Super-efficiency of automatic differentiation for functions defined as a minimum

Pierre Ablin [1] **Gabriel Peyré** [1] **Thomas Moreau** [2]

Abstract

In min-min optimization or max-min optimization, one has to compute the gradient of a function defined as a minimum. In most cases, the minimum has no closed-form, and an approximation is obtained via an iterative algorithm. There are two usual ways of estimating the gradient of the function: using either an *analytic* formula obtained by assuming exactness of the approximation, or *automatic* differentiation through the algorithm. In this paper, we study the asymptotic error made by these estimators as a function of the optimization error. We find that the error of the automatic estimator is close to the square of the error of the analytic estimator, reflecting a *super-efficiency* phenomenon. The convergence of the automatic estimator greatly depends on the convergence of the Jacobian of the algorithm. We analyze it for gradient descent and stochastic gradient descent and derive convergence rates for the estimators in these cases. Our analysis is backed by numerical experiments on toy problems and on Wasserstein barycenter computation. Finally, we discuss the computational complexity of these estimators and give practical guidelines to chose between them.

1. Introduction

In machine learning, many objective functions are expressed as the minimum of another function: functions ℓ defined as

$$\ell(x) = \min_{z\in\mathbb{R}^m} \mathcal{L}\left(z, x\right), \tag{1}$$

where $\mathcal{L} : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$. Such formulation arises for instance in dictionary learning, where x is a dictionary, z a sparse code, and $\mathcal{L}$ is the Lasso cost (Mairal et al., 2010). In this case, ℓ measures the ability of the dictionary x to encode the input data. Another example is the computation of the Wassertein barycenter of distributions in optimal transport (Agueh and Carlier, 2011): x represents the barycenter, ℓ is the sum of distances to x, and the distances themselves are defined by minimizing the transport cost. In the field of optimization, formulation (1) is also encountered as a smoothing technique, for instance in reweighted least-squares (Daubechies et al., 2010) where $\mathcal{L}$ is smooth but not ℓ. In game theory, such problems naturally appear in two-players maximin games(von Neumann, 1928), with applications for instance to generative adversarial nets (Goodfellow et al., 2014). In this setting, ℓ should be maximized.

A key point to optimize ℓ – either maximize or minimize – is usually to compute the gradient of ℓ, $g^*(x) \triangleq \nabla\ell(x)$. If the minimizer $z^*(x) = \arg\min_z \mathcal{L}(z, x)$ is available, the first order optimality conditions impose that $\nabla_1\mathcal{L}\left(z^*(x), x\right) = 0$ and the gradient is given by

$$g^*(x) = \nabla_2\mathcal{L}\left(z^*(x), x\right) \quad . \tag{2}$$

However, in most cases the minimizer $z^*(x)$ of the function is not available in closed-form. It is approximated via an iterative algorithm, which produces a sequence of iterates $z_t(x)$. There are then three ways to estimate $g^*(x)$:

The **analytic** estimator corresponds to plugging the approximation $z_t(x)$ in (2)

$$g^1_t(x) \triangleq \nabla_2\mathcal{L}\left(z_t(x), x\right) \quad . \tag{3}$$

The **automatic** estimator is $g^2_t(x) \triangleq \frac{d}{dx}\left[\mathcal{L}\left(z_t(x), x\right)\right]$, where the derivative is computed with respect to $z_t(x)$ as well. The chain rule gives

$$g^2_t(x) = \nabla_2\mathcal{L}\left(z_t(x), x\right) + \frac{\partial z_t}{\partial x}\nabla_1\mathcal{L}\left(z_t(x), x\right) \quad . \tag{4}$$

This expression can be computed efficiently using automatic differentiation (Baydin et al., 2018), in most cases at a cost similar to that of computing $z_t(x)$.

If $\nabla^2_{11}\mathcal{L}(z^*(x), x)$ is invertible, the implicit function theorem gives $\frac{\partial z^*(x)}{\partial x} = \mathcal{J}(z^*(x), x)$ where

[1] Department de Mathématique et Applications, ENS ULM, Paris, France [2] Université Paris-Saclay, Inria, CEA, Palaiseau, 91120, France. Correspondence to: Pierre Ablin <pierre.ablin@ens.fr>, Thomas Moreau <thomas.moreau@inria.fr>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020. Copyright 2020 by the author(s).

$\mathcal{J}(z,x) \triangleq -\nabla^2_{21}\mathcal{L}\left(z,x\right)\left[\nabla^2_{11}\mathcal{L}\left(z,x\right)\right]^{-1}$. The **implicit** estimator is

$$g^3_t(x) \triangleq \nabla_2\mathcal{L}(z_t(x),x) + \mathcal{J}(z_t(x),x)\nabla_1\mathcal{L}(z_t(x),x). \quad (5)$$

This estimator can be more costly to compute than the previous ones, as a $m \times m$ linear system has to be solved.

These estimates have been proposed and used by different communities. The analytic one corresponds to alternate optimization of $\mathcal{L}$, where one updates x while considering that z is fixed. It is used for instance in dictionary learning (Olshausen and Field, 1997; Mairal et al., 2010) or in optimal transport (Feydy et al., 2019). The second is common in the deep learning community as a way to differentiate through optimization problems (Gregor and Le Cun, 2010). Recently, it has been used as a way to accelerate convolutional dictionary learning (Tolooshams et al., 2018). It has also been used to differentiate through the Sinkhorn algorithm in optimal transport applications (Boursier and Perchet, 2019; Genevay et al., 2018). It integrates smoothly in a machine learning framework, with dedicated libraries (Abadi et al., 2016; Paszke et al., 2019). The third one is found in bi-level optimization, for instance for hyperparameter optimization (Bengio, 2000). It is also the cornerstone of the use of convex optimization as layers in neural networks (Agrawal et al., 2019).

Contribution In this article, we want to answer the following question: *which one of these estimators is the best?* The central result, presented in Sec.2, is the following convergence speed, when $\mathcal{L}$ is differentiable and under mild regularity hypothesis (Prop.2.1, 2.2 and 2.3)

$$\begin{aligned}
|g^1_t(x) - g^*(x)| &= O\left(|z_t(x) - z^*(x)|\right) \quad ,\\
|g^2_t(x) - g^*(x)| &= o\ \left(|z_t(x) - z^*(x)|\right) \quad ,\\
|g^3_t(x) - g^*(x)| &= O\left(|z_t(x) - z^*(x)|^2\right) \quad .
\end{aligned}$$

This is a super-efficiency phenomenon for the automatic estimator, illustrated in Fig.1 on a toy example. As our analysis reveals, the bound on g^2 depends on the convergence speed of the Jacobian of z_t, which itself depends on the optimization algorithm used to produce z_t. In Sec.3, we build on the work of Gilbert (1992) and give accurate bounds on the convergence of the Jacobian for gradient descent (Prop.3.2) and stochastic gradient descent (Prop.3.5 and 3.6) in the strongly convex case. We then study a simple case of non-strongly convex problem (Prop.3.9). To the best of our knowledge, these bounds are novel. This analysis allows us to refine the convergence rates of the gradient estimators. In Sec.4, we start by recalling and extending the consequence of using wrong gradients in an optimization algorithm (Prop.4.1 and 4.2). Then, since each gradient estimator comes at a different cost, we put the convergence bounds developed in the paper in perspective with a complexity analysis. This leads to practical and principled guidelines about which estimator should be used in which case. Finally, we provide numerical illustrations of the aforementioned results in Sec.5.

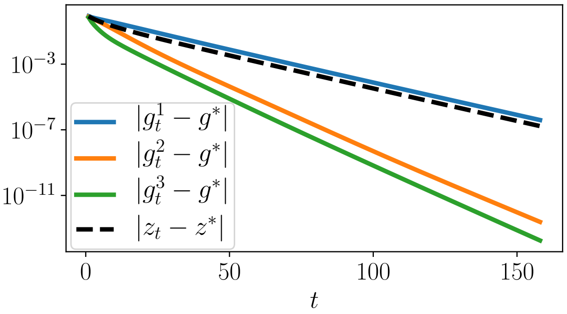

Figure 1. Convergence of the gradient estimators. $\mathcal{L}$ is strongly convex, x is a random point and $z_t(x)$ corresponds to t iterations of gradient descent. As t increases, z_t goes to z^* at a linear rate. g^1_t converges at the same rate while g^2_t and g^3_t are twice as fast.

Notation The ℓ^2 norm of $z \in \mathbb{R}^m$ is $|z| = \sqrt{\sum_{i=1}^m z_i^2}$. The operator norm of $M \in \mathbb{R}^{m\times n}$ is $\|M\| = \sup_{|z|=1}|Mz|$ and the Frobenius norm is $\|M\|_F = \sqrt{\sum_{i,j} M_{ij}^2}$. The vector of size n full of $1'$s is $\mathbb{1}_n$. The Euclidean scalar product is $\langle\cdot,\cdot\rangle$. The derivative of $\mathcal{L}$ with respect to its first variable (resp. second variable) is $\nabla_1\mathcal{L} \in \mathbb{R}^m$ (resp $\nabla_2\mathcal{L} \in \mathbb{R}^n$). The second derivative of $\mathcal{L}$ with respect to its variable $i = 1,2$ and variable $j = 1,2$ is denoted $\nabla^2_{ij}\mathcal{L}$. For sequences $x_t, y_t \in \mathbb{R}$ indexed by t, we denote $x_t = O(y_t)$ when there exists $C > 0$ such that $|x_t| \leq C|y_t|$ for t large enough. We denote $x_t = o(y_t)$ when $\frac{x_t}{y_t} \to 0$ when $t \to +\infty$. Finally, we use the Landau notation Θ to report computation and memory complexity magnitude. The proofs are only sketched in the article, full proofs are deferred to appendix.

2. Convergence speed of gradient estimates

We consider a compact set $K = K_z \times K_x \subset \mathbb{R}^m \times \mathbb{R}^n$. We make the following assumptions on $\mathcal{L}$.

H1: $\mathcal{L}$ is twice differentiable over K with second derivatives $\nabla^2_{21}\mathcal{L}$ and $\nabla^2_{11}\mathcal{L}$ respectively L_{21} and L_{11}-Lipschitz.

H2: For all $x \in K_x$, $z \to \mathcal{L}(z,x)$ has a unique minimizer $z^*(x) \in \text{int}(K_z)$. The mapping $z^*(x)$ is differentiable, with Jacobian $J^*(x) \in \mathbb{R}^{n\times m}$.

H1 implies that $\nabla_1\mathcal{L}$ and $\nabla_2\mathcal{L}$ are Lipschitz, with constants L_1 and L_2. The Jacobian of z_t at $x \in K_x$ is $J_t \triangleq \frac{\partial z_t(x)}{\partial x} \in \mathbb{R}^{n\times m}$. For the rest of the section, we consider a point $x \in K_x$, and we denote $g^* = g^*(x)$, $z^* = z(x)$ and $z_t = z_t(x)$.

2.1. Analytic estimator g^1

The analytic estimator (3) approximates g^* *as well as* z_t approximates z^* by definition of the L_2-smoothness.

Proposition 2.1 (Convergence of the analytic estimator)**.** *The analytic estimator verifies* $|g^1_t - g^*| \leq L_2|z_t - z^*|$.

2.2. Automatic estimator g^2

The automatic estimator (4) can be written as

$$g^2 = g^* + R(J_t)(z_t - z^*) + R_{21} + J_t R_{11} \qquad (6)$$

where

$$R_{21} \triangleq \nabla_2 \mathcal{L}(z_t, x) - \nabla_2 \mathcal{L}(z^*, x) - \nabla^2_{21}\mathcal{L}(z^*, x)(z_t - z^*)$$
$$R_{11} \triangleq \nabla_1 \mathcal{L}(z_t, x) - \nabla^2_{11}\mathcal{L}(z^*, x)(z_t - z^*) \ .$$

are Taylor's rests and

$$R(J) \triangleq J\nabla^2_{11}\mathcal{L}(z^*, x) + \nabla^2_{21}\mathcal{L}(z^*, x) \ . \qquad (7)$$

The implicit function theorem states that $R(J^*) = 0$. Importantly, in a non strongly-convex setting where $\nabla^2_{11}\mathcal{L}(z^*, x)$ is not invertible, it might happen that $R(J_t)$ goes to 0 even though J_t does not converge to J^*. **H1** implies a quadratic bound on the rests

$$|R_{21}| \leq \frac{L_{21}}{2}|z_t - z^*|^2 \text{ and } |R_{11}| \leq \frac{L_{11}}{2}|z_t - z^*|^2. \quad (8)$$

We assume that J_t is bounded $\|J_t\| \leq L_J$. This holds when J_t converges, which is the subject of Sec.3. The triangle inequality in Equation 6 gives:

Proposition 2.2 (Convergence of the automatic estimator)**.** *We define*

$$L \triangleq L_{21} + L_J L_{11} \qquad (9)$$

Then $|g^2_t - g^*| \leq \|R(J_t)\||z_t - z^*| + \frac{L}{2}|z_t - z^*|^2$.

This proposition shows that the rate of convergence of g^2 depends on the speed of convergence of $R(J_t)$. For instance, if $R(J_t)$ goes to 0, we have

$$g^2_t - g^* = o(|z_t - z^*|) \ .$$

Unfortunately, it might happen that, even though z_t goes to z^*, $R(J_t)$ does not go to 0 since differentiation is not a continuous operation. In Sec.3, we refine this convergence rate by analyzing the convergence speed of the Jacobian in different settings.

2.3. Implicit estimator g^3

The implicit estimator (5) is well defined provided that $\nabla^2_{11}\mathcal{L}$ is invertible. We obtain convergence bounds by making a Lipschitz assumption on $\mathcal{J}(z, x) = -\nabla^2_{21}\mathcal{L}(z, x)\left[\nabla^2_{11}\mathcal{L}(z, x)\right]^{-1}$.

Proposition 2.3. *[Convergence of the implicit estimator] Assume that* $\mathcal{J}$ *is* $L_{\mathcal{J}}$*-Lipschitz with respect to its first argument, and that* $\|\mathcal{J}_t\| \leq L_J$*. Then, for L as defined in (9),*

$$|g^3_t - g^*| \leq (\frac{L}{2} + L_{\mathcal{J}} L_1)|z_t - z^*|^2 \ . \qquad (10)$$

Sketch of proof. The proof is similar to that of Prop.2.2, using $\|R(\mathcal{J}(z_t, x))\| \leq L_1 L_{\mathcal{J}}|z_t - z^*|$. □

Therefore this estimator converges twice as fast as g^1, and at least as fast as g^2. Just like g^1 this estimator does not need to store the past iterates in memory, since it is a function of z_t and x. However, it is usually much more costly to compute.

2.4. Link with bi-level optimization

Bi-level optimization appears in a variety of machine-learning problems, such as hyperparameter optimization (Pedregosa, 2016) or supervised dictionary learning (Mairal et al., 2012). It considers problems of the form

$$\min_{x \in \mathbb{R}^n} \ell'(x) \triangleq \mathcal{L}'\left(z^*(x), x\right) \text{ s.t. } z^*(x) \in \arg\min_{z \in \mathbb{R}^m} \mathcal{L}(z, x), \qquad (11)$$

where $\mathcal{L}' : \mathbb{R}^m \times \mathbb{R}^n \to \mathbb{R}$ is another objective function. The setting of our paper is a special instance of bi-level optimization where $\mathcal{L}' = \mathcal{L}$. The gradient of ℓ' is

$$g'^* = \nabla\ell'(x) = \nabla_2\mathcal{L}'(z^*, x) + J^*\nabla_1\mathcal{L}'(z^*, x) \ .$$

When $z_t(x)$ is a sequence of approximate minimizers of $\mathcal{L}$, gradient estimates can be defined as

$$g'^1 = \nabla_2\mathcal{L}'(z_t(x), x),$$
$$g'^2 = \nabla_2\mathcal{L}'(z_t(x), x) + J_t\nabla_1\mathcal{L}'(z_t(x), x),$$
$$g'^3 = \nabla_2\mathcal{L}'(z_t(x), x) + \mathcal{J}(z_t(x), x)\nabla_1\mathcal{L}'(z_t(x), x).$$

Here, $\nabla_1\mathcal{L}'(z^*, x) \neq 0$, since z^* does not minimize $\mathcal{L}'$. Hence, g'^1 does not estimate g'^*. Moreover, in general,

$$\nabla^2_{21}\mathcal{L}'(z^*, x) + J^*\nabla^2_{11}\mathcal{L}'(z^*, x) \neq 0.$$

Therefore, there is no cancellation to allow super-efficiency of g^2 and g^3 and we only obtain linear rates

$$g'^2 - g^* = O(|z_t - z^*|), \ \ g'^3 - g^* = O(|z_t - z^*|) \ .$$

3. Convergence speed of the Jacobian

In order to get a better understanding of the convergence properties of the gradient estimators – in particular g^2 – we analyze it in different settings. A large portion of the analysis is devoted to the convergence of $R(J_t)$ to 0, since it does not directly follow from the convergence of z_t. In most cases, we show convergence of J_t to J^*, and use

$$\|R(J_t)\| \leq L_1\|J_t - J^*\| \qquad (12)$$

in the bound of Prop.2.2.

3.1. Contractive setting

When z_t are the iterates of a fixed point iteration with a contractive mapping, we recall the following result due to Gilbert (1992).

Proposition 3.1 (Convergence speed of the Jacobian)**.** *Assume that z_t is produced by a fixed point iteration*

$$z_{t+1} = \Phi\left(z_t, x\right),$$

where $\Phi : K_z \times K_x \to K_z$ is differentiable. We suppose that Φ is contractive: there exists $\kappa < 1$ such that for all $(z, z', x) \in K_z \times K_z \times K_x$, $|\Phi(z, x) - \Phi(z', x)| \leq \kappa |z - z'|$. Under mild regularity conditions on Φ:

- *z_t converges to a differentiable function z^* such that $z^* = \Phi(z^*, x)$, with Jacobian J^*.*
- *$|z_t - z^*| = O(\kappa^t)$ and $\|J_t - J^*\| = O(t\kappa^t)$*

3.2. Gradient descent in the strongly convex case

We consider the gradient descent iterations produced by the mapping $\Phi(z, x) = z - \rho \nabla_1 \mathcal{L}(z, x)$, with a step-size $\rho \leq 1/L_1$. We assume that $\mathcal{L}$ is μ-strongly convex with respect to z, i.e. $\nabla^2_{11} \mathcal{L} \succeq \mu \mathrm{Id}$ for all $z \in K_z, x \in K_x$. In this setting, Φ satisfies the hypothesis of Prop.3.1, and we obtain precise bounds.

Proposition 3.2. *[Convergence speed of the Jacobian of gradient descent in a strongly convex setting] Let z_t produced by the recursion $z_{t+1} = z_t - \rho \nabla_1 \mathcal{L}(z_t, x)$ with $\rho \leq 1/L_1$ and $\kappa \triangleq 1 - \rho\mu$. We have $|z_t - z^*| \leq \kappa^t |z_0 - z^*|$ and $\|J_t - J^*\| \leq t\kappa^{t-1} \rho L |z_0 - z^*|$ where L is defined in (9).*

Sketch of proof (C.1). We show that $\delta_t = \|J_t - J^*\|$ satisfies the recursive inequality $\delta_{t+1} \leq \kappa \delta_t + \rho L |z_0 - z^*| \kappa^t$. □

As a consequence, Prop. 2.1, 2.2, 2.3 together with Eq. (12) give in this case

$$\begin{aligned} |g^1 - g^*| &\leq L_2 |z_0 - z^*| \kappa^t, \\ |g^2 - g^*| &\leq (\rho L_1 t + \frac{\kappa}{2}) L |z_0 - z^*|^2 \kappa^{2t-1}, \\ |g^3 - g^*| &\leq (\frac{L}{2} + L_{\mathcal{J}} L_1) |z_0 - z^*|^2 \kappa^{2t} \ . \end{aligned} \tag{13}$$

We get the convergence speed $g^2 - g^* = O(t\kappa^{2t})$, which is almost twice better than the rate for g^1. Importantly, the order of magnitude in Prop.3.2 is tight, as it can be seen in the appendix Prop.C.1, where we exhibit a function reaching this upper-bound.

3.3. Stochastic gradient descent in z

We provide an analysis of the convergence of J_t in the stochastic gradient descent setting, assuming once again the μ-strong convexity of $\mathcal{L}$. We suppose that $\mathcal{L}$ is an expectation

$$\mathcal{L}(z, x) = \mathbb{E}_\xi [C(z, x, \xi)] \ ,$$

where ξ is drawn from a distribution d, and C is twice differentiable. Stochastic gradient descent (SGD) with steps ρ_t iterates

$$z_{t+1}(x) = z_t(x) - \rho_t \nabla_1 C\left(z_t(x), x, \xi_{t+1}\right) \text{ where } \xi_{t+1} \sim d \ .$$

In the stochastic setting, Prop.2.2 becomes

Proposition 3.3. *Define*

$$\delta_t = \mathbb{E}\left[\|J_t - J^*\|_F^2\right] \textit{ and } d_t = \mathbb{E}\left[|z_t - z^*|^2\right] \ . \tag{14}$$

We have $\mathbb{E}[|g^2 - g^*|] \leq L_1 \sqrt{\delta_t}\sqrt{d_t} + \frac{L}{2} d_t$.

Sketch of proof (C.4). We use Cauchy-Schwarz and the norm inequality $\|\cdot\| \leq \|\cdot\|_F$ to bound $\mathbb{E}\left[\|R(J_t)\| |z_t - z^*|\right]$. □

We begin by deriving a recursive inequality on δ_t, inspired by the analysis techniques of d_t.

Proposition 3.4. *[Bounding inequality for the Jacobian] We assume bounded Hessian noise, in the sense that $\mathbb{E}\left[\|\nabla^2_{11} C(z, x, \xi)\|_F^2\right] \leq \sigma^2_{11}$ and $\mathbb{E}\left[\|\nabla^2_{21} C(z, x, \xi)\|_F^2\right] \leq \sigma^2_{21}$. Let $r = \min(n, m)$, and $B^2 = \sigma^2_{21} + L_J^2 \sigma^2_{11}$. We have*

$$\delta_{t+1} \leq (1 - 2\rho_t\mu)\delta_t + 2\rho_t \sqrt{r} L \sqrt{d_t}\sqrt{\delta_t} + \rho_t^2 B^2. \tag{15}$$

Sketch of proof (C.5). A standard strong convexity argument gives the bound

$$\delta_{t+1} \leq (1 - 2\rho_t\mu)\delta_t + 2\rho_t\sqrt{r} L \mathbb{E}[\|J_t - J^*\|_F |z_t - z_0|] + \rho_t^2 B^2.$$

The middle term is then bounded using Cauchy-Schwarz inequality. □

Therefore, any convergence bound on d_t provides another convergence bound on δ_t by unrolling Eq. (15). We first analyze the fixed step-size case by using the simple "bounded gradients" hypothesis and bounds on d_t from Moulines and Bach (2011). In this setting, the iterates converge linearly until they reach a threshold caused by gradient variance.

Proposition 3.5. *[SGD with constant step-size] Assume that the gradients have bounded variance $\mathbb{E}_\xi[|\nabla_1 C(z, x, \xi)|^2] \leq \sigma^2$. Assume $\rho_t = \rho < 1/L_1$, and let $\kappa_2 = \sqrt{1 - 2\rho\mu}$ and $\beta = \sqrt{\frac{\sigma^2\rho}{2\mu}}$. In this setting*

$$\delta_t \leq \left(\kappa_2^t \left(\|J^*\|_F + t\alpha\right) + B_2\right)^2 \ ,$$

where $\alpha = \frac{\rho\sqrt{r}L}{\kappa_2} |z^* - z_0|$ and $B_2 = \frac{\rho\sqrt{r}L\beta}{\kappa_2(1-\kappa_2)} + \frac{\rho B}{(1-\kappa_2)}$.

Sketch of proof (C.6). Moulines and Bach (2011) give $d_t \leq \kappa_2^{2t}|z_0 - z^*|^2 + \beta^2$, which implies $\sqrt{d_t} \leq \kappa_2^t|z_0 - z^*| + \beta$. A bit of work on Eq. (15) then gives

$$\sqrt{\delta_{t+1}} \leq \kappa_2\sqrt{\delta_t} + \rho\kappa_2^t + (1-\kappa_2)B_2$$

Unrolling the recursion gives the proposed bound. □

This bound showcases that the familiar "two-stages" behavior also stands for δ_t: a transient linear decay at rate $(1-2\rho\mu)^t$ in the first iterations, and then convergence to a stationary noise level B_2^2. This bound is not tight enough to provide a good estimate of the noise level in δ_t. Let B_3^2 the limit of the sequence defined by the recursion (15). We find

$$B_3^2 = (1-2\rho\mu)B_3^2 + 2\rho\sqrt{r}L\beta B_3 + \rho^2 B^2,$$

which gives by expanding β

$$B_3 = \sqrt{\rho}\frac{\sqrt{r}L\sigma}{(2\mu)^{\frac{3}{2}}}\left(1 + \sqrt{1 + \frac{4\mu^2 B^2}{rL^2\sigma^2}}\right) .$$

This noise level scales as $\sqrt{\rho}$, which is observed in practice. In this scenario, Prop. 2.1, 2.2 and 2.3 show that $g^1 - g^*$ reaches a noise level proportional to $\sqrt{\rho}$, just like d_t, while both $g^2 - g^*$ and $g^3 - g^*$ reach a noise level proportional to ρ: g^2 and g^3 can estimate g^* to an accuracy that can never be reached by g^1. We now turn to the decreasing step-size case.

Proposition 3.6. *[SGD with decreasing step-size] Assume that $\rho_t = \rho_0 t^{-\alpha}$ with $\alpha \in (0,1)$. Assume a bound on d_t of the form $d_t \leq d^2 t^{-\alpha}$. Then*

$$\delta_t \leq 4\frac{\rho_0 B^2\mu + rL^2d^2}{\mu^2}t^{-\alpha} + o(t^{-\alpha}) .$$

Sketch of proof (C.7). We use (15) to obtain a recursion

$$\delta_{t+1} \leq \left(1 - \mu\rho_0 t^{-\alpha}\right)\delta_t + (B^2\rho_0^2 + \frac{rL^2d^2\rho_0}{\mu})t^{-2\alpha} ,$$

which is then unrolled. □

When $\rho_t \propto t^{-\alpha}$, we have $d_t = O(t^{-\alpha})$ so the assumption $d_t \leq d^2t^{-\alpha}$ is verified for some d. One could use the precise bounds of (Moulines and Bach, 2011) to obtain non-asymptotic bounds on δ_t as well. The rate $t^{-\alpha}$ in this bound is tight, as we exhibit a function with this rate in the appendix Prop.C.2.

Overall, we recover bounds for δ_t with the same behavior than the bounds for d_t. Pluging them in Prop.3.3 gives the asymptotic behaviors for the gradient estimators.

Proposition 3.7 (Convergence speed of the gradient estimators for SGD with decreasing step)**.** *Assume that $\rho_t = Ct^{-\alpha}$ with $\alpha \in (0,1)$. Then*

$$\mathbb{E}_\xi[|g^1 - g^*|] = O(\sqrt{t^{-\alpha}}),\ \mathbb{E}_\xi[|g^2 - g^*|] = O(t^{-\alpha})$$
$$\mathbb{E}_\xi[|g^3 - g^*|] = O(t^{-\alpha})$$

The super-efficiency of g^2 and g^3 is once again illustrated, as they converge at the same speed as d_t.

3.4. Beyond strong convexity

All the previous results rely critically on the strong convexity of $\mathcal{L}$. A function f with minimizer z^* is p-Łojasiewicz (Attouch and Bolte, 2009) when $\mu(f(z) - f(z^*))^{p-1} \leq \|\nabla f(z)\|^p$ for some $\mu > 0$. Any strongly convex function is 2-Łojasiewicz: the set of p-Łojasiewicz functions for $p \geq 2$ offers a framework beyond strong-convexity that still provides convergence rates on the iterates. The general study of gradient descent on this class of function is out of scope for this paper. We analyze a simple class of p-Łojasiewicz functions, the least mean p-th problem, where

$$\mathcal{L}(z,x) \triangleq \frac{1}{p}\sum_{i=1}^{n}(x_i - [Dz]_i)^p \tag{16}$$

for p an even integer and D is overcomplete ($\text{rank}(D) = n$). In this simple case, $\mathcal{L}(\cdot, x)$ is minimized by cancelling $x - Dz$, and $g^* = (x - Dz^*)^{p-1} = 0$.

In the case of least squares ($p = 2$) we can perfectly describe the behavior of gradient descent, which converges linearly.

Proposition 3.8. *Let z_t the iterates of gradient descent with step $\rho \leq \frac{1}{L_1}$ in (16) with $p = 2$, and $z^* \in \arg\min \mathcal{L}(z,x)$. It holds*

$$g^1 = D(z_t - z^*),\ g^2 = D(z_{2t} - z^*) \text{ and } g^3 = 0.$$

Proof. The iterates verify $z_t - z^* = (I_n - D^\top D)^t(z_0 - z^*)$, and we find $J_t\nabla_1\mathcal{L}(z_t, x) = (I_n - (I_n - D^\top D)^t)(x - Dz_t)$. The result follows. □

The automatic estimator therefore goes exactly twice as fast as the analytic one to g^*, while the implicit estimator is exact. Then, we analyze the case where $p \geq 4$ in a more restrictive setting.

Proposition 3.9. *For $p \geq 4$, we assume $DD^\top = I_n$. Let $\alpha \triangleq \frac{p-1}{p-2}$. We have*

$$|g_t^1| = O(t^{-\alpha}),\quad |g_t^2| = O(t^{-2\alpha}),\quad g_t^3 = 0 .$$

Sketch of proof (C.8). We first show that the residuals $r_t = x - Dz_t$ verify $r_t = \left(\frac{1}{\rho(p-2)t}\right)^{\frac{1}{p-2}}(1 + O(\frac{\log(t)}{t}))$, which gives the result for g^1. We find $g_t^2 = M_t r_t^{p-1}$ where $M_t = I_n - J_t D^\top$ verifies $M_{t+1} = M_t(I_n - (p-1)\rho \mathrm{diag}(r_t^{p-2}))$. Using the development of r_t and unrolling the recursion concludes the proof. □

For this problem, g^2 is of the order of magnitude of g^1 squared and as $p \to +\infty$, we see that the rate of convergence of g^1 goes to t^{-1}, while the one of g^2 goes to t^{-2}.

4. Consequence on optimization

In this section, we study the impact of using the previous inexact estimators for first order optimization. These estimators nicely fit in the framework of inexact oracles introduced by Devolder et al. (2014).

4.1. Inexact oracle

We assume that ℓ is μ_2-strongly convex and L_2-smooth with minimizer x^*. A (δ, μ, L)-inexact oracle is a couple (ℓ_δ, g_δ) such that $\ell_\delta : \mathbb{R}^m \to \mathbb{R}$ is the inexact value function, $g_\delta : \mathbb{R}^m \to \mathbb{R}^m$ is the inexact gradient and for all x, y

$$\frac{\mu}{2}|x-y|^2 \leq \ell(x) - \ell_\delta(y) - \langle g_\delta(y)|x-y\rangle \leq \frac{L}{2}|x-y|^2 + \delta \quad . \tag{17}$$

Devolder et al. (2013) show that if the gradient approximation g^i verifies $|g^*(x) - g^i(x)| \leq \Delta_i$ for all x, then (ℓ, g^i) is a $(\delta_i, \frac{\mu_2}{2}, 2L_2)$-inexact oracle, with

$$\delta_i = \Delta_i^2(\frac{1}{\mu_2} + \frac{1}{2L_2}) \quad . \tag{18}$$

We consider the optimization of ℓ with inexact gradient descent: starting from $x_0 \in \mathbb{R}^n$, it iterates

$$x_{q+1} = x_q - \eta g_t^i(x_q) \quad , \tag{19}$$

with $\eta = \frac{1}{2L_2}$, a fixed t and $i = 1, 2$ or 3.

Proposition 4.1. *[Devolder et al. 2013, Theorem 4] The iterates x_q with estimate g^i verify*

$$\ell(x_q) - \ell(x^*) \leq 2L_2(1 - \frac{\mu_2}{4L_2})^q|x_0 - x^*|^2 + \delta_i$$

with δ_i defined in (18).

As q goes to infinity, the error made on $\ell(x^*)$ tends towards $\delta_i = O(|g_t^i - g^*|^2)$. Thus, a more precise gradient estimate achieves lower optimization error. This illustrates the importance of using gradients estimates with an error Δ_i as small as possible.

Table 1. Computational and memory costs for a quadratic loss $\mathcal{L}$. Here $c \geq 1$ is to the relative cost of automatic differentiation.

Estimator	Computational cost	Memory cost
g_t^1	$\Theta(mnt)$	$\Theta(m)$
g_t^2	$\Theta(cmnt)$	$\Theta(mt)$
g_t^3	$\Theta(mnt + m^3 + m^2n))$	$\Theta(m(m+n))$

We now consider stochastic optimization for our problem, with loss ℓ defined as

$$\ell(x) = \mathbb{E}_\upsilon[h(x, \upsilon)] \text{ with } h(x, \upsilon) = \min_z H(z, x, \upsilon) \quad .$$

Stochastic gradient descent with constant step-size $\eta \leq \frac{1}{2L_2}$ and inexact gradients iterates

$$x_{q+1} = x_q - \eta g_t^i(x_q, \upsilon_{q+1}) \quad ,$$

where $g_t^i(x_q, \upsilon_{q+1})$ is computed by an approximate minimization of $z \to H(z, x_q, \upsilon_{q+1})$.

Proposition 4.2. *We assume that H is μ_2-strongly convex, L_2-smooth and verifies*

$$\mathbb{E}[|\nabla h(x, \upsilon) - \nabla \ell(x)|^2] \leq \sigma^2.$$

The iterates x_q of SGD with approximate gradient g^i and step-size η verify

$$\mathbb{E}|x_q - x^*|^2 \leq (1 - \frac{\eta\mu_2}{2})^q|x_0 - x^*| + \frac{2\eta}{\mu_2}\sigma^2 + \frac{4}{\mu_2}\delta_i$$

with $\delta_i = \Delta_i^2(\frac{1}{\mu_2} + \frac{1}{2L_2} + 2\eta)$.

The proof is deferred to Appendix D. In this case, it is pointless to achieve an estimation error on the gradient Δ_i smaller than some fraction of the gradient variance σ^2.

As a final note, these results extend without difficulty to the problem of maximizing ℓ, by considering gradient ascent or stochastic gradient ascent.

4.2. Time and memory complexity

In the following, we put our results in perspective with a computational and memory complexity analysis, allowing us to provide practical guidelines for optimization of ℓ.

Computational complexity of the estimators The cost of computing the estimators depends on the cost function $\mathcal{L}$. We give a complexity analysis in the least squares case (16) which is summarized in Table 1. In this case, computing the gradient $\nabla_1\mathcal{L}$ takes $\Theta(mn)$ operations, therefore the cost of computing z_t with gradient descent is $\Theta(mnt)$. Computing g_t^1 comes at the same price. The estimator g^2 requires a reverse pass on the computational graph, which costs a factor $c \geq 1$ of the forward computational cost: the final cost is $\Theta(cmn)$. Griewank and Walther (2008) showed that typically $c \in [2, 3]$. A popular technique to alleviate the cost of back-propagation is to only backpropagate through the last k iterations of the algorithm, which it equivalent to assuming $J_{t-k} = 0$ (Shaban et al., 2019). This technique trades gradient accuracy for time and memory complexity.

While it is beyond the scope of this paper, one could analyse this method using the tools developed in Sec.3, where the recursive inequalities in the proof of Prop.3.2 or in Prop.3.4 should be unrolled k times. Finally, computing g^3 requires a costly $\Theta(m^2 n)$ Hessian computation, and a $\Theta(m^3)$ linear system inversion. The final cost is $\Theta(mnt + m^3 + m^2 n)$. The linear scaling of g_t^1 and g_t^2 is highlighted in Fig.A.3.

Linear convergence: a case for the analytic estimator In the time it takes to compute g_t^2, one can at the same cost compute g_{ct}^1. If z_t converges linearly at rate κ^t, Prop.2.1 shows that $g_{ct}^1 - g^* = O(\kappa^{ct})$, while Prop.2.2 gives, at best, $g_t^2 - g^* = O(\kappa^{2t})$: g_{ct}^1 is a better estimator of g^* than g_t^2, provided that $c \geq 2$. In the quadratic case, we even have $g_t^2 = g_{2t}^1$. Further, computing g^2 might requires additional memory: g_{ct}^1 should be preferred over g_t^2 in this setting. However, our analysis is only asymptotic, and other effects might come into play to tip the balance in favor of g^2.

As it appears clearly in Table 1, choosing g^1 over g^3 depends on t: when $mnt \gg m^3 + m^2 n$, the additional cost of computing g^3 is negligible, and it should be preferred since it is more accurate. This is however a rare situation in a large scale setting.

Sublinear convergence We have provided two settings where z_t converges sub-linearly. In the stochastic gradient descent case with a fixed step-size, one can benefit from using g^2 over g^1, since it allows to reach an accuracy that can never be reached by g^1. With a decreasing step-size, reaching $|g_t^1 - g^*| \leq \varepsilon$ requires $\Theta(\varepsilon^{-2/\alpha})$ iterations, while reaching $|g_t^2 - g^*| \leq \varepsilon$ only takes $\Theta(\varepsilon^{-1/\alpha})$ iterations. For ε small enough, we have $c\varepsilon^{-1/\alpha} < \varepsilon^{-2/\alpha}$: it is always beneficial to use g^2 if memory capacity allows it.

The story is similar for the simple non-strongly convex problem studied in Sec.3.4: because of the slow convergence of the algorithms, g_t^2 is *much* closer to g^* than g_{ct}^1. Although our analysis was carried in the simple least mean p-th problem, we conjecture it could be extended to the more general setting of p-Łojasiewicz functions (Attouch and Bolte, 2009).

Memory footprint of the estimators Another critical aspect when choosing between these estimators is their memory footprint. While g^1 does not require extra memory compared to computing the loss, computing g^2 usually requires to store in memory all intermediate variables, which might be a burden as it requires memory $\Theta(mt)$. Checkpointing can reduce the memory cost for g^2 to $\Theta(m\sqrt{t})$ but with a computational complexity twice as large (Hascoet and Araya-Polo, 2006). Also, some optimization algorithms are *invertible*, such as SGD with momentum (Maclaurin et al., 2015). Using these algorithms removes the need to store each intermediate variable, since they can be recomputed on the fly. Practical implementations of this method still requires a small memory per iteration in order to enforce numerical stability of the inversion. Note that truncated backpropagation (Shaban et al., 2019) can also be used to reduce the memory cost of using automatic differentiation, with approximation. Finally, the main memory cost for estimator g^3 is storing the second order Hessian of size $\Theta(m(m+n))$.

5. Experiments

All experiments are performed in Python using `pytorch` (Paszke et al., 2019). The code to reproduce the figures is available online.[1]

5.1. Considered losses

In our experiments, we considered several losses with different properties. For each experiments, the details on the size of the problems are reported in Sec.A.1.

Regression For a design matrix $D \in \mathbb{R}^{n\times m}$ and a regularization parameter $\lambda > 0$, we define

$$\mathcal{L}_1(z,x) = \frac{1}{2}|x - Dz|^2 + \frac{\lambda}{2}|z|^2 \ ,$$
$$\mathcal{L}_2(z,x) = \sum_{i=1}^{n} \log\left(1 + e^{-x_i[Dz]_i}\right) + \frac{\lambda}{2}|z|^2 \ ,$$
$$\mathcal{L}_3(z,x) = \frac{1}{p}|x - Dz|^p; \quad p = 4 \ .$$

$\mathcal{L}_1$ corresponds to Ridge Regression, which is quadratic and strongly convex when $\lambda > 0$. $\mathcal{L}_2$ is the Regularized Logistic Regression. It is strongly convex when $\lambda > 0$. $\mathcal{L}_3$ is studied in Sec.3.4, and defined with $DD^\top = I_n$.

Regularized Wasserstein Distance The Wasserstein distance defines a distance between probability distributions. In Cuturi (2013), a regularization of the problem is proposed, which allows to compute it efficiently using the Sinkhorn algorithm, enabling many large scale applications. As we will see, the formulation of the problem fits nicely in our framework. The set of histograms is $\Delta_+^m = \{a \in \mathbb{R}_+^m | \ \sum_{i=1}^m a_i = 1\}$. Consider two histograms $a \in \Delta_+^{m_a}$ and $b \in \Delta_+^{m_b}$. The set of couplings is $U(a,b) = \{P \in \mathbb{R}_+^{m_a\times m_b} | \ P\mathbb{1}_{m_b} = a, \ P^\top\mathbb{1}_{m_a} = b\}$. The histogram a (resp. b) is associated with set of m_a (resp. m_b) points in dimension k, $(X_1, \dots, X_{m_a}) \in \mathbb{R}^k$ (resp $(Y_1, \dots, Y_{m_b})$). The cost matrix is $C \in \mathbb{R}^{m_a\times m_b}$ such that $C_{ij} = |X_i - Y_j|^2$. For $\epsilon > 0$, the entropic regularized Wasserstein distance is $W_\epsilon^2(a,b) = \min_{P\in U(a,b)}\langle C, P\rangle + \epsilon\langle\log(P), P\rangle$. The dual formulation of the previous variational formulation is (Peyré and Cuturi, 2019, Prop. 4.4.):

$$W_\epsilon^2(a,b) = \min_{z_a,z_b} \underbrace{\langle a, z_a\rangle + \langle b, z_b\rangle + \epsilon\langle e^{-z_a/\epsilon}, e^{-C/\epsilon}e^{-z_b/\epsilon}\rangle}_{\mathcal{L}_4((z_a,z_b),a)} \tag{20}$$

[1] See `https://github.com/tomMoral/diffopt`.

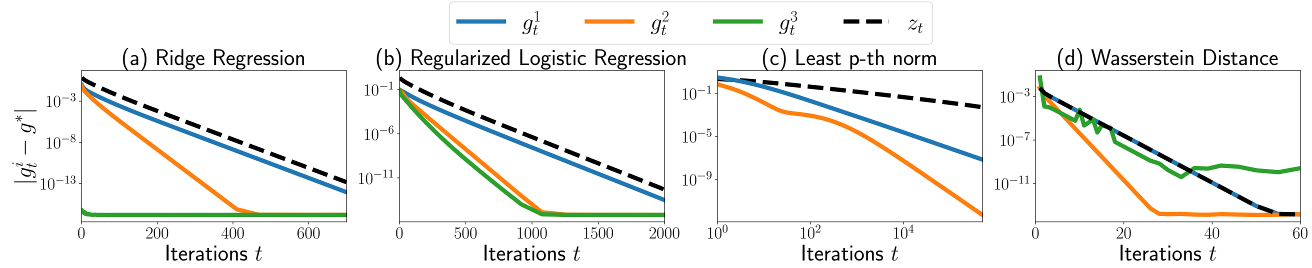

Figure 2. Evolution of $|g_t^i - g^*|$ with the number of iteration t for (a) the Ridge Regression $\mathcal{L}_1$, (b) the Regularized Logistic Regression $\mathcal{L}_2$, (c) the Least Mean p-th Norm $\mathcal{L}_3$ in *log-scale* and (d) the Wasserstein Distance $\mathcal{L}_4$. In all cases, we can see the asymptotic super-efficiency of the g_2 estimator compared to g_1. The g_3 estimator is better in most cases but it is unstable in (d). It is not included in (c) as it is equal to 0 due to the particular shape of $\mathcal{L}_3$.

This loss is strongly convex up to a constant shift on z_a, z_b. The Sinkhorn algorithm performs alternate minimization of $\mathcal{L}_4$:

$$z_a \leftarrow \epsilon(\log(e^{-C/\epsilon}e^{-z_b/\epsilon}) - \log(a)),$$
$$z_b \leftarrow \epsilon(\log(e^{-C^\top/\epsilon}e^{-z_a/\epsilon}) - \log(b)) \ \ .$$

This optimization technique is not covered by the results in Sec.3, but we will see that the same conclusions hold in practice.

5.2. Examples of super-efficiency

To illustrate the tightness of our bounds, we evaluate numerically the convergence of the different estimators g^1, g^2 and g^3 toward g^* for the losses introduced above. For all problems, g^* is computed by estimating $z^*(x)$ with gradient descent for a very large number of iterations and then using (2).

Gradient Descent Fig.2 reports the evolution of $|g_t^i - g^*|$ with t for the losses $\{\mathcal{L}_j\}_{j=1}^4$, where z_t is obtained by gradient descent for $\mathcal{L}_1, \mathcal{L}_2$ and $\mathcal{L}_3$, and by Sinkhorn iterations for $\mathcal{L}_4$. For the strongly convex losses (a),(b), $|g_t^1 - g^*|$ converges linearly with the same rate as $|z_t - z^*|$ while $|g_t^2 - g^*|$ converges about twice as fast. This confirms the theoretical findings of Prop.3.1 and (13). The estimator g^3 also converges with the predicted rates in (a),(b), however, it fails in (d) as the Hessian of $\mathcal{L}_4$ is ill-conditionned, leading to numerical instabilities. For the non-strongly convex loss $\mathcal{L}_3$, Fig.2.(c) shows that the rates given in Prop.3.9 are correct as g_1 converges with a rate $t^{-\frac{3}{2}}$ while g_t^2 converges as t^{-3}. Here, we did not include g_t^3 as it is exact due to the particular form of $\mathcal{L}_3$.

Stochastic Gradient Descent In Fig.3, we investigate the evolution of expected performances of g^i for the SGD, in order to validate the results of Sec.3.3. We consider $\mathcal{L}_2$. The left part (a) displays the asymptotic expected performance $\mathbb{E}[|g_t^i - g^*|]$ in the fixed step case, as a function of the step ρ, computed by running the SGD with sufficiently many iterations to reach a plateau. As predicted in Sec.3.3, the noise level scales as $\sqrt{\rho}$ for g^1 while it scales like ρ for g^2 and g^3. The right part (b) displays the evolution of $\mathbb{E}[|g_t^i - g^*|]$ as a function of t, where the step-size is decreasing $\rho_t \propto t^{-\alpha}$. Here again, the asymptotic rates predicted by Prop.3.7 is showcased: $g^1 - g^*$ is $O(\sqrt{t^{-\alpha}})$ while $g^2 - g^*$ and $g^3 - g^*$ are $O(t^{-\alpha})$.

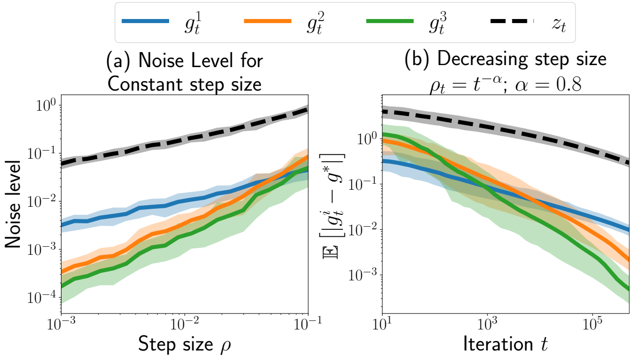

Figure 3. Expected performances of g^i for the SGD; (a) noise level as $t \to +\infty$ for a constant step-size ρ; (b) Expected error as a function of the number of iteration for decreasing step-size $\rho_t = Ct^{-0.8}$. The solid line displays the mean values and the shaded area the first and last decile.

5.3. Example on a full training problem

We are now interested in the minimization of ℓ with respect to x, possibly under constraints. We consider the problem of computing Wasserstein barycenters using mirror descent, as proposed in (Cuturi and Doucet, 2014). For a set of histograms $b_1, \dots, b_N \in \Delta_+^m$ and a cost matrix $C \in \mathbb{R}^{n\times m}$, the entropic regularized Wasserstein barycenter of the b_i's is

$$x \in \arg\min_{x\in\Delta_+^n} \ell(x) = \sum_{i=1}^N W_\epsilon^2(x, b_i) \ \ ,$$

where W_ϵ^2 is defined in (20), and we have:

$$\ell(x) = \min_{z_x^1,\dots,z_x^N, z_b^1,\dots,z_b^N} \sum_{i=1}^N \mathcal{L}_4\left((z_x^i, z_b^i), x\right) \ \ . \tag{21}$$

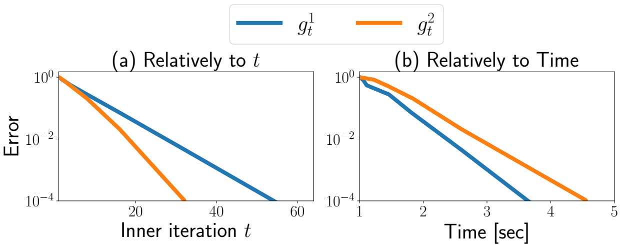

Figure 4. Final optimization error δ_i relatively to (a) the number of inner iterations used to estimation g^i; (b) the time taken to reach this optimization error level.

The dual variables z_x^i, z_b^i are obtained with t iterations of the Sinkhorn algorithm. In this simple setting, $\nabla_2 \mathcal{L}_4\left((z_x, z_b), x\right) = z_x$. The cost function is then optimized by mirror descent, with approximate gradient g^i: $x_{q+1} = P_\Delta(\exp(-\eta g^i)x_q)$, where $P_\Delta(x) = x / \sum_{i=1}^n x_i$ is the projection on Δ_+^n. Fig.4 displays the scale of the error $\delta_i = \ell(x_q) - \ell(x^*)$. We excluded g^3 here as the computation were unstable – as seen in Fig.2.(c) – and too expensive. The error decreases much faster with number of inner iteration t by using g_t^2 compared to g_t^1. However, when looking at the time taken to reach the asymptotic error, we can see that g^1 is a better estimator in this case. This illustrates the fact that while g^2 is almost twice as good at approximating g^* as g^1, it is at least twice as expensive, as discussed in Sec.4.2.

6. Conclusion

In this work, we have described the asymptotic behavior of three classical gradient estimators for a special instance of bilevel estimation. We have highlighted a super-efficiency phenomenon of automatic differentiation. However, our complexity analysis shows that it is faster to use the standard analytic estimator when the optimization algorithm converges linearly, and that the super-efficiency can be leveraged for algorithms with sub-linear convergence. This conclusion should be taken with caution, as our analysis is only asymptotic. This suggests a new line of research interested in the non-asymptotic behavior of these estimators. Extending our results to a broader class of non-strongly convex functions would be another interesting direction, as we observe empirically that for logistic-regression, $g_2 - g^* \simeq (g_1 - g^*)^2$. However, as the convexity alone does not ensure the convergence of the iterates, it raises interesting question for the gradient estimation. Finally, it would also be interesting to extend our analysis to non-smooth problems, for instance when z_t is obtained with the proximal gradient descent algorithm as in the case of ISTA for dictionary learning.

Acknowledgment

The authors thank Louis Thiry for fruitful discussions. P.A. and G.P. acknowledge support from the European Research Council (ERC project NORIA).

References

Martın Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pages 265–283, 2016.

Akshay Agrawal, Brandon Amos, Shane Barratt, and Stephen Boyd. Differentiable Convex Optimization Layers. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 9558–9570, 2019.

Martial Agueh and Guillaume Carlier. Barycenters in the wasserstein space. *SIAM Journal on Mathematical Analysis*, 43(2):904–924, 2011.

Hedy Attouch and Jérôme Bolte. On the convergence of the proximal algorithm for nonsmooth functions involving analytic features. *Mathematical Programming*, 116(1-2): 5–16, 2009.

Atılım Gunes Baydin, Barak A Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic Differentiation in Machine Learning: A Survey. *Journal of Machine Learning Research (JMLR)*, 18:1–43, 2018.

Yoshua Bengio. Gradient-based optimization of hyperparameters. *Neural computation*, 12(8):1889–1900, 2000.

Etienne Boursier and Vianney Perchet. Utility/Privacy Trade-off through the lens of Optimal Transport. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, October 2019.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2292–2300, 2013.

Marco Cuturi and Arnaud Doucet. Fast computation of wasserstein barycenters. *Journal of Machine Learning Research (JMLR)*, 2014.

Ingrid Daubechies, Ronald DeVore, Massimo Fornasier, and C Sinan Güntürk. Iteratively reweighted least squares minimization for sparse recovery. *Communications on Pure and Applied Mathematics: A Journal Issued by the*

Courant Institute of Mathematical Sciences, 63(1):1–38, 2010.

Olivier Devolder, François Glineur, and Yurii Nesterov. First-order methods with inexact oracle: The strongly convex case. Discussion paper, CORE, 2013.

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

Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouvé, and Gabriel Peyré. Interpolating between Optimal Transport and MMD using Sinkhorn Divergences. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 2681–2690, 2019.

Aude Genevay, Gabriel Peyre, and Marco Cuturi. Learning generative models with sinkhorn divergences. In *International Conference on Artificial Intelligence and Statistics (AISTAT)*, pages 1608–1617, 2018.

Jean-Charles Gilbert. Automatic differentiation and Iterative Processes. *Optimization Methods and Software*, 1:13–21, 1992.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2672–2680, 2014.

Karol Gregor and Yann Le Cun. Learning Fast Approximations of Sparse Coding. In *International Conference on Machine Learning (ICML)*, pages 399–406, 2010.

Andreas Griewank and Andrea Walther. *Evaluating derivatives: principles and techniques of algorithmic differentiation*, volume 105. Siam, 2008.

Laurent Hascoet and Mauricio Araya-Polo. Enabling user-driven Checkpointing strategies in Reverse-mode Automatic Differentiation. Technical Report RR-5930, Inria, 2006.

Dougal Maclaurin, David Duvenaud, and Ryan Adams. Gradient-based hyperparameter optimization through reversible learning. In *International Conference on Machine Learning (ICML)*, pages 2113–2122, 2015.

Julien Mairal, Francis R. Bach, Jean Ponce, and Guillermo Sapiro. Online Learning for Matrix Factorization and Sparse Coding. *Journal of Machine Learning Research (JMLR)*, 11(1):19–60, 2010.

Julien Mairal, Francis R. Bach, and Jean Ponce. Task-driven dictionary learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 34(4):791–804, 2012.

Eric Moulines and Francis R. Bach. Non-Asymptotic Analysis of Stochastic Approximation Algorithms for Machine Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 451–459, 2011.

Bruno A Olshausen and David J Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research*, 37(23):3311–3325, 1997.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems (NeurIPS)*, page 12, 2019.

Fabian Pedregosa. Hyperparameter optimization with approximate gradient. *preprint arXiv:1602.02355*, 2016.

Gabriel Peyré and Marco Cuturi. *Computational optimal transport*, volume 11. Now Publishers, Inc., 2019.

Amirreza Shaban, Ching-An Cheng, Nathan Hatch, and Byron Boots. Truncated Back-propagation for Bilevel Optimization. In *Artificial Intelligence and Statistics (AISTAT)*, pages 1723–1732, 2019.

Bahareh Tolooshams, Sourav Dey, and Demba Ba. Scalable convolutional dictionary learning with constrained recurrent sparse auto-encoders. In *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2018.

John von Neumann. Zur theorie der gesellschaftsspiele. *Mathematische annalen*, 100(1):295–320, 1928.

A. Experiments details and extra experiments

A.1. Experiments details

Super-efficiency of g^2 for gradient descent For Fig.2, the problem sizes are:

- **Ridge regression $\mathcal{L}_1$**: we use an overcomplete design matrix D with $n = 50$ and $m = 100$ with entries $D_{i,j}$ drawn *iid* from a normal distribution $\mathcal{N}(0, 1)$. The vector x to evaluate the gradient is sampled also with *iid* entries following a normal distribution. We take $\lambda = \frac{1}{n}$. To compute $g^*(x)$, we used the gradient descent with step-size $\frac{1}{L_1}$ for $14,000$ iterations.

- **Regularized Logistic regression $\mathcal{L}_2$**: we use an overcomplete design matrix D with $n = 50$ and $m = 100$ with entries $D_{i,j}$ drawn *iid* from a normal distribution $\mathcal{N}(0, 1)$. The vector x to evaluate the gradient is sampled also with *iid* entries following a normal distribution. We take $\lambda = \frac{1}{n}$. To compute $g^*(x)$, we used the gradient descent with step-size $\frac{1}{L_1}$ for $40,000$ iterations.

- **Least mean p-th norm $\mathcal{L}_3$**: In this setting, the convergence is much slower than in the previous ones. We use an overcomplete design matrix D with $n = 5$ and $m = 10$. To meet the condition of Prop.3.9, we sample the entries of a matrix $A \in \mathbb{R}^{n\times m}$ *iid* with normal distribution $\mathcal{N}(0, 1)$, take the SVD of $A = U^\top \Lambda V$ with $U \in \mathbb{R}^{n\times n}$ unitary and V and define $D = U^\top V$. This ensures that $DD^\top = I_n$ We choose $p = 4$, and use $g^* = 0$.

- **Wasserstein Distance $\mathcal{L}_4$**: we consider here the problem of computing the Wasserstein distance between two distributions supported on an Euclidean grid in $[0, 1]$ and C is defined as the ℓ_2 distance between the points of the grid. q is in Δ_+^n with $n = 100$ and $b \in \Delta_+^m$ with $m = 30$. We sample $a \in \Delta_+^m$ by first sampling the $\widehat{a}$ *iid* from a uniform distribution $\mathcal{U}(0, 1)$ and then take $a = \frac{\widehat{a}}{\sum_{l=1}^m a_l|}$. We used $\epsilon = 0.1$ and $g^*(x)$ is computed by running $2,000$ iteration of Sinkhorn.

Super-efficiency of g^2 for SGD For Fig.3, we consider for both experiments the penalized logistic loss and an overcomplete design matrix D with $n = 30$ and $m = 50$ with entries $D_{i,j}$ drawn *iid* from a normal distribution $\mathcal{N}(0, 1)$. The vector x to evaluate the gradient is sampled also with *iid* entries following a normal distribution. We take $\lambda = \frac{1}{n}$. To compute $g^*(x)$, we used the gradient descent with step-size $\frac{1}{L_1}$ for $1,000$ iterations.

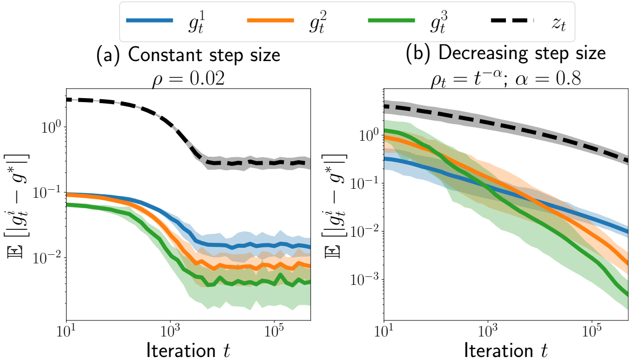

Figure A.1. Evolution of $\mathbb{E}|g^i - g^*|$ with the number of iterations t for (a) SGD with a constant step-size $\rho = 0.02$; (b) SGD with a decreasing step-size $\rho_t = t^{-\alpha}$ for $\alpha = 0.8$.

In Fig.3.(a), we compute the gradient estimates g_t^i using the output z_t of the SGD with constant step-size ρ for 20 values of $\rho \in [0.001, 0.1]$ in log-scale for 50 realization of the SGD. We report the mean value of $|g_t^i - g^*|$ computed for t large enough to have reached the regime where only the noise term is significant in Prop.3.5. This corresponds to the value of $\mathbb{E}|g_t^i - g^*|$ estimated by taking the value at the right end of the curve displayed in Fig.A.1.(a) for different values of ρ.

In Fig.3.(b), we illustrate the evolution with t of $\mathbb{E}|g_t^i - g^*|$ for g_t^i computed for SGD with decreasing step-sizes $t^{-\alpha}$ for $\alpha = .8$. The expectation is estimated by averaging 50 realizations of $|g_t^i - g^*|$ and the area around the curve correspond to the first and 9-th deciles.

A.2. Gradient descent with inexact gradients

To evaluate the impact of the different gradient estimators on the optimization of the global function ℓ, we run mirror descent for the loss ℓ defined in (21). We use $n = m = 1,000$ and $N = 20$ for the dimensions of $x \in \Delta_+^n$ and $b_i \in \Delta_+^m$ and we sample following the same procedure as for the Wasserstein Distance. We set $\epsilon = 0.05$ and we used a step-size of $\eta = 0.05$. We compute x^* by running the mirror descent algorithm with analytic gradient estimator g_t^1 for $t = 1,000$ and $q = 5,000$. Fig.A.3 reports the residual errors $\ell(x_q) - \ell(x^*)$ for g_t^1 and g_t^2 relatively to the number of iterations t used to compute them (a) as well as to the time taken to compute them (b). We exclude g^3 from this analysis as it is much more costly to compute in this case (see Sec.A.3) and it can be ill-conditionned – as it is illustrated in Fig.2. Fig.A.2 displays for each t used to compute Fig.4 the evolution of the cost function in iteration q and in time. We can see here in both figures that as the number of iteration t to compute the gradient increases, the final optimization error decrease, as predicted in Prop.4.1. However, the computational cost for g_t^2 scales with a factor c compared to computing g_t^1 and c is larger than 2 in this case (see Sec.A.3). As the convergence of z_t is linear, using g_t^2 is not beneficial for the global optimization as it possible

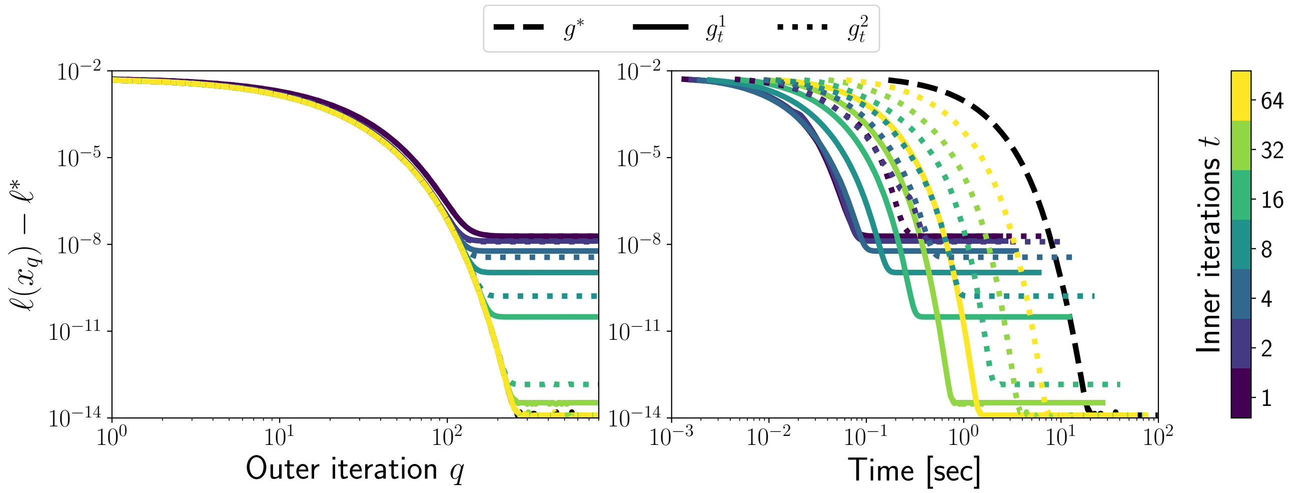

Figure A.2. Evolution of $\ell(x_q) - \ell*$ with q for different values of t and for analytic estimator g_t^1 and automatic estimator g_t^2.

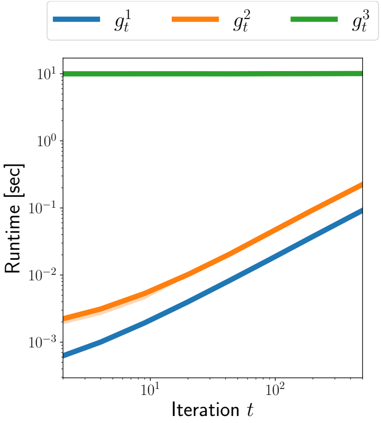

Figure A.3. Evolution of the computation time for the gradient estimators g_t^i with the number of iteration t.

to compute g_{ct}^1 instead which reduces the optimization error compared to g_t^1, as discussed in Sec.4.2.

A.3. Computation time of the gradient

To evaluate the relative computational cost of the gradient estimates g_t^i, we time the computation of the gradient using the loss ℓ defined in (21). We use $n = 500$, $m = 1,000$ and $N = 100$ for the dimensions of $x \in \Delta_+^n$ and $b_i \in \Delta_+^m$ and we sample following the same procedure as for the Wasserstein Distance. Then, we time the computational time for the gradient estimators g_t^i for different values of t, and report in Fig.A.3 the median value of this computation time computed on 50 realization as well as the first and last decile values to get an idea of the variation of this value. The results are coherent with the computational complexity analysis in Table 1, g^1 and g^2 computation time scales linearly with t, with a constant factor between them which capture the value of c which is around 3.5 in our case. For this scale of problem, g^3 requires inverting N matrices $n \times n$. This cost dominates the cost of computing z_t, g_t^1 and g_t^2 for small value of t and it becomes less prohibitive as t grows.

B. Proof for Sec.2

Proposition 2.3. *[Convergence of the implicit estimator] Assume that $\mathcal{J}$ is $L_\mathcal{J}$-Lipschitz with respect to its first argument, and that $\|\mathcal{J}_t\| \leq L_J$. Then, for L as defined in (9),*

$$|g_t^3 - g^*| \leq (\frac{L}{2} + L_\mathcal{J} L_1)|z_t - z^*|^2 \ . \qquad (10)$$

Proof. We define $\mathcal{J}_t = \mathcal{J}(z_t, x)$. The implicit gradient g^3 reads

$$g^3 = g^* + R(\mathcal{J}_t)(z_t - z^*) + R_{21} + \mathcal{J}_t R_{11} \ .$$

Then, we have

$$R(\mathcal{J}_t) = R(\mathcal{J}_t) - R(J^*) = (\mathcal{J}_t - J^*)\, \nabla_{11}^2 \mathcal{L}(z^*, x)$$

We recall that $J^* = \mathcal{J}(z^*, x)$. It follows that

$$\|\mathcal{J}_t - J^*\| \leq L_\mathcal{J} |z_t - z^*| \ ,$$

and

$$\|R(\mathcal{J}_t)\| \leq L_\mathcal{J} L_1 |z_t - z^*| \ .$$

The result follows using Equation 8. □

C. Proof Sec.3

C.1. Proof of Prop.3.2

Proposition 3.2. *[Convergence speed of the Jacobian of gradient descent in a strongly convex setting] Let z_t produced by the recursion $z_{t+1} = z_t - \rho \nabla_1 \mathcal{L}(z_t, x)$ with*

$\rho \leq 1/L_1$ and $\kappa \triangleq 1-\rho\mu$. We have $|z_t - z^*| \leq \kappa^t |z_0 - z^*|$ and $\|J_t - J^*\| \leq t\kappa^{t-1}\rho L|z_0 - z^*|$ where L is defined in (9).

Proof. Differentiating the gradient descent recursion, we find that J_t follows the recursion

$$J_{t+1} = J_t - \rho G_t \ \ , \tag{22}$$

where $G_t = J_t \nabla^2_{11}\mathcal{L}\left(z_t, x\right) + \nabla^2_{21}\mathcal{L}\left(z_t, x\right)$. Using $\|I - \rho\nabla^2_{11}\mathcal{L}(z,x)\| \leq \kappa$, a first crude upper bounding gives

$$\|J_{t+1}\| \leq \kappa\|J_t\| + \alpha \ \ ,$$

where α is an upper-bound of $\|\nabla^2_{21}\mathcal{L}\|$. This shows that $\|J_t\|$ is bounded. Next, denoting $\tilde{G}_t = J_t\nabla^2_{11}\mathcal{L}\left(z^*, x\right) + \nabla^2_{21}\mathcal{L}\left(z^*, x\right)$, we find

$$\begin{aligned}\Delta^t \triangleq G_t - \tilde{G}_t =& J_t\left(\nabla^2_{11}\mathcal{L}\left(z_t, x\right) - \nabla^2_{11}\mathcal{L}\left(z^*, x\right)\right) \\ &+ \nabla^2_{21}\mathcal{L}\left(z_t, x\right) - \nabla^2_{21}\mathcal{L}\left(z^*, x\right)\end{aligned}$$

Using the third-order differentiability of $\mathcal{L}$, the rate of convergence of z_t, and that J_t is bounded, we find that there exists $\beta > 0$ such that $\|\Delta_t\| \leq \beta\kappa_t$. Eq. 22 finally gives

$$\begin{aligned}J_{t+1} - J^* = &\left(Id - \rho\nabla^2_{11}\mathcal{L}(z^*, x)\right)(J_t - J^*) \\ &- \rho\nabla^2_{11}\mathcal{L}(z^*, x)\Delta_t \ \ .\end{aligned}$$

Taking norms and using the triangular inequality, we find

$$\|J_{t+1} - J^*\| \leq \kappa\|J_t - J^*\| + \gamma\kappa^t \ \ ,$$

where $\gamma = \rho\mu\beta$. Unrolling the recursion gives, as expected,

$$\|J_t - J^*\| \leq \gamma t\kappa^{t-1} \ \ .$$

□

C.2. Tightness of the bound in Prop.3.2

Importantly, the rate of $O(t\kappa^t)$ given in Prop.3.2 is tight. Indeed, it is reached with in the following example.

Proposition C.1. *For $x \in \mathbb{R}^n$, let $\lambda_x = \sum_{i=1}^m x_i$. Consider $\mathcal{L}(z, x) = \frac{1}{2}\lambda_x\|z\|^2$. The iterates produced by gradient descent with step $\rho \leq 1/\lambda_x$ verify $z_t = \kappa^t z_0$ with $\kappa = 1 - \rho\lambda_x$, and we have $J_t = -\rho t\kappa^{t-1}\mathbb{1}_n z_0^\top$.*

C.3. Tightness of the bound in Prop.3.6

The rate of $O(t^{-\alpha})$ given in Prop.3.6 is tight. Indeed, it is reached with in the following example.

Proposition C.2. *Consider samples ξ drawn from $\mathcal{N}(0, 1)$. Let $C(z, x, \xi) = \frac{1}{2}|\xi z - x|^2$. SGD on this function with steps $\rho_t = t^{-\alpha}$ produces a sequence of Jacobians such that $\delta_t = t^{-\alpha} + o(t^{-\alpha})$*

Proof. The iterates verify:

$$z_{t+1} = z_t - \rho_t\xi_{t+1}(\xi_{t+1}z_t - x)$$

and the Jacobian sequence:

$$\begin{aligned}J_{t+1} &= J_t - \rho_t\xi_{t+1}(\xi_{t+1}J_t - I_n) && (23)\\ &= (1 - \rho_t\xi^2_{t+1})J_t + \rho_t\xi_{t+1}I_n && (24)\end{aligned}$$

Hence:

$$\begin{aligned}\mathbb{E}_{\xi_{t+1}}[\|J_{t+1}\|^2] = \mathbb{E}_{\xi_{t+1}}[&(1 - \rho_t\xi^2_{t+1})^2\|J_t\|^2 + \rho_t^2\xi^2_{t+1} && (25)\\ &+ 2\rho_t\xi_{t+1}(1 - \rho_t\xi^2_{t+1})J_t)] && (26)\end{aligned}$$

which gives:

$$\mathbb{E}_{\xi_{t+1}}[\|J_{t+1}\|^2] = (1 - 2\rho_t + 3\rho_t^2)\|J_t\|^2 + \rho_t^2$$

and taking expectations over the whole past:

$$\delta_{t+1} = (1 - 2\rho_t + 3\rho_t^2)\delta_t + \rho_t^2$$

Taking $\rho_t = t^{-\alpha}$ and unrolling the recursion gives

$$\delta_t = t^{-\alpha} + o(t^{-\alpha})$$

which proves the tightness of the bound.

□

C.4. Proof of Prop.3.3

Proposition 3.3. *Define*

$$\delta_t = \mathbb{E}\left[\|J_t - J^*\|^2_F\right] \text{ and } d_t = \mathbb{E}\left[|z_t - z^*|^2\right] \ \ . \tag{14}$$

We have $\mathbb{E}[|g^2 - g^*|] \leq L_1\sqrt{\delta_t}\sqrt{d_t} + \frac{L}{2}d_t$.

Proof. Taking expectations in Prop.2.2 and using Equation 12 gives

$$\mathbb{E}[|g^2 - g^*|] \leq L_1\mathbb{E}[\|J_t - J^*\||z_t - z^*|] + \frac{L}{2}\underbrace{\mathbb{E}[|z_t - z^*|^2]}_{d_t} \ \ .$$

Cauchy-Schwarz on the first term gives

$$\mathbb{E}[\|J_t - J^*\||z_t - z^*|] \leq \sqrt{\mathbb{E}\left[\|J_t - J^*\|^2\right]}\sqrt{d_t}$$

and then $\|J_t - J^*\|^2 \leq \|J_t - J^*\|^2_F$ gives the advertised result. □

C.5. Proof of Prop.3.4

Proposition 3.4. *[Bounding inequality for the Jacobian] We assume bounded Hessian noise, in the sense that* $\mathbb{E}\left[\|\nabla^2_{11}C(z,x,\xi)\|_F^2\right] \leq \sigma_{11}^2$ *and* $\mathbb{E}\left[\|\nabla^2_{21}C(z,x,\xi)\|_F^2\right] \leq \sigma_{21}^2$*. Let* $r = \min(n,m)$*, and* $B^2 = \sigma_{21}^2 + L_J^2\sigma_{11}^2$*. We have*

$$\delta_{t+1} \leq (1-2\rho_t\mu)\delta_t + 2\rho_t\sqrt{r}L\sqrt{d_t}\sqrt{\delta_t} + \rho_t^2B^2. \quad (15)$$

Proof. Let $U_t = \nabla^2_{11}C(z_t,x,\xi_{t+1})J_t + \nabla^2_{12}C(z_t,x,\xi_{t+1})$. We have $\mathbb{E}_{\xi_{t+1}}[U_t] = \nabla^2_{11}\mathcal{L}(z^*,x)(J_t - J^*) + \Delta_t$, where $\Delta_t = \left(\nabla^2_{11}\mathcal{L}(z_t,x) - \nabla^2_{11}\mathcal{L}(z^*,x)\right)J_t + \nabla^2_{12}\mathcal{L}(z_t,x) - \nabla^2_{12}\mathcal{L}(z^*,x)$. We find

$$\begin{aligned}\|J_{t+1} - J^*\|_F^2 =& \|J_t - J^*\|_F^2 \\ & - 2\rho_t\langle J_t - J^*, U_t\rangle_F + \rho_t^2\|U_t\|_F^2\end{aligned}$$

Taking expectations with respect to ξ_{t+1} yields

$$\begin{aligned}\mathbb{E}_{\xi_{t+1}}\left[\|J_{t+1} - J^*\|_F^2\right] = & \\ & \|J_t - J^*\|_F^2 \\ & - 2\rho_t\langle J_t - J^*, \nabla^2_{11}\mathcal{L}(z^*,x)(J_t - J^*)\rangle_F \\ & - 2\rho_t\langle J_t - J^*, \Delta_t\rangle_F + \rho_t^2\mathbb{E}_{\xi_{t+1}}\left[\|U_t\|_F^2\right]\end{aligned}$$

Using strong-convexity for the second term and Cauchy-Schwarz for the third term, we find

$$\begin{aligned}\mathbb{E}_{\xi_{t+1}}\left[\|J_{t+1} - J^*\|_F^2\right] \leq & (1-2\rho_t\mu)\|J_t - J^*\|_F^2 \\ & + 2\rho_t\|J_t - J^*\|_F\|\Delta_t\|_F \\ & + \rho_t^2\mathbb{E}_{\xi_{t+1}}\left[\|U_t\|_F^2\right]\end{aligned}$$

Taking expectations over the whole past

$$\begin{aligned}\delta_{t+1} \leq & (1-2\rho_t\mu)\delta_t \\ & + 2\rho_t\mathbb{E}\left[\|J_t - J^*\|_F\|\Delta_t\|_F\right] + \rho_t^2\mathbb{E}\left[\|U_t\|_F^2\right]\end{aligned}$$

To majorize the last term,

$$\|U_t\|_F^2 \leq \underbrace{\|\nabla^2_{11}C(z,x,\xi_{t+1})J_t\|_F^2}_{\leq\|J_t\|^2\|\nabla^2_{11}C(z,x,\xi_{t+1}\|_F^2} + \|\nabla^2_{21}C(z,x,\xi_{t+1}\|_F^2$$

Therefore

$$\mathbb{E}\left[\|U_t\|_F^2\right] \leq L_J\sigma_{11}^2 + \sigma_{21}^2$$

Cauchy-Schwarz on the middle term yields

$$\begin{aligned}\mathbb{E}\left[\|J_t - J^*\|_F\|\Delta_t\|_F\right] \leq & \sqrt{r}\sqrt{\delta_t}\sqrt{\mathbb{E}[\|\Delta_t\|^2]} \\ \leq & \sqrt{r}L\sqrt{\delta_t d_t} \ .\end{aligned}$$

Combining everything provides the final bound. □

C.6. Proof of Prop.3.5

Proposition 3.5. *[SGD with constant step-size] Assume that the gradients have bounded variance* $\mathbb{E}_\xi[|\nabla_1C(z,x,\xi)|^2] \leq \sigma^2$*. Assume* $\rho_t = \rho < 1/L_1$*, and let* $\kappa_2 = \sqrt{1-2\rho\mu}$ *and* $\beta = \sqrt{\frac{\sigma^2\rho}{2\mu}}$*. In this setting*

$$\delta_t \leq \left(\kappa_2^t\left(\|J^*\|_F + t\alpha\right) + B_2\right)^2 \ ,$$

where $\alpha = \frac{\rho\sqrt{r}L}{\kappa_2}|z^* - z_0|$ *and* $B_2 = \frac{\rho\sqrt{r}L\beta}{\kappa_2(1-\kappa_2)} + \frac{\rho B}{(1-\kappa_2)}$*.*

Proof. We start by obtaining a simpler bound than 15 by completing the squares

$$\begin{aligned}(1-2\rho_t\mu)\delta_t + 2\sqrt{r}L\rho_t\sqrt{d_t}\sqrt{\delta_t} \leq & \\ \left(\sqrt{1-2\rho_t\mu}\sqrt{\delta_t} + \frac{\sqrt{r}L\rho_t}{\sqrt{1-2\rho_t\mu}}\sqrt{d_t}\right)^2 &\end{aligned}$$

And bouding crudely $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$, we obtain a simple recursion on $\sqrt{\delta_t}$

$$\sqrt{\delta_{t+1}} \leq \sqrt{1-2\rho_t\mu}\sqrt{\delta_t} + \frac{\sqrt{r}L\rho_t}{\sqrt{1-2\rho_t\mu}}\sqrt{d_t} + \rho_tB \ . \quad (27)$$

(Moulines and Bach, 2011) give

$$\mathbb{E}[|z_t - z^*|^2] \leq (1-2\rho\mu)^t|z_0 - z^*|^2 + \beta^2 \ .$$

Using $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$ for $a, b \geq 0$, we get

$$\sqrt{d_t} \leq \kappa_2^t|z_0 - z^*| + \beta$$

Eq. (27) then gives

$$\sqrt{\delta_{t+1}} \leq \kappa_2\sqrt{\delta_t} + \rho\kappa_2^t + (1-\kappa_2)B_2$$

Unrolling the recursion and using $\sum_{i=0}^t \kappa_2^i \leq \frac{1}{1-\kappa_2}$ gives the proposed bound on δ_t. □

C.7. Proof of Prop.3.6

Proposition 3.6. *[SGD with decreasing step-size] Assume that* $\rho_t = \rho_0t^{-\alpha}$ *with* $\alpha \in (0,1)$*. Assume a bound on* d_t *of the form* $d_t \leq d^2t^{-\alpha}$*. Then*

$$\delta_t \leq 4\frac{\rho_0B^2\mu + rL^2d^2}{\mu^2}t^{-\alpha} + o(t^{-\alpha}) \ .$$

Proof. Under the assumptions, Eq. (15) becomes

$$\delta_{t+1} \leq (1-2\mu Ct^{-\alpha})\delta_t + 2\sqrt{r}LdCt^{-\frac{3}{2}\alpha}\sqrt{\delta_t} + B^2C^2t^{-2\alpha} \ .$$

We get rid of the problematic middle term using the inequality, valid for all $\chi > 0$,

$$t^{-\frac{3}{2}\alpha}\sqrt{\delta_t} \leq \frac{1}{2}\left(\frac{\chi\delta_t}{t^\alpha} + \frac{1}{\chi t^{2\alpha}}\right),$$

which gives

$$\delta_{t+1} \le \left(1 - (2\mu C - \chi\sqrt{r}LdC)t^{-\alpha}\right)\delta_t \\ + (B^2C^2 + \frac{\sqrt{r}LdC}{\chi})t^{-2\alpha} \ .$$

We take $\chi = \frac{\mu}{\sqrt{rLd}}$, so that the first term becomes $1 - \mu C t^{-\alpha}$. Note that it does not give optimal rates, but makes computations much simpler. In (Moulines and Bach, 2011), it is shown that for $a, b > 0$, a recursion satisfying

$$\delta_{t+1} \le (1 - at^{-\alpha})\delta_t + bt^{-2\alpha}$$

verifies $\delta_t \le 4\frac{b}{a}t^{-\alpha} + o(t^{-\alpha})$. The result follows by taking $a = \mu C$ and $b = B^2C^2 + \frac{rL^2d^2C}{\mu^2}$. □

C.8. Proof of Prop.3.9

Proposition 3.9. *For $p \ge 4$, we assume $DD^\top = I_n$. Let $\alpha \triangleq \frac{p-1}{p-2}$. We have*

$$|g_t^1| = O(t^{-\alpha}), \quad |g_t^2| = O(t^{-2\alpha}), \quad g_t^3 = 0 \ .$$

Proof. Gradient descent iterates

$$z_{t+1} = z_t - \rho D^\top(Dz_t - x)^{p-1} \ .$$

The residuals $r_t = x - Dz_t$ therefore verify the recursion

$$r_{t+1} = r_t - \rho DD^\top r_t^{p-1} \ .$$

Since we assume $DD^\top = I_n$, they verify $r_{t+1} = r_t - \rho r_t^{p-1}$. Each entry of r_t therefore evolves independently, following the 1-d recursive equation

$$u_{t+1} = u_t - \rho u_t^{p-1} \qquad (28)$$

Standard analysis techniques show that this gives $u_t = \left(\frac{1}{\rho(p-2)t}\right)^{\frac{1}{p-2}}(1 + O(\frac{\log(t)}{t}))$, and therefore each coefficient of r_t satisfies the same asymptotic development. The Jacobian verifies

$$J_{t+1} = J_t - (p-1)\rho(J_tD^\top - I_m)\operatorname{Diag}(r_t^{p-2})D,$$

and denoting $M_t = I_n - J_tD^\top$, we find

$$M_{t+1} = M_t(I_n - (p-1)\rho\operatorname{Diag}(r_t^{p-2})) \ .$$

Since the rightmost term is diagonal, we can rewrite this recursion as:

$$M_{t+1} = M_t\operatorname{Diag}(\mathbb{1}_n - (p-1)\rho r_t^{p-2})$$

Unrolling this recursion gives:

$$M_t = M_0\operatorname{Diag}(\prod_{j\le t-1}\mathbb{1} - (p-1)\rho r_t^{p-2})$$

We can then majorize each coefficient in the Diag by:

$$\prod_{j\le t-1}(1 - (p-1)\rho u_j^{p-2}) \le \exp(\sum_{j\le t-1} -(p-1)\rho u_j^{p-2}) \ ,$$

where u_t follows the recursion (28). The asymptotic development of u_t gives:

$$u_t^{p-2} = \frac{1}{\rho(p-2)t} + O(\frac{\log(t)}{t^2})$$

and as a consequence, denoting $\alpha = \frac{p-1}{p-2}$:

$$\exp(\sum_{j\le t-1} -(p-1)\rho u_j^{p-2}) = O(t^{-\alpha})$$

Overall, we have $M_t = O(t^{-\alpha})$ and $r_t^{p-1} = O(t^{-\alpha})$, so $g_2 = M_tr_t^{p-1} = O(t^{-2\alpha})$. □

D. Proof of Prop.4.2

We start by giving the convergence rate of the SGD with (δ, L, μ)-inexact oracle for a function f defined on $x \in \mathbb{R}^n$ as

$$f(x) = \mathbb{E}_\upsilon[F(x, \upsilon)]$$

for υ a random variable distributed with probability d_υ. For $\upsilon_0 \sim d_\upsilon$, we denote $(F_\delta(\cdot, \upsilon_0), G_\delta(\cdot, \upsilon_0))$ a (δ, L, μ)-inexact oracle of $F(\cdot, \upsilon_0)$, uniform in υ_0.

Lemma D.1. *For a μ-strongly convex L-smooth function f and a (δ, μ, L)-inexact oracle (F_δ, G_δ) of F such that*

$$\mathbb{E}_\upsilon[F_\delta(x,\upsilon)] = f_\delta(x) \ , \ \mathbb{E}_\upsilon[G_\delta(x,\upsilon)] = g_\delta(x) \ , \\ \textit{and} \quad \mathbb{E}_\upsilon[|G_\delta(x,\upsilon) - g_\delta(x)|^2] \le \sigma^2 \ .$$

Then, the iterates of the stochastic gradient descent with constant step-size $\eta < \frac{1}{L}$ verify

$$\mathbb{E}|x_q - x^*|^2 \le (1 - \eta\mu)^q|x_0 - x^*|^2 + \frac{\eta}{\mu}\sigma^2 + \frac{2}{\mu}\delta.$$

Proof. Consider the solution estimate x_{q+1} at iteration q, obtained through stochastic gradient descent *i.e.* $x_{q+1} = x_q - \eta\nabla G_\delta(x_q, \upsilon_{q+1})$. We denote $r_{q+1} = \mathbb{E}[|x_{q+1} - x^*|^2]$ and $\widehat{r}_{q+1} = \mathbb{E}[|x_{q+1} - x^*|^2|\upsilon_{q+1}]$. Then

$$|x_{q+1} - x^*|^2 = |x_q - x^*|^2 - 2\eta\langle G_\delta(x_q, \upsilon_{q+1}), x_q - x^*\rangle \\ + \eta^2|G_\delta(x_q, \upsilon_{q+1})|^2$$

We take the expectation relatively to υ_{q+1}

$$\begin{aligned}\widehat{r}_{q+1} \le& |x_q - x^*|^2 - 2\eta\langle g_\delta(x_q), x_q - x^*\rangle \\ &+ \eta^2|g_\delta(x_q)|^2 + \eta^2\mathbb{E}_{\upsilon_q}[|G_\delta(x_q, \upsilon_q) - g_\delta(x_q)|^2] \\ \le& |x_q - x^*|^2 + 2\eta\langle g_\delta(x_q), x^* - x_q\rangle \\ &+ \eta^2|g_\delta(x_q)|^2 + \eta^2\sigma^2\end{aligned}$$

Using $\langle g_\delta | x^* - x_q\rangle \leq f(x^*) - f_\delta(x_q) - \frac{\mu}{2}|x_q - x^*|^2$, we get

$$
\begin{aligned}
\widehat{r}_{q+1}^2 \leq & |x_q - x^*|^2 + 2\eta(f(x^*) - f_\delta(x_q) - \frac{\mu}{2}|x_q - x^*|^2) \\
& + \eta^2 |g_\delta(x_q)|^2 + \eta^2\sigma^2 \\
\leq & (1-\eta\mu)|x_q - x^*|^2 + \eta^2\sigma^2 \\
& + 2\eta(f(x^*) - f_\delta(x_q) + \frac{\eta}{2}|g_\delta(x_q)|^2)
\end{aligned}
$$

We introduce $\widetilde{x} = x_q - \eta g_\delta(x_q)$. Using the rhs of the δ-inexact oracle, we have

$$
f(\widetilde{x}) - f_\delta(x_q) - \langle g_\delta(x_q) | \widetilde{x} - x_q\rangle \leq \frac{L\eta^2}{2}|g_\delta(x_q)|^2 + \delta
$$

i.e

$$
f(\widetilde{x}) - f_\delta(x_q) + \frac{\eta}{2}|g_\delta(x_q)|^2 \leq \delta - (1-\eta L)\frac{\eta}{2}|g_\delta(x_q)|^2 \ .
$$

Using this in the previous equation, we obtain

$$
\begin{aligned}
r_{q+1}^2 \leq & (1-\eta\mu)|x_q - x^*|^2 + \eta^2\sigma^2 + 2\eta\delta \\
& + \underbrace{2\eta(f(x^*) - f(\widetilde{x})) - (1-\eta L)\eta^2|g_\delta(x_q)|^2}_{\leq 0} \ ,
\end{aligned}
$$

where the two terms on the second line are non-positive. Indeed, $\eta \leq \frac{1}{L}$ and $f(x^*) \leq f(\widetilde{x})$. Taking the expectation relatively to $\upsilon_0, \dots, \upsilon_{q-1}$ gives the following recursion relationship

$$
r_{q+1}^2 \leq (1-\eta\mu) r_q^2 + \eta^2\sigma^2 + 2\eta\delta \ .
$$

applying this recursion q times yields

$$
r_{q+1}^2 \leq (1-\eta\mu)^q r_0^2 + (\eta^2\sigma^2 + 2\eta\delta)\sum_{k=0}^{q}(1-\eta\mu)^k \ ,
$$

and we obtain the desired results as $\sum_{k=0}^{q}(1-\eta\mu) < \frac{1}{\eta\mu}$ □

Proposition 4.2. *We assume that H is μ_2-strongly convex, L_2-smooth and verifies*

$$
\mathbb{E}[|\nabla h(x,\upsilon) - \nabla \ell(x)|^2] \leq \sigma^2.
$$

The iterates x_q of SGD with approximate gradient g^i and step-size η verify

$$
\mathbb{E}|x_q - x^*|^2 \leq (1 - \frac{\eta\mu_2}{2})^q |x_0 - x^*| + \frac{2\eta}{\mu_2}\sigma^2 + \frac{4}{\mu_2}\delta_i
$$

with $\delta_i = \Delta_i^2(\frac{1}{\mu_2} + \frac{1}{2L_2} + 2\eta)$.

Proof. As shown in Sec.4.1, (ℓ, g^i) is a $(\widetilde{\delta}_i, \frac{\mu_2}{2}, 2L_2)$-inexact oracle with $\widetilde{\delta}_i = \Delta_i^2(\frac{1}{\mu_2} + \frac{1}{2L_2})$.

The variance of the stochastic inexact oracles can be bounded as follow

$$
\begin{aligned}
\mathbb{E}[|g^i(x,\upsilon) - g^i(x)|^2] \leq & 2(E[|g^i(x,\upsilon) - \nabla h(x,\upsilon)|^2 \\
& + |\nabla h(x,\upsilon) - \nabla \ell(x)|^2 \qquad (29) \\
& + |\nabla \ell(x) - g^i(x)|^2]) \ , \\
\leq & \underbrace{2(\sigma^2 + 2\Delta_i^2)}_{\widetilde{\sigma}^2} \ . \qquad (30)
\end{aligned}
$$

Using these two bounds with the previous result yield

$$
\begin{aligned}
\mathbb{E}|x_q - x^*|^2 \leq & (1 - \frac{\eta\mu}{4})^q |x_0 - x^*| + \frac{4\eta}{\mu}\widetilde{\sigma}^2 + \frac{2}{\mu}\widetilde{\delta} \qquad (31) \\
\leq & (1 - \frac{\eta\mu}{4})^q |x_0 - x^*| + \frac{4\eta}{\mu}\sigma^2 + \frac{2}{\mu}\delta \qquad (32)
\end{aligned}
$$

with $\delta = \Delta_i^2(\frac{1}{L} + \frac{1}{\mu} + 2\eta)$

□