assumptions (Subsection 2.1, last paragraph), the sequence $(\hat{\boldsymbol{x}}_K^{(k)})_{k\in\mathbb{N}}$ converges like $\mathcal{O}(\tau^k)$ to $\varphi(\boldsymbol{x}^{(K)}, \boldsymbol{u})\boldsymbol{s}$. The term $\varphi(\boldsymbol{x}^{(K)}, \boldsymbol{u}) \to D_{\boldsymbol{u}}\boldsymbol{x}^*$ like $\mathcal{O}(\tau^K)$ as $K \to \infty$ (Corollary 4). Similarly, Christianson (1994) showed that under the same assumptions, the sequence $(\tilde{\boldsymbol{u}}_K^{(n)})_{n\in\mathbb{N}}$ converges like $\mathcal{O}(\tau^k)$ to $\boldsymbol{r}^T\varphi(\boldsymbol{x}^{(K)}, \boldsymbol{u})$.

Remark 5. *The reverse accumulation strategy of Christianson (1994) is slightly different from* (IM-RI) *but he also used the last iterate only in his technique. With little effort it is possible to show that his results also extend to* (IM-RI).

The other advantage of using this approach is that we do not have any overhead of memory in the reverse mode so that K can be as large as desired for both modes. Also, we require less computational power for both modes because we only need to compute the derivative $Dg(\boldsymbol{x}^{(K)}, \boldsymbol{u})$ once. The above discussion shows that, as compared to exact AD, the inexact approach provides better convergence rate and computational performance and is also memory efficient when using reverse mode.

In Section 3, we apply these results on gradient descent in the setting of ($\mathcal{P}$). We show convergence of the sequences generated by exact and inexact AD of gradient descent for the objective functions that satisfy the assumptions defined at the start of this section. In Section 4, we show that the sequences computed by exact and inexact AD of the Heavy-ball method

also converge to the desired limits for these functions. We infer from our results that, whenever the Heavy-ball method accelerates the convergence of original sequence, the derivative sequences are also accelerated. Finally, in Section 5, we show that these results hold empirically as well.

3 AD of Gradient Descent

The update rule for gradient descent with constant step size $\alpha > 0$ applied to ($\mathcal{P}$) is given by:

$$\boldsymbol{x}^{(k+1)} \coloneqq \boldsymbol{x}^{(k)} - \alpha \nabla_{\boldsymbol{x}} f(\boldsymbol{x}^{(k)}, \boldsymbol{u}) \,, \tag{GD}$$

which we recognize as the special case of (IM) with $g(\boldsymbol{x}, \boldsymbol{u}) = \boldsymbol{x} - \alpha \nabla_{\boldsymbol{x}} f(\boldsymbol{x}, \boldsymbol{u})$. We define the map $R_{GD} : \mathbb{R}^N \times \mathbb{R} \to \mathbb{R}^{N \times N}$ as:

$$R_{GD}(\boldsymbol{x}, \alpha) = I - \alpha \nabla^2_{\boldsymbol{x}} f(\boldsymbol{x}, \boldsymbol{u}) \tag{3}$$

and use it to summarize some properties of (GD) in the following lemma. This map will be useful in proving the results for AD of (GD) as well.

Lemma 6. *For any $\boldsymbol{u} \in \mathcal{U}$, if the sequence $(\boldsymbol{x}^{(k)})_{k\in\mathbb{N}}$ is generated by (GD), then under Assumptions A1 and A2 and for $\alpha \leq 1/L(\boldsymbol{u})$, the sequence $(f(\boldsymbol{x}^{(k)}), \boldsymbol{u})_{k\in\mathbb{N}}$ is decreasing and converges to $f(\boldsymbol{x}^*(\boldsymbol{u}), \boldsymbol{u})$. Also, the sequence $(\boldsymbol{x}^{(k)})_{k\in\mathbb{N}}$ lies in $\mathcal{X}(\boldsymbol{u})$ and converges to $\boldsymbol{x}^*(\boldsymbol{u})$ and there exists $k_0 \geq 0$ and $q_{GD} \in [0, 1)$, such that, for all $k \geq k_0$:*

$$\left\| \boldsymbol{e}^{(k)} \right\| \leq q_{GD}^{k-k_0} \left\| \boldsymbol{e}^{(k_0)} \right\| \,,$$

where $\boldsymbol{e}^{(k)} \coloneqq \boldsymbol{x}^{(k)} - \boldsymbol{x}^*$.

The proof is in Section A.2.

Remark 7. *If $f(\cdot, \boldsymbol{u})$ is $m(\boldsymbol{u})$-strongly convex for all $\boldsymbol{u} \in \mathcal{U}$, then the choice of step size $\alpha = \alpha^*_{GD} \coloneqq 2/(L(\boldsymbol{u}) + m(\boldsymbol{u}))$ gives the best convergence rate of $q_{GD} = q^*_{GD} \coloneqq (L(\boldsymbol{u}) - m(\boldsymbol{u}))/(L(\boldsymbol{u}) + m(\boldsymbol{u}))$ for (GD) (Polyak, 1987).*

To perform AD on (GD), we similarly start with $\boldsymbol{x}^{(0)} \coloneqq \boldsymbol{a}$ and break the algorithm after K iterations. Therefore, the update rule for forward mode AD reads for $k = 0, \ldots, K-1$ as:

$$\dot{\boldsymbol{x}}^{(k+1)} \coloneqq R^{(k)}_{GD} \dot{\boldsymbol{x}}^{(k)} - \alpha \nabla_{\boldsymbol{xu}} f(\boldsymbol{x}^{(k)}, \boldsymbol{u}) \dot{\boldsymbol{u}} \tag{GD-F}$$

and for reverse mode as:

$$\begin{aligned} \bar{\boldsymbol{u}}^{(n+1)}_K &\coloneqq \bar{\boldsymbol{u}}^{(n)}_K - \alpha \bar{\boldsymbol{x}}^{(K-n)} \nabla_{\boldsymbol{xu}} f(\boldsymbol{x}^{(K-n-1)}, \boldsymbol{u}) \\ \bar{\boldsymbol{x}}^{(K-n-1)} &\coloneqq \bar{\boldsymbol{x}}^{(K-n)} R^{(K-n-1)}_{GD} \,, \end{aligned} \tag{GD-R}$$

where we set $R^{(k)}_{GD} \coloneqq R_{GD}(\boldsymbol{x}^{(k)}, \alpha)$. The convergence results for exact AD are shown in the following proposition.

Proposition 8. *For any $\boldsymbol{u} \in \mathcal{U}$, if the sequence $(\dot{\boldsymbol{x}}^{(k)})_{k\in\mathbb{N}}$ is generated by (GD-F), then under Assumptions A1 and A2 and for $\alpha \leq 1/L(\boldsymbol{u})$, it converges to $\dot{\boldsymbol{x}}^*(\boldsymbol{u}) = D_{\boldsymbol{u}} \boldsymbol{x}^*(\boldsymbol{u}) \boldsymbol{s}$ and there exists $k_0 \geq 0$, $C_1 > 0$ and $q_{GD} \in [0, 1)$, such that, for all $k \geq k_0$:*

$$\left\| \dot{\boldsymbol{e}}^{(k)} \right\| \leq q_{GD}^{k-k_0} \left\| \dot{\boldsymbol{e}}^{(k_0)} \right\| + C_1 (k - k_0) q_{GD}^{k-k_0} \left\| \boldsymbol{e}^{(k_0)} \right\| \,,$$

where $\dot{\boldsymbol{e}}^{(k)} \coloneqq \dot{\boldsymbol{x}}^{(k)} - \dot{\boldsymbol{x}}^*$.

The proof is in Section A.3.

Remark 9.

- *The convergence of the exact AD of (GD) is like $\mathcal{O}(k q_{GD}(\boldsymbol{u})^k)$.*
- *If $f(\cdot, \boldsymbol{u})$ is $m(\boldsymbol{u})$-strongly convex for all $\boldsymbol{u} \in \mathcal{U}$, then the optimal choice of step size gives the best convergence rate of $q_{GD} = q^*_{GD}$ for (GD-F) (Remark 7).*

Similarly, we apply inexact AD on gradient descent to obtain the update rule for forward mode as:

$$\hat{\boldsymbol{x}}^{(k+1)}_K \coloneqq R^{(K)}_{GD} \hat{\boldsymbol{x}}^{(k)}_K - \alpha \nabla_{\boldsymbol{xu}} f(\boldsymbol{x}^{(K)}, \boldsymbol{u}) \hat{\boldsymbol{u}} \,. \tag{GD-FI}$$

and for reverse mode as:

$$\begin{aligned} \tilde{\boldsymbol{u}}^{(n+1)}_K &\coloneqq \tilde{\boldsymbol{u}}^{(n)}_K - \alpha \tilde{\boldsymbol{x}}^{(K-n)} \nabla_{\boldsymbol{xu}} f(\boldsymbol{x}^{(K)}, \boldsymbol{u}) \\ \tilde{\boldsymbol{x}}^{(K-n-1)} &\coloneqq \tilde{\boldsymbol{x}}^{(K-n)} R^{(K)}_{GD} \,. \end{aligned} \tag{GD-RI}$$

In the following proposition, we state convergence results for inexact AD of (GD) and show that it achieves faster convergence as compared to exact AD. We drop the argument $(\boldsymbol{x}^{(K)}, \boldsymbol{u})$ for the maps $\nabla^2_{\boldsymbol{x}} f$ and $\nabla_{\boldsymbol{xu}} f$ for simplicity.

Proposition 10. *For any $\boldsymbol{u} \in \mathcal{U}$, if the sequences $(\hat{\boldsymbol{x}}^{(k)}_K)_{k\in\mathbb{N}}$ and $(\tilde{\boldsymbol{u}}^{(n)}_K)_{n\in\mathbb{N}}$ are generated by (GD-FI) and (GD-RI) respectively with sufficiently large $K \in \mathbb{N}$ such that $\boldsymbol{x}^{(K)} \in B_{\varepsilon(\boldsymbol{u})}(\boldsymbol{x}^*(\boldsymbol{u}))$, then under Assumptions A1 and A2 and for $\alpha \leq 1/L(\boldsymbol{u})$, these sequences*

converge to $\varphi(\boldsymbol{x}^{(k)},\boldsymbol{u})\boldsymbol{s}$ *and* $\boldsymbol{r}^T\varphi(\boldsymbol{x}^{(k)},\boldsymbol{u})$ *respectively and there exists* $q_{GD}\in[0,1)$ *such that for all* $k,n\in\mathbb{N}$, *we have:*

$$\left\|\hat{\boldsymbol{x}}_K^{(k)}-\varphi(\boldsymbol{x}^{(K)},\boldsymbol{u})\boldsymbol{s}\right\|\leq q_{GD}^k\frac{\kappa}{m(\boldsymbol{u})}\|\boldsymbol{s}\|$$

and

$$\left\|\tilde{\boldsymbol{u}}_K^{(n)}-\boldsymbol{r}^T\varphi(\boldsymbol{x}^{(K)},\boldsymbol{u})\right\|\leq q_{GD}^n\frac{\kappa}{m(\boldsymbol{u})}\|\boldsymbol{r}\|\,.$$

The proof is in Section A.4.

Remark 11.

- *The convergence of the inexact AD of* (GD) *is like* $\mathcal{O}(q_{GD}(\boldsymbol{u})^k)$ *which is better than that of exact AD (Remark 9).*
- *Again if* $f(\cdot,\boldsymbol{u})$ *is strongly convex for any* $\boldsymbol{u}\in\mathcal{U}$, *then the optimal choice of step size gives best convergence rate of* $q_{GD}=q_{GD}^*$ *for* (GD-FI) *and* (GD-RI).
- *The error bound in the above proposition shows that, with the estimate* $\boldsymbol{x}^{(K)}$ *of the minimizer, the sequences* $(\hat{\boldsymbol{x}}_K^{(k)})_{k\in\mathbb{N}}$ *and* $(\tilde{\boldsymbol{u}}_K^{(n)})_{n\in\mathbb{N}}$ *are quite similar and difference comes only due to different initializations of* $\hat{\boldsymbol{u}}$ *and* $\tilde{\boldsymbol{x}}^{(K)}$. *This effect is visible in Figure 1 (bottom row).*

When using backtracking line search (Boyd and Vandenberghe, 2004) for computing the step size α, its dependence on $\boldsymbol{x}^{(k)}$ for every $k\in\mathbb{N}$ makes (GD) non-differentiable. But this does not affect the differentiability of the minimizer $\boldsymbol{x}^*(\boldsymbol{u})$. Following consequence of Proposition 10 shows that the inexact approach is still usable in this case.

Corollary 12. *If* $\boldsymbol{x}^{(K)}\in B_{\varepsilon(\boldsymbol{u})}(\boldsymbol{x}^*(\boldsymbol{u}))$ *is generated by* (GD) *using backtracking line search, then the sequences* $(\hat{\boldsymbol{x}}_K^{(k)})_{k\in\mathbb{N}}$ *and* $(\tilde{\boldsymbol{u}}_K^{(n)})_{n\in\mathbb{N}}$ *computed with* α *set to the step size evaluated at the last iteration of* (GD) *converge to* $\varphi(\boldsymbol{x}^{(k)},\boldsymbol{u})\boldsymbol{s}$ *and* $\boldsymbol{r}^T\varphi(\boldsymbol{x}^{(k)},\boldsymbol{u})$ *respectively.*

The proof is in Section A.5.

4 AD of Heavy-ball Method

We now turn our attention to the Heavy-ball method applied to $(\mathcal{P})$ whose update rule for $k=0,\dots,K-1$, is given by:

$$\boldsymbol{x}^{(k+1)}=\boldsymbol{x}^{(k)}-\alpha\nabla_{\boldsymbol{x}}f(\boldsymbol{x}^{(k)},\boldsymbol{u})+\beta(\boldsymbol{x}^{(k)}-\boldsymbol{x}^{(k-1)})\,, \tag{HB}$$

with initialization $\boldsymbol{x}^{(-1)}:=\boldsymbol{x}^{(0)}$ and constant step size $\alpha>0$ and momentum parameter $\beta\in[0,1)$. We similarly define the map $R_{HB}:\mathbb{R}^N\times\mathbb{R}\times\mathbb{R}\to\mathbb{R}^{N\times N}$ as:

$$R_{HB}(\boldsymbol{x},\alpha,\beta)=(1+\beta)I-\alpha\nabla_{\boldsymbol{x}}^2f(\boldsymbol{x},\boldsymbol{u})\,. \tag{4}$$

and state the following lemma to outline some properties of (HB).

Lemma 13. *For any* $\boldsymbol{u}\in\mathcal{U}$, *if the sequence* $(\boldsymbol{x}^{(k)})_{k\in\mathbb{N}}$ *is generated by* (HB), *then under Assumptions A1 and A2 and for* $\beta\in[0,1)$ *and* $\alpha\leq 2(1+\beta)/L(\boldsymbol{u})$, *the sequence* $(f(\boldsymbol{x}^{(k)}),\boldsymbol{u})_{k\in\mathbb{N}}$ *is decreasing and converges to* $f(\boldsymbol{x}^*(\boldsymbol{u}),\boldsymbol{u})$. *Also, the sequence* $(\boldsymbol{x}^{(k)})_{k\in\mathbb{N}}$ *lies in* $\mathcal{X}(\boldsymbol{u})$ *and converges to* $\boldsymbol{x}^*(\boldsymbol{u})$. *In particular for all* $\gamma>0$, *there exists* c *such that:*

$$\left\|\boldsymbol{e}^{(k)}\right\|\leq c(q_{HB}+\gamma)^{k-k_0}\,,$$

for some $q_{HB}\in[0,1)$ *and* $k\geq k_0\geq 0$.

The proof is in Section A.6.

Remark 14. *If* $f(\cdot,\boldsymbol{u})$ *is* $m(\boldsymbol{u})$*-strongly convex for all* $\boldsymbol{u}\in\mathcal{U}$, *then the choices of* $\alpha=\alpha_{HB}^*:=4/(\sqrt{L(\boldsymbol{u})}+\sqrt{m(\boldsymbol{u})})^2$ *and* $\beta=\beta_{HB}^*:=(q_{HB}^*)^2$ *provides the best convergence rate of* $q_{HB}=q_{HB}^*:=(\sqrt{L(\boldsymbol{u})}-\sqrt{m(\boldsymbol{u})})/(\sqrt{L(\boldsymbol{u})}+\sqrt{m(\boldsymbol{u})})$ *for* (HB) *which is better than that of* (GD) *(Polyak, 1987).*

We assign $R_{HB}(\boldsymbol{x}^{(k)},\alpha,\beta)$ to $R_{HB}^{(k)}$ and start with $\dot{\boldsymbol{x}}^{(-1)}:=\dot{\boldsymbol{x}}^{(0)}$ to get the update rule for forward mode AD for $k=0,\dots,K-1$ as:

$$\dot{\boldsymbol{x}}^{(k+1)}:=R_{HB}^{(k)}\dot{\boldsymbol{x}}^{(k)}-\alpha\nabla_{\boldsymbol{xu}}f(\boldsymbol{x}^{(k)},\boldsymbol{u})\dot{\boldsymbol{u}}-\beta\dot{\boldsymbol{x}}^{(k-1)}\,, \tag{HB-F}$$

For reverse mode AD we have for $n=0,\dots,K-1$:

$$\begin{aligned}\bar{\boldsymbol{u}}_K^{(n+1)}&:=\bar{\boldsymbol{u}}_K^{(n)}-\alpha\bar{\boldsymbol{x}}^{(K-n)}\nabla_{\boldsymbol{xu}}f(\boldsymbol{x}^{(K-n-1)},\boldsymbol{u})\\ \bar{\boldsymbol{x}}^{(K-n-1)}&:=\bar{\boldsymbol{x}}^{(K-n)}R_{HB}^{(K-n-1)}-\beta\bar{\boldsymbol{x}}^{(K-n+1)}\,,\end{aligned} \tag{HB-R}$$

where we set $\bar{\boldsymbol{x}}^{(K+1)}:=0$. We state similar results for the convergence of AD of the Heavy-ball method.

Proposition 15. *For any* $\boldsymbol{u}\in\mathcal{U}$, *if the sequence* $(\dot{\boldsymbol{x}}^{(k)})_{k\in\mathbb{N}}$ *is generated by* (HB-F), *then under Assumptions A1 and A2 and for* $\beta\in[0,1)$ *and* $\alpha\leq 2(1+\beta)/L(\boldsymbol{u})$, *it converges to* $\dot{\boldsymbol{x}}^*=D_{\boldsymbol{u}}\boldsymbol{x}^*\boldsymbol{s}$. *In particular, for all* $\gamma>0$, *there exist* c_1,c_2 *such that:*

$$\begin{aligned}\left\|\dot{\boldsymbol{e}}^{(k)}\right\|\leq\ & c_1(q_{HB}+\gamma)^{k-k_0}+\\ & C_1c_2(k-k_0)(q_{HB}+\gamma)^{k-k_0}\,,\end{aligned}$$

for some $q_{HB} \in [0,1)$, $C_1 \geq 0$ *and* $k \geq k_0 \geq 0$.

The proof is in Section A.7.

Remark 16. *Again, If* $f(\cdot, \boldsymbol{u})$ *is* $m(\boldsymbol{u})$*-strongly convex for all* $\boldsymbol{u} \in \mathcal{U}$*, the optimal choices of* α *and* β *provides the best convergence rate of* $q_{HB} = q_{HB}^*$ *for* (HB-F) *which is better than that of* (GD-F) *(Remark 9).*

We give similar update rules for inexact AD of the Heavy-ball method as well. For forward mode, we set $\hat{\boldsymbol{x}}_K^{(-1)} := \hat{\boldsymbol{x}}_K^{(0)}$ and update the new iterates for $k = 0, \ldots, K-1$ as:

$$\hat{\boldsymbol{x}}_K^{(k+1)} := R_{HB}^{(K)} \hat{\boldsymbol{x}}_K^{(k)} - \alpha \nabla_{\boldsymbol{xu}} f(\boldsymbol{x}^{(K)}, \boldsymbol{u}) \hat{\boldsymbol{u}} - \beta \hat{\boldsymbol{x}}^{(k-1)} \tag{HB-FI}$$

and for reverse mode, we set $\tilde{\boldsymbol{x}}^{(K+1)} := 0$ and perform following iterations for $n = 0, \ldots, K-1$:

$$\begin{aligned} \tilde{\boldsymbol{u}}_K^{(n+1)} &:= \tilde{\boldsymbol{u}}_K^{(n)} - \alpha \tilde{\boldsymbol{x}}^{(K-n)} \nabla_{\boldsymbol{xu}} f(\boldsymbol{x}^{(K)}, \boldsymbol{u}) \\ \tilde{\boldsymbol{x}}^{(K-n-1)} &:= \tilde{\boldsymbol{x}}^{(K-n)} R_{HB}^{(K)} - \beta \tilde{\boldsymbol{x}}^{(K-n+1)} . \end{aligned} \tag{HB-RI}$$

We show that inexact AD of (HB) also converges to the desired limits.

Proposition 17. *For any* $\boldsymbol{u} \in \mathcal{U}$*, if the sequences* $(\hat{\boldsymbol{x}}_K^{(k)})_{k \in \mathbb{N}}$ *and* $(\tilde{\boldsymbol{u}}_K^{(n)})_{n \in \mathbb{N}}$ *are generated by* (HB-FI) *and* (HB-RI) *respectively with sufficiently large* $K \in \mathbb{N}$ *such that* $\boldsymbol{x}^{(K)} \in B_{\varepsilon(\boldsymbol{u})}(\boldsymbol{x}^*(\boldsymbol{u}))$*, then under Assumptions A1 and A2 and for* $\beta \in [0,1)$ *and* $\alpha \leq 2(1+\beta)/L(\boldsymbol{u})$*, these sequences converge to* $\varphi(\boldsymbol{x}^{(k)}, \boldsymbol{u})\boldsymbol{s}$ *and* $\boldsymbol{r}^T \varphi(\boldsymbol{x}^{(k)}, \boldsymbol{u})$ *respectively. In particular, for all* $\gamma > 0$*, there exist* c *such that:*

$$\left\| \hat{\boldsymbol{x}}_K^{(k)} - \varphi(\boldsymbol{x}^{(K)}, \boldsymbol{u})\boldsymbol{s} \right\| \leq c(q_{HB} + \gamma)^k \frac{\kappa}{m(\boldsymbol{u})} \|\boldsymbol{s}\|$$

and

$$\left\| \tilde{\boldsymbol{u}}_K^{(n)} - \boldsymbol{r}^T \varphi(\boldsymbol{x}^{(k)}, \boldsymbol{u}) \right\| \leq c(q_{HB} + \gamma)^n \frac{\kappa}{m(\boldsymbol{u})} \|\boldsymbol{r}\| ,$$

for some $q_{HB} \in [0,1)$ *and for every* $k, n \in \mathbb{N}$.

The proof is in Section A.8.

Remark 18. *Arguments made for inexact AD of gradient descent in Remark 11 similarly extend to* (HB-FI) *and* (HB-RI).

Corollary 19. *With the inexact scheme, it is possible to compute the estimate* $\boldsymbol{x}^{(K)}$ *using one algorithm and compute the derivative iterates using the other.*

The proof is in Section A.9.

5 Experiments

Given a feature matrix $A \in \mathbb{R}^{M \times N}$ with rows $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_M \in \mathbb{R}^N$ and target vector $\boldsymbol{b} \in \{0,1\}^M$, we consider a *regularized logistic regression problem* with objective function $f_N : \mathbb{R}^N \times \mathbb{R}^N_{++} \to \mathbb{R}$ defined as:

$$f_N(\boldsymbol{x}, \boldsymbol{u}) := \sum_{i=1}^{M} \log(1 + \exp(-b_i \langle \boldsymbol{a}_i, \boldsymbol{x} \rangle)) + \frac{1}{2} \sum_{j=1}^{N} u_i x_i^2 ,$$

for $\boldsymbol{u} = (u_1, \ldots, u_N) \in \mathbb{R}^N_{++}$. Moreover, we define the scalar variant $f_1(\boldsymbol{x}, u)$ that assumes all parameters to be identical $u_1 = \ldots = u_N$, which we identify with a single parameter $u \in \mathbb{R}_{++}$.

It can be shown that for a given u (resp. $\boldsymbol{u}$), $f_1(\cdot, u)$ (resp. $f_N(\cdot, \boldsymbol{u})$) is m-strongly convex with $m = u$ (resp. $m = \min_{j \leq N} u_j$) and has L-Lipschitz gradient with $L = \|A\|_2^2 + u$ (resp. $L = \|A\|_2^2 + \max_{j \leq N} u_j$). We can also show that the derivative maps $D(\nabla_{\boldsymbol{x}} f_1)$ and $D(\nabla_{\boldsymbol{x}} f_N)$ are Lipschitz continuous with constant $C \sim \mathcal{O}(\|A\|^3)$. This shows that the assumptions stated in Section 2 are satisfied for both functions.

We compute the derivative of the minimizers of f_1 and f_N with respect to their regularization parameters using the algorithms discussed in this paper. The goal is to validate our theoretical results empirically and, in particular, emphasize the practical advantage of using accelerated algorithms and the inexact approach: The original and the derivative sequence converge faster.

Since we do not have access to the analytical form for the minimizer, we find a good estimate by first applying gradient descent. Once we are very close to the minimizer, we apply Newton's method. Then (2) is used to compute a good estimate for the derivative of the minimizer. All the experiments are performed on Banknote Authentication Dataset Dua and Graff (2017) without any feature transformation and data augmentation.

For f_1, we set $u = 2$ and for f_N we choose $u_j \sim U(0,5)$ for all $j \in [N]$. We run the original algorithms (GD) and (HB) for $K = 6000$ iterations and evaluate the exact derivative algorithms (GD-F), (GD-R), (HB-F) and (HB-R) and the inexact derivative algorithms (GD-FI), (GD-RI), (HB-FI) and (HB-RI). Except for (GD-F) and (HB-F), which are run alongside their original counterparts, the derivative algorithms are executed after the termination of original algorithms for K iterations.

For original iterations, we generate finite sequences $(\boldsymbol{x}^{(k)})_{k \in [K]}$ by starting with $\boldsymbol{x}^{(0)} \in \mathbb{R}^N$. For forward mode derivative iterations, we start with $\dot{\boldsymbol{u}}$ and

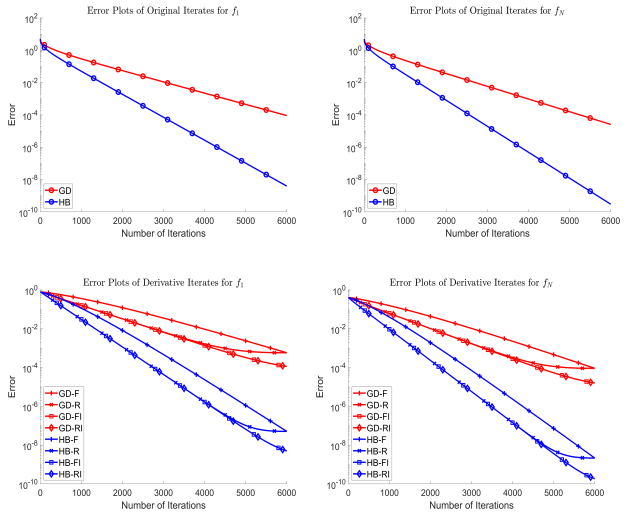

Figure 1: Errors for original (*upper row*) and derivative (*lower row*) sequences computed for f_1 (*left column*) and f_N (*right column*) using optimal algorithm parameters. The original and derivative sequences converge similarly for GD and HB. Moreover, the well-known advantage of acceleration of HB compared with GD is also reflected in the derivative sequences.

$\hat{\boldsymbol{u}}$ set to I_N and generate sequences $(\dot{\boldsymbol{x}}^{(k)})_{k\in[K]}$ and $(\hat{\boldsymbol{x}}_K^{(k)})_{k\in[K]}$ which lie in $\mathbb{R}^{N\times N}$. We might ask that these variables were introduced as vectors in previous sections but it can be seen that, computationally, this methodology makes sense and we expect the sequences to converge to the derivative of the minimizer. Similarly for reverse mode iterations, we start with $\bar{\boldsymbol{x}}^{(K)}$ and $\tilde{\boldsymbol{x}}^{(K)}$ set to I_N and generate finite sequences $(\bar{\boldsymbol{u}}_K^{(n)})_{n\in[K]}$ and $(\tilde{\boldsymbol{u}}_K^{(n)})_{n\in[K]}$.

The importance of optimal step size and momentum selection is explored by two different choices: α_{GD}^* and $\alpha_{GD}^*/3$ for gradient descent and $(\alpha_{HB}^*, \beta_{HB}^*)$ and $(\alpha_{HB}^*/3, \beta_{HB}^*/3)$ for the Heavy-ball method (see Remarks 7 and 14). Since suboptimal algorithm parameters slow down the convergence process for original iterations, we expect the same for derivative iterations.

In Figure 1, we plot the error norm against the number of iterations for optimal algorithm parameters. In Table 1, we also list the final accuracy of all the sequences after K iterations, including the results for suboptimal algorithm parameters.

The number of iterations required to get to the desired accuracy for the derivative sequences depends on the original sequence. For gradient descent, the original sequence takes time to get to the desired accuracy and so do the derivative sequences. For the Heavy-ball method, convergence is much faster for both type of sequences. Notice also the difference between the convergence of the derivative sequences. When performing the automatic differentiation on the sequences in a naive way, i.e., by using (GD-F), (GD-R), (HB-F) and (HB-R), the resulting sequences (Figure 1, lower row) reach their respective limit points relatively slower than their original counterparts (Figure 1, upper row). If we use the faster algorithms however, i.e. those given by (GD-FI), (GD-RI), (HB-FI) and (HB-RI), to compute the derivative sequences (Figure 1, lower row), we find that the number of iterations taken by the original and derivative sequences to get to the desired accuracy is almost the same.

Table 1: Accuracy of the algorithms after $K = 6000$ iterations computed for f_1 and f_N using optimal (o) and suboptimal (s) algorithm parameters.

Algo.	f_1 (o)	f_1 (s)	f_N (o)	f_N (s)
(GD)	9×10^{-5}	0.06	3×10^{-5}	0.04
(HB)	4×10^{-9}	0.01	3×10^{-10}	0.006
(GD-F)	6×10^{-4}	0.1	9×10^{-5}	0.04
(GD-R)	6×10^{-4}	0.1	9×10^{-5}	0.04
(HB-F)	5×10^{-8}	0.03	2×10^{-9}	0.01
(HB-R)	5×10^{-8}	0.03	2×10^{-9}	0.01
(GD-FI)	1×10^{-4}	0.06	2×10^{-5}	0.02
(GD-RI)	1×10^{-4}	0.06	2×10^{-5}	0.02
(HB-FI)	5×10^{-9}	0.01	2×10^{-10}	0.003
(HB-RI)	5×10^{-9}	0.01	2×10^{-10}	0.003

From the above experiments, we see that the behaviour of the original sequences is imitated by that of the derivative sequences. When we use the suboptimal algorithm parameters, we see that original sequences converge at a slower rate. This also leads to slower convergence for the derivative sequences. We also see that by replacing gradient descent with the Heavy-ball method, both the original and derivative sequences are provoked to converge with a better rate.

6 Conclusion

The derivative of the minimizer of a parametric objective function, under certain conditions, can be obtained by differentiating the estimate of the minimizer obtained through gradient descent or the Heavy-ball method. The Heavy-ball method accelerates the convergence of iterates for strongly convex functions. This acceleration is also reflected in the derivative sequences. The derivative computation process can be optimized in terms of time and memory by using the final iterate only, which also results in faster convergence.

References

Azmy, Y. (1997). Post-convergence automatic differentiation of iterative schemes. *Nuclear Science and Engineering*, 125(1):12–18.

Bartholomew–Biggs, M. C. (1998). Using forward accumulation for automatic differentiation of implicitly-defined functions. *Computational Optimization and Applications*, 9(1):65–84.

Beck, T. (1994). Automatic differentiation of iterative processes. *Journal of Computational and Applied Mathematics*, 50(1-3):109–118.

Bell, B. M. and Burke, J. V. (2008). Algorithmic differentiation of implicit functions and optimal values. In *Advances in Automatic Differentiation*, pages 67–77, Berlin, Heidelberg. Springer.

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305.

Bertsekas, D. P. (1999). *Nonlinear Programming*. Athena Scientific.

Boyd, S. and Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press.

Christianson, B. (1994). Reverse accumulation and attractive fixed points. *Optimization Methods and Software*, 3(4):311–326.

Christianson, B. (1998). Reverse accumulation and implicit functions. *Optimization Methods and Software*, 9(4):307–322.

Deledalle, C.-A., Vaiter, S., Fadili, J., and Peyré, G. (2014). Stein unbiased gradient estimator of the risk (sugar) for multiple parameter selection. *SIAM Journal on Imaging Sciences*, 7(4):2448–2487.

Dempe, S., Kalashnikov, V., Prez-Valds, G. A., and Kalashnykova, N. (2015). *Bilevel Programming Problems: Theory, Algorithms and Applications to Energy Networks*. Springer Publishing Company, Incorporated.

Domke, J. (2012). Generic methods for optimization based modeling. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics*, volume 22 of *Proceedings of Machine Learning Research*, pages 318–326, La Palma, Canary Islands. PMLR.

Dua, D. and Graff, C. (2017). UCI machine learning repository.

Fischer, H. (1991). Automatic differentiation of the vector that solves a parametric linear system. *Journal of Computational and Applied Mathematics*, 35(1-3):169–184.

Franceschi, L., Donini, M., Frasconi, P., and Pontil, M. (2017). Forward and reverse gradient-based hyperparameter optimization. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1165–1173, International Convention Centre, Sydney, Australia. PMLR.

Gelfand, I. (1941). Normierte ringe. *Rec. Math. [Mat. Sbornik] N.S.*, 9(51):3–24.

Gilbert, J. C. (1992). Automatic differentiation and iterative processes. *Optimization Methods and Software*, 1(1):13–21.

Giryes, R., Elad, M., and Eldar, Y. C. (2008). Automatic parameter setting for iterative shrinkage methods. In *Proceedings of the IEEE 25th Convention of Electrical and Electronics Engineers in Israel*, pages 820–824, Eilat, Israel. IEEE.

Giryes, R., Elad, M., and Eldar, Y. C. (2011). The projected gsure for automatic parameter tuning in iterative shrinkage methods. *Applied and Computational Harmonic Analysis*, 30(3):407–422.

Griewank, A., Bischof, C., Corliss, G., Carle, A., and Williamson, K. (1993). Derivative convergence for iterative equation solvers. *Optimization Methods and Software*, 2(3-4):321–355.

Griewank, A. and Faure, C. (2002). Reduced functions, gradients and Hessians from fixed-point iterations for state equations. *Numerical Algorithms*, 30(2):113–139.

Griewank, A., Juedes, D., and Utke, J. (1996). Algorithm 755: Adol-c: a package for the automatic differentiation of algorithms written in c/c++. *ACM Transactions on Mathematical Software (TOMS)*, 22(2):131–167.

Griewank, A. and Walther, A. (2008). *Evaluating Derivatives: Principles and Techniques of Algorithmic Differentiation*. SIAM, Philadelphia.

Kunisch, K. and Pock, T. (2013). A bilevel optimization approach for parameter learning in variational models. *SIAM Journal on Imaging Sciences*, 6(2):938–983.

LeCun, Y., Bottou, L., Orr, G. B., and Müller, K.-R. (1998). Efficient backprop. In *Neural Networks: Tricks of the Trade*, pages 9–50, London, UK, UK. Springer-Verlag.

Lu, T.-T. and Shiou, S.-H. (2002). Inverses of $2\times$ 2 block matrices. *Computers & Mathematics with Applications*, 43(1-2):119–129.

Maclaurin, D., Duvenaud, D., and Adams, R. P. (2015). Gradient-based hyperparameter optimization through reversible learning. In *Proceedings of the 32nd International Conference on Machine*

Learning, volume 37 of *Proceedings of Machine Learning Research*, pages 2113–2122, Lille, France. PMLR.

Nemirovsky, A. S. and Yudin, D. B. (1983). *Problem complexity and method efficiency in optimization.* Wiley, New York.

Nesterov, Y. (2004). *Introductory Lectures on Convex Optimization - A Basic Course*, volume 87 of *Applied Optimization*. Springer.

Ochs, P., Ranftl, R., Brox, T., and Pock, T. (2016). Techniques for gradient based bilevel optimization with nonsmooth lower level problems. *Journal of Mathematical Imaging and Vision*, 56(2):175–194.

Pedregosa, F. (2016). Hyperparameter optimization with approximate gradient. In Balcan, M. F. and Weinberger, K. Q., editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 737–746, New York, New York, USA. PMLR.

Polyak, B. T. (1964). Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17.

Polyak, B. T. (1987). *Introduction to Optimization*. Optimization Software.

Rockafellar, R. and Wets, R. J.-B. (1998). *Variational Analysis*. Springer Verlag, Heidelberg, Berlin, New York.

Rosemblun, M. L. (1993). *Automatic Differentiation: Overview and Application to Systems of Parameterized Nonlinear Equations*. PhD thesis, Rice University.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning internal representations by error propagation. In Rumelhart, D. E. and Mcclelland, J. L., editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*, pages 318–362. MIT Press, Cambridge, MA.

Schlenkrich, S., Walther, A., Gauger, N. R., and Heinrich, R. (2008). Differentiating fixed point iterations with ADOL-C: Gradient calculation for fluid dynamics. In *Modeling, Simulation and Optimization of Complex Processes*, pages 499–508. Springer.