Statistically Preconditioned Accelerated Gradient Method for Distributed Optimization

Hadrien Hendrikx, Lin Xiao, Sébastien Bubeck, Francis Bach, Laurent Massoulié

Statistically Preconditioned Accelerated Gradient Method for Distributed Optimization

Hadrien Hendrikx [1] **Lin Xiao** [2] **Sébastien Bubeck** [2] **Francis Bach** [1] **Laurent Massoulié** [1]

Abstract

We consider the setting of distributed empirical risk minimization where multiple machines compute the gradients in parallel and a centralized server updates the model parameters. In order to reduce the number of communications required to reach a given accuracy, we propose a *preconditioned* accelerated gradient method where the preconditioning is done by solving a local optimization problem over a subsampled dataset at the server. The convergence rate of the method depends on the square root of the relative condition number between the global and local loss functions. We estimate the relative condition number for linear prediction models by studying *uniform* concentration of the Hessians over a bounded domain, which allows us to derive improved convergence rates for existing preconditioned gradient methods and our accelerated method. Experiments on real-world datasets illustrate the benefits of acceleration in the ill-conditioned regime.

1. Introduction

We consider empirical risk minimization problems of the form

$$\underset{x\in\mathbb{R}^d}{\text{minimize}} \quad \Phi(x) \triangleq F(x) + \psi(x), \tag{1}$$

where F is the empirical risk over a dataset $\{z_1, \dots, z_N\}$:

$$F(x) = \frac{1}{N}\sum_{i=1}^{N} \ell(x, z_i), \tag{2}$$

and ψ is a convex regularization function. We incorporate smooth regularizations such as squared Euclidean norms $(\lambda/2)\|x\|^2$ into the individual loss functions $\ell(x, z_i)$, and leave ψ mainly for non-smooth regularizations such as the ℓ_1-norm or the indicator function of a constraint set.

In modern machine learning applications, the dataset is often very large and has to be stored at multiple machines. For simplicity of presentation, we assume $N = mn$, where m is the number of machines and n is the number of samples stored at each machine. Let $\mathcal{D}_j = \{z_1^{(j)}, \dots, z_n^{(j)}\}$ denote the dataset at machine j and define the local empirical risk

$$f_j(x) = \frac{1}{n}\sum_{i=1}^{n} \ell\big(x, z_i^{(j)}\big), \quad j = 1, \dots, m. \tag{3}$$

The overall empirical risk of Equation (2) can then be written as

$$F(x) = \frac{1}{m}\sum_{j=1}^{m} f_j(x) = \frac{1}{nm}\sum_{j=1}^{m}\sum_{i=1}^{n} \ell\big(x, z_i^{(j)}\big).$$

We assume that F is L_F-smooth and σ_F-strongly convex over $\operatorname{dom}\psi$, in other words,

$$\sigma_F I_d \preceq \nabla^2 F(x) \preceq L_F I_d, \quad \forall x \in \operatorname{dom}\psi, \tag{4}$$

where I_d is the $d \times d$ identity matrix. The condition number of F is defined as $\kappa_F = L_F/\sigma_F$.

We focus on a basic setting of distributed optimization where the m machines (workers) compute the gradients in parallel and a centralized server updates the variable x. Specifically, during each iteration $t = 0, 1, 2, \dots$,

(i) the server broadcasts x_t to all m machines;

(ii) each machine j computes the gradient $\nabla f_j(x_t)$ and sends it back to the server;

(iii) the server forms $\nabla F(x_t) = \frac{1}{m}\sum_{j=1}^{m} \nabla f_j(x_t)$ and uses it to compute the next iterate x_{t+1}.

A standard way for solving problem (1) in this setting is to implement the proximal gradient method at the server:

$$x_{t+1} = \underset{x\in\mathbb{R}^d}{\operatorname{argmin}} \Big\{ \nabla F(x_t)^\top x + \psi(x) + \frac{1}{2\eta_t}\|x - x_t\|^2 \Big\}, \tag{5}$$

where $\|\cdot\|$ denotes the Euclidean norm and $\eta_t > 0$ is the step size. Setting $\eta_t = 1/L_F$ leads to linear convergence:

$$\Phi(x_t) - \Phi(x_*) \le \big(1 - \kappa_F^{-1}\big)^t \, \tfrac{L_F}{2}\|x_* - x_0\|^2, \tag{6}$$

[1]INRIA, DIENS, PSL Research University, Paris, France. [2]Microsoft Research, Redmond, WA, USA.. Correspondence to: Hadrien Hendrikx <hadrien.hendrikx@inria.fr>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

where $x_* = \arg\min \Phi(x)$ (e.g., Beck, 2017, Section 10.6). In other words, in order to reach $\Phi(x_t) - \Phi(x_*) \le \epsilon$, we need $O(\kappa_F \log(1/\epsilon))$ iterations, which is also the number of communication rounds between the workers and the server. If we use accelerated proximal gradient methods (e.g., Nesterov, 2004; Beck and Teboulle, 2009; Nesterov, 2013) at the server, then the iteration/communication complexity can be improved to $O(\sqrt{\kappa_F} \log(1/\epsilon))$.

1.1. Statistical Preconditioning

In general, for minimizing $F(x) = (1/m)\sum_{j=1}^m f_j(x)$ with first-order methods, the communication complexity of $O(\sqrt{\kappa_F}\log(1/\epsilon))$ cannot be improved (Arjevani and Shamir, 2015; Scaman et al., 2017). However, for distributed empirical risk minimization (ERM), the additional finite-sum structure of each f_j in (3) allows further improvement. A key insight here is that if the datasets $\mathcal{D}_j$ at different workers are i.i.d. samples fr (3om the same source distribution, then the local empirical losses f_j are statistically very similar to each other and to their average F, especially when n is large. *Statistical preconditioning* is a technique to further reduce communication complexity based on this insight.

An essential tool for preconditioning in first-order methods is the Bregman divergence. The Bregman divergence of a strictly convex and differentiable function ϕ is defined as

$$D_\phi(x, y) \triangleq \phi(x) - \phi(y) - \nabla\phi(y)^\top (x - y). \tag{7}$$

We also need the following concepts of relative smoothness and strong convexity (Bauschke et al., 2017; Lu et al., 2018).

Definition 1. *Suppose $\phi : \mathbb{R}^d \to \mathbb{R}$ is convex and twice differentiable. The function F is said to be $L_{F/\phi}$-smooth and $\sigma_{F/\phi}$-strongly convex with respect to ϕ if for all $x \in \mathbb{R}^d$,*

$$\sigma_{F/\phi}\nabla^2\phi(x) \preceq \nabla^2 F(x) \preceq L_{F/\phi}\nabla^2\phi(x). \tag{8}$$

The classical definition in (4) can be viewed as relative smoothness and strong convexity where $\phi(x) = (1/2)\|x\|^2$. Moreover, it can be shown that (8) holds if and only if for all $x, y \in \mathbb{R}^d$

$$\sigma_{F/\phi} D_\phi(x, y) \le D_F(x, y) \le L_{F/\phi} D_\phi(x, y). \tag{9}$$

Consequently, we define the relative condition number of F with respect to ϕ as $\kappa_{F/\phi} = L_{F/\phi}/\sigma_{F/\phi}$.

Following the Distributed Approximate Newton (DANE) method by Shamir et al. (2014), we construct the reference function ϕ by adding some extra regularization to one of the local loss functions (say f_1, without loss of generality):

$$\phi(x) = f_1(x) + \frac{\mu}{2}\|x\|^2. \tag{10}$$

Then we replace $(1/2)\|x - x_t\|^2$ in the proximal gradient method (5) with the Bregman divergence of ϕ, i.e.,

$$x_{t+1} = \operatorname*{argmin}_{x\in\mathbb{R}^d} \left\{ \nabla F(x_t)^\top x + \psi(x) + \frac{1}{\eta_t} D_\phi(x, x_t) \right\}. \tag{11}$$

In this case, worker 1 acts as the server to compute x_{t+1}, which requires solving a nontrivial optimization problem involving the local loss function f_1.

According to Shamir et al. (2014) and Lu et al. (2018), with $\eta_t = 1/L_{F/\phi}$, the sequence $\{x_t\}$ generated by (11) satisfies

$$\Phi(x_t) - \Phi(x_*) \le \left(1 - \kappa_{F/\phi}^{-1}\right)^t L_{F/\phi} D_\phi(x_*, x_0), \tag{12}$$

which is a direct extension of (6). Therefore, the effectiveness of preconditioning hinges on how much smaller $\kappa_{F/\phi}$ is compared to κ_F. Roughly speaking, the better f_1 or ϕ approximates F, the smaller $\kappa_{F/\phi}$ (≥ 1) is. In the extreme case of $f_1 \equiv F$ (with only one machine $m = 1$), we can choose $\mu = 0$ and thus $\phi \equiv F$, which leads to $\kappa_{F/\phi} = 1$, and we obtain the solution within one step.

In general, we choose μ to be an upper bound on the spectral norm of the matrix difference $\nabla^2 f_1 - \nabla^2 F$. Specifically, we assume that *with high probability*, for the operator norm between matrices (i.e., the largest singular value),

$$\left\|\nabla^2 f_1(x) - \nabla^2 F(x)\right\| \le \mu, \quad \forall\, x \in \operatorname{dom}\psi, \tag{13}$$

which implies (Zhang and Xiao, 2018, Lemma 3),

$$\frac{\sigma_F}{\sigma_F + 2\mu}\nabla^2\phi(x) \preceq \nabla^2 F(x) \preceq \nabla^2\phi(x). \tag{14}$$

Now we invoke a statistical argument based on the empirical average structure in (3). Without loss of generality, we assume that $\mathcal{D}_1$ contains the first n samples of $\{z_1, \ldots, z_N\}$ and thus $\nabla^2 f_1(x) = \frac{1}{n}\sum_{i=1}^n \nabla^2\ell(x, z_i)$. For any fixed x, we can use Hoeffding's inequality for matrices (Tropp, 2015) to obtain, with probability $1 - \delta$,

$$\left\|\frac{1}{n}\sum_{i=1}^n \nabla^2\ell(x, z_i) - \nabla^2 F(x)\right\| \le \sqrt{\frac{32 L_\ell^2 \log(d/\delta)}{n}}, \tag{15}$$

where L_ℓ is the uniform upper bound on $\|\nabla^2\ell(x, z_i)\|$.

If the losses $\ell(x, z_i)$ are *quadratic* in x, then the Hessians are constant and (13) holds with $\mu = \widetilde{O}(L_\ell/\sqrt{n})$, hiding the factor $\log(d/\delta)$. In this case, we derive from (14) that

$$\kappa_{F/\phi} = 1 + \frac{2\mu}{\sigma_F} = 1 + \widetilde{O}\left(\frac{\kappa_\ell}{\sqrt{n}}\right), \tag{16}$$

where we assume $\sigma_F \approx \sigma_\ell$, where $\nabla^2\ell(x, z_i) \succeq \sigma_\ell I_d$ for all x. Therefore, for large n, whenever we have $\kappa_{F/\phi} < \kappa_F$, the communication complexity $O(\kappa_{F/\phi}\log(1/\epsilon))$ is better than without preconditioning.

For non-quadratic loss functions, we need to ensure that (13) holds *uniformly* over a compact domain with high probability. Standard ball-packing arguments encounter an additional factor of $\sqrt{d}$ (e.g., Zhang and Xiao, 2018, Lemma 6). In this case, we have $\mu = \widetilde{O}(L_\ell\sqrt{d/n})$ and

$$\kappa_{F/\phi} = 1 + \frac{2\mu}{\sigma_F} = 1 + \widetilde{O}\left(\frac{\kappa_\ell\sqrt{d}}{\sqrt{n}}\right), \tag{17}$$

which suggests that the benefit of preconditioning may degrade or disappear in high dimension.

1.2. Contributions and Outline

In this paper, we make the following two contributions.

First, we propose a Statistically Preconditioned Accelerated Gradient (SPAG) method that can further reduce the communication complexity. Accelerated methods with $O(\sqrt{\kappa_{F/\phi}}\log(1/\epsilon))$ complexity have been developed for quadratic loss functions (see related works in Section 2). However, Dragomir et al. (2019) have shown that acceleration is not possible in general in the relatively smooth and strongly convex setting, and that more assumptions are needed. Here, by leveraging the fact the reference function ϕ itself is smooth and strongly convex, we obtain

$$\Phi(x_t) - \Phi(x_*) \le \prod_{\tau=1}^{t}\left(1 - \frac{1}{\sqrt{\kappa_{F/\phi}G_\tau}}\right) L_{F/\phi}D_\phi(x_*, x_0),$$

where $1 \le G_t \le \kappa_\phi$ and $G_t \to 1$ geometrically. Moreover, G_t can be calculated at each iteration and serve as numerical certificate of the actual convergence rate. In all of our experiments, we observe $G_t \approx 1$ even in early iterations, which results in $O(\sqrt{\kappa_{F/\phi}}\log(1/\epsilon))$ iterations empirically.

Second, we derive refined bounds on the relative condition number for linear prediction models. Linear models such as logistic regression have the form $\ell(x, z_i) = \ell_i(a_i^\top x) + (\lambda/2)\|x\|^2$. Assume that $\ell_i''(a_i^\top x) \le 1$ and $\|a_i\| \le R$ for all i, which implies $L_\ell = R^2$ and $\kappa_\ell = R^2/\lambda$. Then the Hoeffding bounds in (16) for quadratics becomes $\kappa_{F/\phi} = 1 + \widetilde{O}\left(\frac{R^2}{\sqrt{n}\lambda}\right)$, and for nonquadratics, the bound in (17) (from previous work) becomes $\kappa_{F/\phi} = 1 + \widetilde{O}\left(\frac{R^2\sqrt{d}}{\sqrt{n}\lambda}\right)$. We show that:

- For quadratic losses, the bound on relative condition number can be improved by a factor of $\sqrt{n}$, i.e.,

$$\kappa_{F/\phi} = \frac{3}{2} + O\left(\frac{R^2}{n\lambda}\log\left(\frac{d}{\delta}\right)\right).$$

- For non-quadratic losses, we derive a uniform concentration bound to remove the dependence of $\kappa_{F/\phi}$ on d,

$$\kappa_{F/\phi} = 1 + O\left(\frac{R^2}{\sqrt{n}\lambda}\left(RD + \sqrt{\log(1/\delta)}\right)\right),$$

  where D is the diameter of $\operatorname{dom}\phi$ (bounded domain). We also give a refined bound when the inputs a_i are sub-Gaussian.

These new bounds on $\kappa_{F/\phi}$ improve the convergence rates for all existing accelerated and non-accelerated preconditioned gradient methods (see related work in Section 2).

We start by discussing related work in Section 2. In Section 3, we introduce SPAG and give its convergence analysis. In Section 4, we derive sharp bounds on the relative condition number, and discuss their implications on the convergence rates of SPAG and other preconditioned gradient methods. We present experimental results in Section 5.

2. Related Work

Shamir et al. (2014) considered the case $\psi \equiv 0$ and introduced the statistical preconditioner (10) in DANE. Yet, they define a separate $\phi_j(x) = f_j(x) + (\mu/2)\|x\|^2$ for each worker j, compute m separate local updates using (11), and then use their average as x_{t+1}. For quadratic losses, they obtain the communication complexity $\widetilde{O}((\kappa_\ell^2/n)\log(1/\epsilon))$, which is roughly $O(\kappa_{F/\phi}^2\log(1/\epsilon))$ in our notation, which is much worse than their result without averaging of $O(\kappa_{F/\phi}\log(1/\epsilon))$ given in Section 1.1. We further improve this to $O(\sqrt{\kappa_{F/\phi}}\log(1/\epsilon))$ using acceleration.

Zhang and Xiao (2015) proposed DiSCO, an inexact damped Newton method, where the Newton steps are computed by a distributed conjugate gradient method with a similar preconditioner as (10). They obtain a communication complexity of $\widetilde{O}((\sqrt{\kappa_\ell}/n^{1/4})\log(1/\epsilon))$ for quadratic losses and $\widetilde{O}(\sqrt{\kappa_\ell}(d/n)^{1/4}\log(1/\epsilon))$ for self-concordant losses. Comparing with (16) and (17), in both cases they correspond to $O(\sqrt{\kappa_{F/\phi}}\log(1/\epsilon))$ in our notation. Reddi et al. (2016) use the Catalyst framework (Lin et al., 2015) to accelerate DANE; their method, called AIDE, achieves the same improved complexity for quadratic functions. We obtain similar results for smooth convex functions using direct acceleration.

Yuan and Li (2019) revisited the analysis of DANE and found that the worse complexity of $\widetilde{O}((\kappa_\ell^2/n)\log(1/\epsilon))$ is due to the lost statistical efficiency when averaging m different updates computed by (11). They propose to use a single local preconditioner at the server and obtain a communication complexity of $\widetilde{O}((1 + \kappa_\ell/\sqrt{n})\log(1/\epsilon))$ for quadratic functions. In addition, they propose a variant of DANE with heavy-ball momentum (DANE-HB), and show that it has communication complexity $\widetilde{O}((\sqrt{\kappa_\ell}/n^{1/4})\log(1/\epsilon))$ for quadratic loss functions, matching that of DiSCO and AIDE. For non-quadratic functions, they show DANE-HB has accelerated local convergence rate near the solution.

Wang et al. (2018) proposed GIANT, an approximate New-

ton method that approximates the overall Hessian by the harmonic mean of the local Hessians. It is equivalent to DANE in the quadratic case. They obtain a communication complexity that has logarithmic dependence on the condition number but requires local sample size $n > d$. Mahajan et al. (2018) proposed a distributed algorithm based on local function approximation, which is related to the preconditioning idea of DANE. Wang and Zhang (2019) apply statistical preconditioning to speed up a mini-batch variant of SVRG (Johnson and Zhang, 2013), but they rely on generic Catalyst acceleration and their convergence results only hold for a very small ball around the optimum.

Distributed optimization methods that use dual variables to coordinate solutions to local subproblems include ADMM (Boyd et al., 2010) and CoCoA (Jaggi et al., 2014; Ma et al., 2015; 2017). Numerical experiments demonstrate that they benefit from statistical similarities of local functions in the early iterations (Xiao et al., 2019), but their established communication complexity is no better than $O(\kappa_F \log(1/\epsilon))$.

3. The SPAG Algorithm

Although our main motivation in this paper is distributed optimization, the SPAG algorithm works in the general setting of minimizing relatively smooth and strongly convex functions. In this section, we first present SPAG in the more general setting (Algorithm 1), then explain how to run it for distributed empirical risk minimization.

In the general setting, we consider convex optimization problems of the form (1), where ψ is a closed convex function and F satisfies the following assumption.

Assumption 1. *F is L_F-smooth and σ_F-strongly convex. In addition, it is $L_{F/\phi}$-smooth and $\sigma_{F/\phi}$-strongly convex with respect to a differentiable convex function ϕ, and ϕ itself is L_ϕ-smooth and σ_ϕ-strongly convex.*

Algorithm 1 requires an initial point $x_0 \in \operatorname{dom}\psi$ and two parameters $L_{F/\phi}$ and $\sigma_{F/\phi}$. During each iteration, Line 6 finds $a_{t+1} > 0$ by solving a quadratic equation, then Line 7 calculates three scalars α_t, β_t and η_t, which are used in the later updates for the three vectors y_t, v_{t+1} and x_{t+1}. The function $V_t(\cdot)$ being minimized in Line 10 is defined as

$$V_t(x) = \eta_t\big(\nabla F(y_t)^\top x + \psi(x)\big) + (1-\beta_t)D_\phi(x, v_t) + \beta_t D_\phi(x, y_t). \quad (18)$$

The inequality that needs to be satisfied in Line 12 is

$$D_\phi(x_{t+1}, y_t) \le \alpha_t^2 G_t\Big((1-\beta_t)D_\phi(v_{t+1}, v_t) + \beta_t D_\phi(v_{t+1}, y_t)\Big), \quad (19)$$

where G_t is a scaling parameter depending on the properties of D_ϕ. It is a more flexible version of the *triangle scaling gain* introduced by Hanzely et al. (2018).

Algorithm 1 SPAG$(L_{F/\phi}, \sigma_{F/\phi}, x_0)$

1: $v_0 = x_0$, $A_0 = 0$, $B_0 = 1$, $G_{-1} = 1$
2: **for** $t = 0, 1, 2, \ldots$ **do**
3: $\quad G_t = \max\{1, G_{t-1}/2\}/2$
4: $\quad$ **repeat**
5: $\quad\quad G_t \leftarrow 2G_t$
6: $\quad\quad$ Find a_{t+1} such that $a_{t+1}^2 L_{F/\phi} G_t = A_{t+1}B_{t+1}$ where $A_{t+1} = A_t + a_{t+1}$, $B_{t+1} = B_t + a_{t+1}\sigma_{F/\phi}$
7: $\quad\quad \alpha_t = \frac{a_{t+1}}{A_{t+1}}$, $\beta_t = \frac{a_{t+1}}{B_{t+1}}\sigma_{F/\phi}$, $\eta_t = \frac{a_{t+1}}{B_{t+1}}$
8: $\quad\quad y_t = \frac{1}{1-\alpha_t\beta_t}\big((1-\alpha_t)x_t + \alpha_t(1-\beta_t)v_t\big)$
9: $\quad\quad$ Compute $\nabla F(y_t)$ *(requires communication)*
10: $\quad\quad v_{t+1} = \arg\min_x V_t(x)$
11: $\quad\quad x_{t+1} = (1-\alpha_t)x_t + \alpha_t v_{t+1}$
12: $\quad$ **until** Inequality (19) is satisfied
13: **end for**

As we will see in Theorem 1, smaller G_t's correspond to faster convergence rate. Algorithm 1 implements a *gain-search* procedure to automatically find a small G_t. At the beginning of each iteration, the algorithm always trys to set $G_t = G_{t-1}/2$ as long as $G_{t-1} \ge 2$ (G_{t-1} is divided by 4 in Line 3 since it is always multiplied by 2 in Line 5). Whenever (19) is not satisfied, G_t is multiplied by 2. When the inequality (19) is satisfied, G_t is within a factor of 2 from its smallest possible value. The following lemma guarantees that the gain-search loop always terminates within a small number of steps (see proof in Appendix A).

Lemma 1. *If Assumption 1 holds, then the inequality (19) holds with $G_t = \kappa_\phi = L_\phi/\sigma_\phi$.*

Therefore, if $\phi = (1/2)\|\cdot\|^2$, then we can set $G_t = 1$ and there is no need to check (19). In general, Algorithm 1 always produces $G_t < 2\kappa_\phi$ for all $t \ge 0$. Following the argument from Nesterov (2013, Lemma 4), the total number of gain-searches performed up to iteration t is bounded by

$$2(t+1) + \log_2(G_t),$$

which also bounds the total number of gradient evaluations. Thus the overhead is roughly twice as if there were no gain-search. Next we present a convergence theorem for SPAG.

Theorem 1. *Suppose Assumption 1 holds. Then the sequences generated by SPAG satisfy for all $t \ge 0$,*

$$\big(\Phi(x_t) - \Phi(x_*)\big) + \sigma_{F/\phi} D_\phi(x_*, v_t) \le \frac{1}{A_t} D_\phi(x_*, v_0),$$

where $A_t = \frac{1}{4\sigma_{F/\phi}}\Big(\prod_{\tau=0}^{t-1}(1+\gamma_\tau) - \prod_{\tau=0}^{t-1}(1-\gamma_\tau)\Big)^2$, *and* $\gamma_t = \frac{1}{2\sqrt{\kappa_{F/\phi} G_t}}$.

The proof of Theorem 1 relies on the techniques of Nesterov and Stich (2017), and the details are given in Appendix A. We can estimate the convergence rate as follows:

$$\frac{1}{A_t} = O\Big(\prod_{\tau=0}^{t}\Big(1-\frac{1}{\sqrt{\kappa_{F/\phi}G_\tau}}\Big)\Big) = O\bigg(\Big(1-\frac{1}{\sqrt{\kappa_{F/\phi}\widetilde{G}_t}}\Big)^t\bigg),$$

where $\widetilde{G}_t$ is such that $\widetilde{G}_t^{-1/2} = (1/t)\sum_{\tau=0}^{t} G_t^{-1/2}$, that is, $\widetilde{G}_t^{1/2}$ is the harmonic mean of $G_0^{1/2},\dots,G_{t-1}^{1/2}$. In addition, it can be shown that $A_t \geq t^2/(4L_{F/\phi}\widetilde{G}_t)$. Therefore, as $\sigma_{F/\phi}\to 0$, Theorem 1 gives an accelerated sublinear rate:

$$\Phi(x_t) - \Phi(x_*) \leq \frac{4L_{F/\phi}\widetilde{G_t}}{t^2} D_\phi(x_*, x_0).$$

To estimate the worst case when $\sigma_{F/\phi} > 0$, we replace G_t by κ_ϕ to obtain the iteration complexity $O\big(\sqrt{\kappa_{F/\phi}\kappa_\phi}\log(1/\epsilon)\big)$. Since $\kappa_{F/\phi}\kappa_\phi \approx \kappa_F$, this is roughly $O\big(\sqrt{\kappa_F}\log(1/\epsilon)\big)$, the same as without preconditioning. However, the next lemma shows that under a mild condition, we always have $G_t \to 1$ geometrically.

Lemma 2. *Suppose Assumption 1 holds and in addition, $\nabla^2\phi$ is M-Lipschitz-continuous, i.e., for all $x, y \in \operatorname{dom}\psi$,*

$$\big\|\nabla^2\phi(x) - \nabla^2\phi(y)\big\| \leq M\|x-y\|.$$

Then the inequality (19) holds with

$$G_t = \min\big\{\kappa_\phi,\ 1 + (M/\sigma_\phi)d_t\big\}, \tag{20}$$

where $d_t = \|v_{t+1}-v_t\| + \|v_{t+1}-y_t\| + \|x_{t+1}-y_t\|$.

In particular, if ϕ is quadratic, then we have $M = 0$ and $G_t = 1$ always satisfies (19). In this case, the convergence rate in Theorem 1 satisfies $1/A_t = O\big(\big(1-1/\sqrt{\kappa_{F/\phi}}\big)^t\big)$.

In general, $M \neq 0$, but it can be shown that the sequences generated by Algorithm 1, $\{x_t\}$, $\{y_t\}$ and $\{v_t\}$ all converge to x_* at the rate $\big(1-1/\sqrt{\kappa_F}\big)^t$ (see, e.g., Lin and Xiao, 2015, Theorem 1). As a result, $d_t \to 0$ and thus $G_t \to 1$ at the same rate. Consequently, the convergence rate established in Theorem 1 quickly approaches $O\big(\big(1-1/\sqrt{\kappa_{F/\phi}}\big)^t\big)$.

The asymptotic nature of the preconditioned convergence rate in the nonquadratic case (G_t converges to 1 instead of being a small constant) seems to be unavoidable, given the recent work on lower bounds for mirror descent methods by Dragomir et al. (2019).

3.1. Implementation for Distributed Optimization

In distributed optimization, Algorithm 1 is implemented at the server. During each iteration, communication between the server and the workers only happens when computing $\nabla F(y_t)$. Checking if the inequality (19) holds locally requires that the server has access to the preconditioner ϕ.

If the datasets on different workers are i.i.d. samples from the same source distribution, then we can use any f_j in the definition of ϕ in (10) and assign worker j as the server. However, this is often not the case in practice and obtaining i.i.d. datasets on different workers may involve expensive shuffling and exchanging large amount of data among the workers. In this case, a better alternative is to randomly sample small portions of the data on each worker and send them to a dedicated server. We call this sub-sampled dataset $\mathcal{D}_0$ and the local loss at the server f_0, which is defined the same way as in (3). Then the server implements Algorithm 1 with $\phi(x) = f_0(x) + (\mu/2)\|x\|^2$. Here we only need $\mathcal{D}_0$ be a uniform sub-sample of $\cup_{j=1}^m \mathcal{D}_j$, which is critical for effective preconditioning. On the other hand, it is not a problem at all if the datasets at the workers, $\mathcal{D}_1,\dots,\mathcal{D}_m$, are not shuffled to to be i.i.d., because it does not change the average gradients $\nabla F(y_t)$. In the rest of the paper, we omit the subscript to simply use f to represent the local empirical loss function. As discussed in Section 1.1, if

$$\big\|\nabla^2 f(x) - \nabla^2 F(x)\big\| \leq \mu, \quad \forall\, x \in \operatorname{dom}\psi \tag{21}$$

with high probability, then according to (14), we can choose

$$L_{F/\phi} = 1, \qquad \sigma_{F/\phi} = \frac{\sigma_F}{\sigma_F + 2\mu}$$

as the input to Algorithm 1. In the next section, we leverage matrix concentration bounds to estimate how μ varies with the number of subsamples n. With sufficiently large n, we can make μ small so that the relative condition number $\kappa_{F/\phi} = 1 + 2\mu/\sigma_F$ is much smaller than κ_F.

4. Bounding the Relative Condition Number

In this section, we derive refined matrix concentration bounds for linear prediction models. Suppose the overall dataset consists of N samples $\{z_1,\dots,z_N\}$, where each $z_i = (a_i, b_i)$ with $a_i \in \mathbb{R}^d$ being a feature vector and b_i the corresponding label or regression target. Linear models (including logistic and ridge regression) have the form $\ell(x, z_i) = \ell_i(a_i^\top x) + \frac{\lambda}{2}\|x\|^2$, where ℓ_i is twice differentiable and may depend on b_i, and $\lambda > 0$. We further assume that $\ell_i'' = \ell_j''$ for all i and j, which is valid for logistic and ridge regression as well. Since $f(x) = (1/n)\sum_{i=1}^n \ell(x, z_i)$, we have

$$\nabla^2 f(x) = \frac{1}{n}\sum_{i=1}^{n} \ell_i''(a_i^\top x) a_i a_i^\top + \lambda I_d. \tag{22}$$

Here we omit the subscript j in f_j since we only need one subsampled dataset at the server, as explained in Section 3.1. For the overall loss function defined in (2), the Hessian $\nabla^2 F(x)$ is defined similarly by replacing n with N.

We assume for simplicity that the strong convexity of F mainly comes from regularization, that is, $\sigma_F = \sigma_\ell = \lambda$, but

the results can be easily extended to account for the strong convexity from data. We start by showing tight results for quadratics, and then provide uniform concentration bounds of Hessians for more general loss functions. Finally, we give a refined bound when the a_i's are sub-Gaussian.

4.1. Quadratic Case

We assume in this section that $\ell_i(a_i^\top x) = (a_i^\top x - b_i)^2/2$, and that there exists a constant R such that $\|a_i\| \leq R$ for all $i = 1, \dots, N$. In this case we have $L_\ell = R^2$ and $\kappa_\ell = R^2/\lambda$. Since the Hessians do not depend on x, we use the notation

$$H_F = \nabla^2 F(x), \quad H_f = \nabla^2 f(x).$$

Previous works (Shamir et al., 2014; Reddi et al., 2016; Yuan and Li, 2019) use the Hoeffding bound (15) to obtain

$$\left(1 + \frac{2\mu}{\lambda}\right)^{-1} (H_f + \mu I_d) \preceq H_F \preceq H_f + \mu I_d, \quad (23)$$

$$\text{with} \qquad \mu = \frac{R^2}{\sqrt{n}} \sqrt{32 \log(d/\delta)}. \quad (24)$$

Our result is given in the following theorem.

Theorem 2. *Suppose ℓ_i is quadratic and $\|a_i\| \leq R$ for all i. For a fixed $\delta > 0$, if $n > \frac{28}{3} \log\left(\frac{2d}{\delta}\right)$, then the following inequality holds with probability at least $1 - \delta$:*

$$\left(\frac{3}{2} + \frac{2\mu}{\lambda}\right)^{-1} (H_f + \mu I_d) \preceq H_F \preceq 2\,(H_f + \mu I_d)\,, \quad (25)$$

$$\textit{with} \qquad \mu = \frac{1}{2}\left(\frac{28R^2}{3n} \log\left(\frac{2d}{\delta}\right) - \lambda\right)^+. \quad (26)$$

Thus, for this choice of μ, $\sigma_{F/\phi} = \left(\frac{3}{2} + \frac{2\mu}{\lambda}\right)^{-1}$, $L_{F/\phi} = 2$ and so $\kappa_{F/\phi} = O\left(1 + \frac{\kappa_\ell}{n} \log\left(\frac{d}{\delta}\right)\right)$ with probability $1 - \delta$.

Theorem 2 improves on the result in (24) by a factor of $\sqrt{n}$. The reason is that matrix inequality (23) is derived from the additive bound $\|H_f - H_F\| \leq \mu$ (e.g., Shamir et al., 2014; Yuan and Li, 2019). We derive the matrix inequality (25) directly from a multiplicative bound using the matrix Bernstein inequality (see proof in Appendix B.1). Note that by using matrix Bernstein instead of matrix Hoeffding inequality (Tropp, 2015), one can refine the bound for μ in (23) from $L_\ell/\sqrt{n}$ to $\sqrt{L_\ell L_F/n}$, which can be as small as L_ℓ/n in the extreme case when all the a_i's are orthogonal. Our bound in (26) states that $\mu = \widetilde{O}(L_\ell/n)$ in general for quadratic problems, leading to $\kappa_{F/\phi} = \widetilde{O}(1 + \kappa_\ell/n)$.

Remark 1. Theorem 2 is proved by assuming random sampling with replacement. In practice, we mostly use random sampling without replacement, which usually concentrates even more than with replacement (Hoeffding, 1963).

Remark 2. In terms of reducing $\kappa_{F/\phi}$, there is not much benefit to having $\mu < \lambda$. Indeed, higher values of μ regularize the inner problem of minimizing $V_t(x)$ in (18), because the condition number of $D_\phi(x, y) = D_f(x, y) + (\mu/2)\|x - y\|^2$ is $(L_f + \mu)/(\lambda + \mu)$. Increasing μ can thus lead to substantially easier subproblems when $\mu > \lambda$, which reduces the computation cost at the server, although this may sometimes affect the rate of convergence.

4.2. Non-quadratic Case

For non-quadratic loss functions, we need $\nabla^2 f(x)$ to be a good approximation of $\nabla^2 F(x)$ for all iterations of the SPAG algorithm. It is tempting to argue that concentration only needs to hold for the iterates of SPAG, and a union bound would then give an extra $\log T$ factors for T iterations. Yet this only works for one step since x_t depends on the points chosen to build f for $t > 0$, so the $\ell''(a_i^\top x_t) a_i a_i^\top$ are not independent for different i (because of x_t). Therefore, the concentration bounds need to be written at points that do not depend on f. In order to achieve this, we restrict the optimization variable within a bounded convex set and prove uniform concentration of Hessians over the set. Without loss of generality, we consider optimization problems constrained in $\mathcal{B}(0, D)$, the ball of radius D centered at 0. Correspondingly, we set the nonsmooth regularization function as $\psi(x) = 0$ if $x \in \mathcal{B}(0, D)$ and infinity otherwise.

If the radius D is small, it is then possible to leverage the quadratic bound by using the inequality

$$\begin{aligned}\|H_f(x) - H_F(x)\| \;\leq\; &\|H_f(x) - H_f(y)\| \\ &+ \|H_f(y) - H_F(y)\| + \|H_F(x) - H_F(y)\|.\end{aligned}$$

Thus, under a Lipschitz-continuous Hessian assumption (which we have), only concentration at point y matters. Yet, such bounding is only meaningful when x is close to y, thus leading to the very small convergence radius of Wang and Zhang (2019, Theorem 13), in which they use concentration at the optimal point x_*. Using this argument for several y's that pave $\mathcal{B}(0, D)$ leads to an extra $\sqrt{d}$ multiplicative factor since concentration needs to hold at exponentially (in d) many points, as discussed in Section 1.1. We take a different approach in this work, and proceed by directly bounding the supremum for all $x \in \mathcal{B}(0, D)$, thus looking for the smallest μ that satisfies:

$$\sup_{x \in \mathcal{B}(0,D)} \|H_f(x) - H_F(x)\|_{\text{op}} \leq \mu. \quad (27)$$

Equation (23) can then be used with this specific μ. We now introduce Assumption 2, which is for example verified for logistic regression with $B_\ell = 1/4$ and $M_\ell = 1$.

Assumption 2. *There exist B_ℓ and M_ℓ such that ℓ_i'' is M_ℓ-Lipschitz continuous and $0 \leq \ell_i''(a^\top x) \leq B_\ell$ almost surely for all $x \in \mathcal{B}(0, D)$.*

Theorem 3. *If ℓ_i satisfies Assumption 2, then Equation (27) is satisfied with probability at least $1-\delta$ for*

$$\mu = \sqrt{4\pi}\frac{R^2}{\sqrt{n}}\left(B_\ell\left[2+\sqrt{\frac{1}{2\pi}\log(\delta^{-1})}\right]+RM_\ell D\right).$$

Sketch of proof. The high probability bound on the supremum is obtained using Mc Diarmid inequality (Boucheron et al., 2013). This requires a bound on its expectation, which is obtained using symmetrization and the Sudakov-Fernique Lemma (Boucheron et al., 2013). The complete proof can be found in Appendix B.2. □

The bound of Theorem 3 is relatively tight as long as $RM_\ell D < B_\ell\sqrt{\log(\delta^{-1})}$. Indeed, using the matrix Bernstein inequality for a fixed $x \in \mathcal{B}(0,D)$ would yield $\mu = O\left(R\sqrt{L_F}B_\ell\log(d/\delta)/\sqrt{n}\right)$. Therefore, Theorem 3 is tight up to a factor $R/\sqrt{L_F}$ in this case.

Remark 3. We consider D to be fixed in this work, although obtaining a meaningful solution to the ERM problem may require D to depend on the dimension d. Yet, D would actually depend on the intrinsic dimension of the data, which can be much smaller than d, especially when features are sparse.

4.3. Sub-Gaussian Bound

We show in this section that the bound of Theorem 3 can be improved under a stronger sub-Gaussian assumption on a.

Definition 2. *The random variable $a \in \mathbb{R}^d$ is sub-Gaussian with parameter $\rho > 0$ if one has for all $\epsilon > 0$, $x \in \mathcal{B}(0,D)$:*

$$\mathbb{P}(|a_i^\top x)| \geq \rho\epsilon) \leq 2e^{-\frac{\epsilon^2}{2\|x\|^2}}. \tag{28}$$

Theorem 4. *If ℓ_i satisfies Assumption 2 and the a_i are sub-Gaussian with constant ρ, then denoting $\tilde{B} = B_\ell/(M_\ell D)$, there exists $C > 0$ such that Equation (27) is satisfied with probability $1-\delta$ for*

$$\mu = C\frac{\rho^2 M_\ell D}{\sqrt{n}}(d+\log(\delta^{-1}))\left[\frac{\rho+\tilde{B}}{\sqrt{d}}+\frac{\rho+(R^2\tilde{B})^{\frac{1}{3}}}{\sqrt{n}}\right].$$

Recall that this value of μ can be plugged into Equation (23) to bound the relative condition number.

Sketch of proof. This bound is a specific instantiation of a more general result based on chaining, which is a standard argument for proving results on suprema of empirical processes (Boucheron et al., 2013). The complete proof can be found in Appendix B.3. □

The sub-Gaussian assumption (28) always holds with $\rho = R$, the almost sure bound on $\|a_i\|$. However Theorem 4 improves over Theorem 3 only with a stronger sub-Gaussian assumption, i.e., when $\rho < R$. In particular for a_i uniform over $\mathcal{B}(0,R)$, one has $\rho = R/\sqrt{d}$. Assuming further that the $(R^2B)^{1/3}/\sqrt{n}$ term dominates yields $\mu = O(R^2(R^2B)^{1/3}/n)$, a $\sqrt{n}$ improvement over Theorem 3. We expect tighter versions of Theorem 4, involving the effective dimension d_{eff} of vectors a_i instead of the full dimension d, to hold.

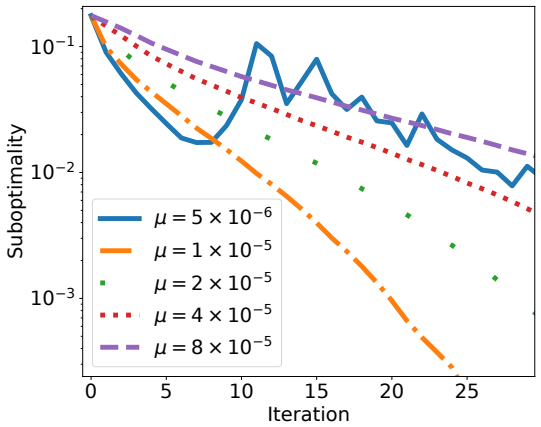


Figure 1. Effect of μ on the convergence speed of SPAG on RCV1 with $\lambda = 10^{-7}$ and $n = 10^4$.

5. Experiments

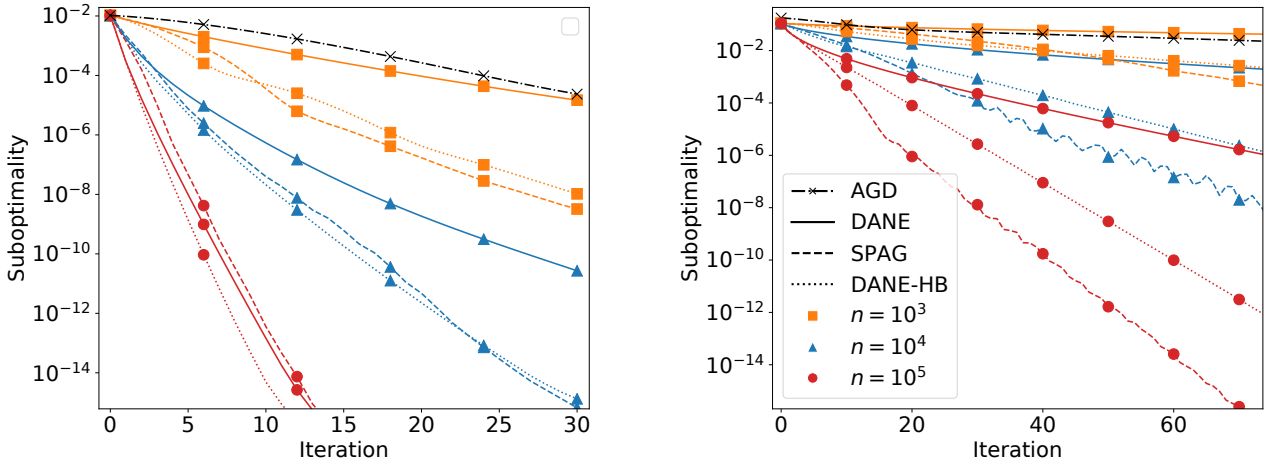
We have seen in the previous section that preconditioned gradient methods can outperform gradient descent by a large margin in terms of communication rounds, which was already observed empirically (Shamir et al., 2014; Reddi et al., 2016; Yuan and Li, 2019). We compare in this section the performances of SPAG with those of DANE and its heavy-ball acceleration, HB-DANE (Yuan and Li, 2019), as well as accelerated gradient descent (AGD). Due to its better convergence guarantees (Shamir et al., 2014; Yuan and Li, 2019), DANE refers in this section to the proximal gradient method with the Bregman divergence associated to $\phi = f_1 + (\mu/2)\|\cdot\|^2$ (without averaging over m workers).

We apply these algorithms to train linear prediction models over a dataset $\{(a_i, b_i)\}_{i=1}^N$, where each $a_i \in \mathbb{R}^d$ is a feature vector and b_i is the corresponding label or regression target. Specifically, we solve the problem

$$\underset{x\in\mathbb{R}^d}{\text{minimize}} \quad F(x) = \frac{1}{N}\sum_{i=1}^n \ell_i(a_i^\top x) + \frac{\lambda}{2}\|x\|^2,$$

where $\ell_i(a_i^\top x) = \log(1+\exp(-b_i(a_i^\top x)))$ for logistic regression with $b_i \in \{-1,+1\}$ and $\ell_i(a_i^\top x) = (a_i^\top x - b_i)^2$ for ridge regression with $b_i \in \mathbb{R}$. We use two datasets from LibSVM[1], RCV1 (Lewis et al., 2004) and the preprocessed version of KDD2010 (algebra) (Yu et al., 2010).

[1]Accessible at https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html



(a) Regularization $\lambda = 10^{-5}$

(b) Regularization $\lambda = 10^{-7}$

Figure 2. Logistic regression on RCV1 dataset. The legend is the same for all figures. We use $\mu = 0.1/n$, except for $n = 10^5$ for which we use $\mu = 2 \cdot 10^{-6}$.

Note that, as mentioned in Section 3.1, the number of nodes used by SPAG does not affect its iteration complexity (but change the parallelism of computing $\nabla F(x_t)$). Only the size n of the dataset used for preconditioning matters. We initialize all algorithms at the same point, which is the minimizer of the server's entire local loss (with 10^5 samples regardless of how many samples are used for preconditioning). Note that this is not mandatory but provides a good communication-free initialization, although it requires partially solving the server's local loss. Not performing this would simply yield a worse initialization.

Tuning μ. Although μ can be estimated using concentrations results, as done in Section 4, these bounds are too loose to be used in practice. Yet, they show that μ depends very weakly on λ. This is verified experimentally, and we therefore use the same value for μ regardless of λ. To test the impact of μ on the iteration complexity, we fix a step-size of 1 and plot the convergence speed of SPAG for several values of μ. We see on Figure 1 that the value of μ drastically affects convergence, actually playing a role similar to the inverse of a step-size. Indeed, the smaller the μ the faster the convergence, up to a point at which the algorithm is not stable anymore. Convergence could be obtained for smaller values of μ by taking a smaller step-size. Yet, the step-size needs to be tuned for each value of μ, and we observed that this does not lead to significant improvements in practice. Thus, unless explicitly stated, we stick to the guidelines for DANE by Shamir et al. (2014), i.e., we choose $L_{F/\phi} = 1$ and tune μ. The tuning strategy for μ is the following: we tune the base value of μ by starting from $0.1/n$ for the smallest n and then decreasing it as long as it is stable, or increasing it as long as it is unstable. Then, we keep this base value and obtain μ for other values of n by relying on the fact that μ should be proportional to $1/n$ (slightly adjusting when necessary if the algorithm becomes unstable). Therefore, even though some tuning is required to set μ, this tuning is guided by the insight provided by Section 4.

Setting acceleration parameters. SPAG and HB-DANE require additional parameters compared with DANE. Yet, we use in our experiments the values given by the theory, *i.e.*, we use SPAG with $\sigma_{F/\phi}^{-1} = 1 + 2\mu/\lambda$ and HB-DANE with $\beta = (1 - (1 + 2\mu/\lambda)^{-1/2})^2$. Fine tuning these parameters only leads to small improvements for both algorithms, as described in Appendix C. Therefore, SPAG and HB-DANE do not require more parameter tuning than DANE. We tune both the learning rate and the momentum of AGD.

Line search for G_t. As explained in Section 3, the optimal G_t is obtained through a line search. Yet, we observed in all our experiments that $G_t = 1$ most of the time. This is due to the fact that we start at the minimizer of the local cost function, which can be close to the global solution. In addition, Equation (20) can actually be verified for $G_t < 1$, even in the quadratic. Therefore, the line search generally has no added cost (apart from checking that $G_t = 1$ works) and the effective rate in our experiments is $\kappa_{F/\phi}^{-1/2}$. Experiments for Figures 2 and 4 use $G_t = 1$ for simplicity.

Local subproblems. Local problems are solved using a sparse implementation of dual-free SDCA (Shalev-Shwartz, 2016). In practice, the ill-conditionned regime is very hard, especially when μ is small. Indeed, the local subproblems are very hard to solve, and it should be beneficial to use accelerated algorithms to solve the inner problems. In our experiments, we warm-start the local problems (initializing

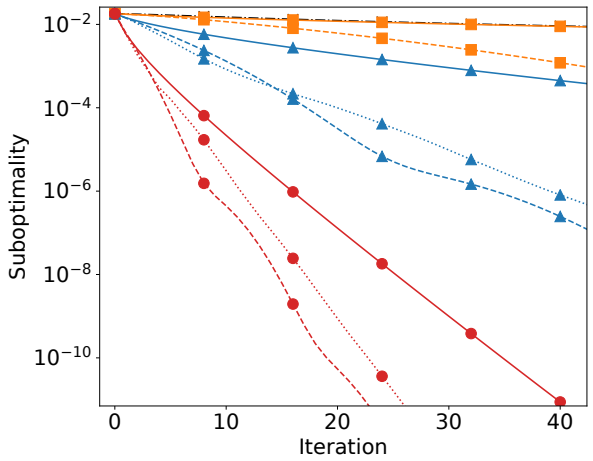


Figure 3. Ridge regression on RCV1 dataset with $\lambda = 10^{-5}$. The legend is the same as in Figure 4. We use $\mu = 4/n$, and $L_{F/\phi} = 1$, except for SPAG with $n = 10^5$ for which we use $L_{F/\phi}^{-1} = 0.9$ as it proved to be more stable.

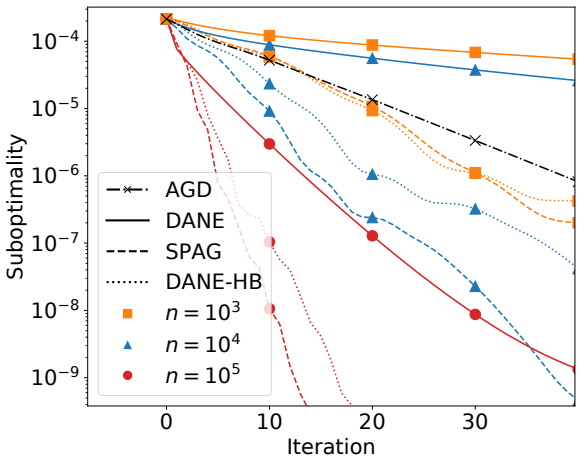


Figure 4. Logistic regression on KDD2010 with $\lambda = 10^{-7}$. We use $\mu = 0.1/(2n)$, except for $n = 10^3$ where $\mu = 10^{-5}$, and $L_{F/\phi} = 2$ for SPAG.

on the solution of the previous one), and keep doing passes over the preconditioning dataset until $\|\nabla V_t(x_t)\| \leq 10^{-9}$ (checked at each epoch).

RCV1. Figure 2 presents results for logistic regression on the RCV1 dataset with different regularization weights. All algorithms are run with $N = 677399$ (split over 4 nodes) and $d = 47236$. We see that in Figure 2(a), the curves can be clustered by values of n, meaning that when regularization is relatively high ($\lambda = 10^{-5}$), increasing the preconditioning sample size has a greater effect than acceleration since the problem is already well-conditioned. In particular, acceleration does not improve the convergence rate when $n = 10^5$ and $\lambda = 10^{-5}$. When regularization is smaller ($\lambda = 10^{-7}$), SPAG and HB-DANE outperform DANE even when ten times less samples are used for preconditioning, as shown in Figure 2(b). As discussed in Appendix C, finer tuning (without using the theoretical parameters) of the momentum marginally improves the performances of SPAG and HB-DANE, at the cost of a grid search. SPAG generally outperforms HB-DANE in our experiments, but both methods have comparable asymptotic rates.

Figure 3 presents results for ridge regression on the same RCV1 dataset, using the class labels as the regression targets. In this case, ϕ is quadratic so we do not need line search on G_t and always set $G_t = 1$. The results are very similar to the logistic regression case.

KDD2010. Figure 4 presents the results of larger scale experiments on a random subset of the KDD2010 dataset with $N = 7557074$ (split over 80 nodes), $d = 20216830$ and $\lambda = 10^{-7}$. The conclusions are similar to the experiments on RCV1, i.e., acceleration allows to use significantly less samples at the server for a given convergence speed. AGD competes with DANE when λ and n are small, but it is outperformed by SPAG in all our experiments. More experiments investigating the impact of line search, tuning and inaccurate local solutions are presented in Appendix C.

6. Conclusion

We have introduced SPAG, an accelerated algorithm that performs statistical preconditioning for large-scale distributed optimization. Although our motivation in this paper is distributed empirical risk minimization, SPAG applies to much more general settings that can benefit from statistical preconditioning. We have given tight bounds on the relative condition number, a crucial quantity to understand the convergence rate of preconditioned algorithms. We have also shown, both in theory and in experiments, that acceleration allows SPAG to efficiently leverage rough preconditioning when only limited number of local samples are available. Preliminary experiments suggest that SPAG is more robust to inaccurate solution of the inner problems than HB-DANE. Characterizing the effects of inaccurate inner solutions in the preconditioning setting would be an interesting extension of this work.

Acknowledgements

This work was funded in part by the French government under management of Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute). We also acknowledge support from the European Research Council (grant SEQUOIA 724063) and from the MSR-INRIA joint centre.

References

Yossi Arjevani and Ohad Shamir. Communication complexity of distributed convex learning and optimization. In *Advances in Neural Information Processing Systems 28*, pages 1756–1764, 2015.

Heinz H. Bauschke, Jérôme Bolte, and Marc Teboulle. A descent lemma beyond Lipschitz gradient continuity: first-order methods revisited and applications. *Mathematics of Operations Research*, 42(2):330–348, 2017.

Amir Beck. *First-Order Methods in Optimization*. MOS-SIAM Series on Optimization. SIAM, 2017.

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

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 2013.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3 (1):1–122, 2010.

Gong Chen and Marc Teboulle. Convergence analysis of a proximal-like minimization algorithm using Bregman functions. *SIAM Journal on Optimization*, 3(3):538–543, August 1993.

Radu-Alexandru Dragomir, Adrien Taylor, Alexandre d'Aspremont, and Jérôme Bolte. Optimal complexity and certification of Bregman first-order methods. *arXiv preprint arXiv:1911.08510*, 2019.

Filip Hanzely, Peter Richtarik, and Lin Xiao. Accelerated Bregman proximal gradient methods for relatively smooth convex optimization. arXiv:1808.03045, 2018.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

Martin Jaggi, Virginia Smith, Martin Takac, Jonathan Terhorst, Sanjay Krishnan, Thomas Hofmann, and Michael I. Jordan. Communication-efficient distributed dual coordinate ascent. In *Advances in Neural Information Processing Systems 27*, pages 3068–3076, 2014.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems 26*, pages 315–323, 2013.

David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. RCV1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5 (Apr):361–397, 2004.

Hongzhou Lin, Julien Mairal, and Zaid Harchaoui. A universal catalyst for first-order optimization. In *Advances in Neural Information Processing Systems*, pages 3384–3392, 2015.

Qihang Lin and Lin Xiao. An adaptive accelerated proximal gradient method and its homotopy continuation for sparse optimization. *Computational Optimization and Applications*, 60(3):633–674, Apr 2015.

Haihao Lu, Robert M Freund, and Yurii Nesterov. Relatively smooth convex optimization by first-order methods, and applications. *SIAM Journal on Optimization*, 28(1):333–354, 2018.

Chenxin Ma, Virginia Smith, Martin Jaggi, Michael I. Jordan, Peter Richtárik, and Martin Takáč. Adding vs. averaging in distributed primal-dual optimization. In *Proceedings of the International Conference on Machine Learning*, pages 1973–1982, 2015.

Chenxin Ma, Virginia Smith, Martin Jaggi, Michael I. Jordan, Peter Richtárik, and Martin Takáč. Distributed optimization with arbitrary local solvers. *Optimization Methods and Software*, 32(4):813–848, 2017.

Dhruv Mahajan, Nikunj Agrawal, S. Sathiya Keerthi, Sundararajan Sellamanickam, and Leon Bottou. An efficient distributed learning algorithm based on effective local functional approximations. *Journal of Machine Learning Research*, 19(74):1–37, 2018.

Yurii Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer, Boston, 2004.

Yurii Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming, Ser. B*, 140:125–161, 2013.

Yurii Nesterov and Sebastian U Stich. Efficiency of the accelerated coordinate descent method on structured optimization problems. *SIAM Journal on Optimization*, 27 (1):110–123, 2017.

Sashank J. Reddi, Jakub Konečnỳ, Peter Richtárik, Barnabás Póczós, and Alex Smola. AIDE: Fast and communication efficient distributed optimization. *arXiv preprint arXiv:1608.06879*, 2016.

R. T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

Kevin Scaman, Francis Bach, Sébastien Bubeck, Yin Tat Lee, and Laurent Massoulié. Optimal algorithms for smooth and strongly convex distributed optimization in networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 3027–3036, 2017.

Shai Shalev-Shwartz. Sdca without duality, regularization, and individual convexity. In *International Conference on Machine Learning*, pages 747–754, 2016.

Ohad Shamir, Nati Srebro, and Tong Zhang. Communication-efficient distributed optimization using an approximate Newton-type method. In *International Conference on Machine Learning*, pages 1000–1008, 2014.

Joel A. Tropp. An introduction to matrix concentration inequalities. *Foundations and Trends in Machine Learning*, 8(1-2):1–230, 2015.

Roman Vershynin. *High-Dimensional Probability, An Introduction with Applications in Data Science*. Cambridge University Press, 2019.

Jialei Wang and Tong Zhang. Utilizing second order information in minibatch stochastic variance reduced proximal iterations. *Journal of Machine Learning Research*, 20 (42):1–56, 2019.

Shusen Wang, Farbod Roosta-Khorasani, Peng Xu, and Michael W Mahoney. GIANT: Globally improved approximate Newton method for distributed optimization. In *Advances in Neural Information Processing Systems*, pages 2332–2342, 2018.

Lin Xiao, Adams Wei Yu, Qihang Lin, and Weizhu Chen. DSCOVR: Randomized primal-dual block coordinate algorithms for asynchronous distributed optimization. *Journal of Machine Learning Research*, 20(43):1–58, 2019.

Hsiang-Fu Yu, Hung-Yi Lo, Hsun-Ping Hsieh, Jing-Kai Lou, Todd G McKenzie, Jung-Wei Chou, Po-Han Chung, Chia-Hua Ho, Chun-Fu Chang, Yin-Hsuan Wei, et al. Feature engineering and classifier ensemble for KDD cup 2010. In *KDD Cup*, 2010.

Xiao-Tong Yuan and Ping Li. On convergence of distributed approximate Newton methods: Globalization, sharper bounds and beyond. *arXiv preprint arXiv:1908.02246*, 2019.

Yuchen Zhang and Lin Xiao. DiSCO: Distributed optimization for self-concordant empirical loss. In *International Conference on Machine Learning*, pages 362–370, 2015.

Yuchen Zhang and Lin Xiao. Communication-efficient distributed optimization of self-concordant empirical loss. In *Large-Scale and Distributed Optimization*, number 2227 in Lecture Notes in Mathematics, chapter 11, pages 289–341. Springer, 2018.

Appendix

A. Convergence Analysis of SPAG

This section provides proofs for Lemma 1, Theorem 1 and Lemma 2 presented in Section 3. Before getting to the proofs, we first comment on the nature of the accelerated convergence rate obtained in Theorem 1.

Note that SPAG (Algorithm 1) can be considered as an accelerated variant of the general mirror descent method considered by Bauschke et al. (2017) and Lu et al. (2018). Specifically, we can replace D_ϕ by the Bregman divergence of any convex function of Legendre type (Rockafellar, 1970, Section 26). Recently, Dragomir et al. (2019) show that fully accelerated convergence rates, as those for Euclidean mirror-maps achieved by Nesterov (2004), may not be attainable in the general setting. However, this negative result does not prevent us from obtaining better accelerated rates in the preconditioned setting. Indeed, we choose a smooth and strongly convex mirror map and further assume Lipschitz continuity of its Hessian. For smooth and strongly convex cost functions, the convergence rates of SPAG are almost always better than those obtained by standard accelerated algorithms (without preconditioning) as long as n is not too small, and can be much better with a good preconditioner.

A.1. Proof of Lemma 1

Using the second-order Taylor expansion (mean-value theorem), we have

$$D_\phi(x,y) = \phi(x) - \phi(y) - \langle \nabla\phi(y), x-y\rangle = \frac{1}{2}(x-y)^\top \nabla^2\phi\big(y+t(x-y)\big)(x-y),$$

for some scalar $t \in [0,1]$. We define

$$H(x,y) = \nabla^2\phi\big(y+t(x-y)\big),$$

where the dependence on $t \in [0,1]$ is made implicit with the ordered pair (x,y). Then we can write

$$D_\phi(x,y) = \frac{1}{2}\|x-y\|^2_{H(x,y)}.$$

By Assumption 1, ϕ is L_ϕ-smooth and σ_ϕ-strongly convex, which implies that for all $x, y \in \mathbb{R}^d$,

$$\sigma_\phi\|x-y\|^2 \le \|x-y\|^2_{H(x,y)} \le L_\phi\|x-y\|^2.$$

Let $w_t = (1-\beta_t)v_t + \beta_t y_t$. Then we have $x_{t+1} - y_t = \alpha_t\big(v_{t+1} - w_t\big)$ and

$$\begin{aligned} D_\phi(x_{t+1}, y_t) &= \frac{1}{2}\|x_{t+1}-y_t\|^2_{H(x_{t+1},y_t)} \\ &\le \frac{L_\phi}{2}\|x_{t+1}-y_t\|^2 = \alpha_t^2\frac{L_\phi}{2}\|v_{t+1}-w_t\|^2. \end{aligned}$$

Next we use $v_{t+1} - w_t = (1-\beta_t)(v_{t+1}-v_t) + \beta_t(v_{t+1}-y_t)$ and convexity of $\|\cdot\|^2$ to obtain

$$\begin{aligned} D_\phi(x_{t+1}, y_t) &\le \alpha_t^2\frac{L_\phi}{2}\left((1-\beta_t)\|v_{t+1}-v_t\|^2 + \beta_t\|v_{t+1}-y_t\|^2\right) \\ &\le \alpha_t^2\frac{L_\phi}{2\sigma_\phi}\left((1-\beta_t)\|v_{t+1}-v_t\|^2_{H(v_{t+1},v_t)} + \beta_t\|v_{t+1}-y_t\|^2_{H(v_{t+1},y_t)}\right) \\ &= \alpha_t^2\kappa_\phi\big((1-\beta_t)D_\phi(v_{t+1},v_t) + \beta_t D_\phi(v_{t+1},y_t)\big). \end{aligned}$$

This finishes the proof of Lemma 1.

A.2. Proof of Theorem 1

Theorem 2 is a direct consequence of the following result, which is adapted from Nesterov and Stich (2017).

Theorem 5 (Smooth and strongly convex mirror map ϕ)**.** *Suppose Assumption 1 holds. Then the sequences generated by Algorithm 1 satisfy for all $t \geq 0$,*

$$A_t\big(\Phi(x_t) - \Phi(x_*)\big) + B_t D(x_*, v_t) \leq A_0\big(F(x_0) - F(x_*)\big) + B_0 D(x_*, v_0).$$

Moreover, if we set $A_0 = 0$ and $B_0 = 1$ then for $t \geq 0$,

$$A_t \geq \frac{1}{4\sigma_{F/\phi}}\left[\pi_t^+ - \pi_t^-\right]^2, \qquad B_t = 1 + \sigma_{F/\phi} A_t \geq \frac{1}{4}\left[\pi_t^+ + \pi_t^-\right]^2,$$

where

$$\pi_t^+ = \prod_{i=0}^{t-1}\left(1 + \sqrt{\frac{\sigma_{F/\phi}}{L_{F/\phi} G_t}}\right), \qquad \pi_t^- = \prod_{i=0}^{t-1}\left(1 - \sqrt{\frac{\sigma_{F/\phi}}{L_{F/\phi} G_t}}\right).$$

We first state an equivalent definition of relative smoothness and relative strong convexity (Lu et al., 2018). The function F is said to be $L_{F/\phi}$-smooth and $\sigma_{F/\phi}$-strongly convex with respect to ϕ if for all $x, y \in \mathbb{R}^d$,

$$F(y) + \nabla F(y)^\top (x - y) + \sigma_{L/\phi} D_\phi(x, y) \;\leq\; F(x) \;\leq\; F(y) + \nabla F(y)^\top (x - y) + L_{L/\phi} D_\phi(x, y). \tag{29}$$

Obviously this is the same as (9). We also need the following lemma, which is an extension of a result from Chen and Teboulle (1993, Lemma 3.2), whose proof we omit.

Lemma 3 (Descent property of Bregman proximal point)**.** *Suppose g is a convex function defined over $\operatorname{dom}\phi$ and*

$$v_{t+1} = \operatorname*{argmin}_x \big\{g(x) + (1 - \beta_t) D_\phi(x, v_t) + \beta_t D_\phi(x, y_t)\big\},$$

then for any $x \in \operatorname{dom} h$,

$$g(v_{t+1}) + (1 - \beta_t) D_\phi(v_{t+1}, v_t) + \beta_t D_\phi(v_{t+1}, y_t) \leq g(x) + (1 - \beta_t) D_\phi(x, v_t) + \beta_t D_\phi(x, y_t) - D_\phi(x, v_{t+1}).$$

Proof of Theorem 5. The proof follows the same lines as Nesterov and Stich (2017), with adaptations to use general Bregman divergences. Applying Lemma 3 with $g(x) = \eta_t\big(\nabla f(y_t)^\top x + \psi(x)\big)$, we have for any $x \in \operatorname{dom}\phi$,

$$\begin{aligned} &D(x, v_{t+1}) + (1 - \beta_t) D(v_{t+1}, v_t) + \beta_t D(v_{t+1}, y_t) - (1 - \beta_t) D(x, v_t) - \beta_t D(x, y_t) \\ \leq\; &\eta_t \nabla f(y_t)^\top (x - v_{t+1}) + \eta_t\big(\psi(x) - \psi(v_{t+1})\big). \end{aligned}$$

Since by definition $\eta_t = \frac{a_{t+1}}{B_{t+1}}$, multiplying both sides of the above inequality by B_{t+1} yields

$$\begin{aligned} &B_{t+1} D(x, v_{t+1}) + B_{t+1}\big((1 - \beta_t) D(v_{t+1}, v_t) + \beta_t D(v_{t+1}, y_t)\big) - B_{t+1}(1 - \beta_t) D(x, v_t) - B_{t+1}\beta_t D(x, y_t) \\ \leq\; &a_{t+1} \nabla f(y_t)^\top (x - v_{t+1}) + a_{t+1}\big(\psi(x) - \psi(v_{t+1})\big). \end{aligned}$$

Using the scaling property (19) and the relationships $\alpha_t = \frac{a_{t+1}}{A_{t+1}}$ and $a_{t+1}^2 L_{f/\phi} G_t = A_{t+1} B_{t+1}$, we obtain

$$B_{t+1}\big((1 - \beta_t) D(v_{t+1}, v_t) + \beta_t D(v_{t+1}, y_t)\big) \geq \frac{B_{t+1}}{\alpha_t^2 G_t} D(x_{t+1}, y_t) = \frac{A_{t+1}^2 B_{t+1}}{a_{t+1}^2 G_t} D(x_{t+1}, y_t) = A_{t+1} L_{f/\phi} D(x_{t+1}, y_t).$$

Combining the last two inequalities and using the facts $B_{t+1}(1 - \beta_t) = B_t$ and $B_{t+1}\beta_t = a_{t+1}\sigma_{f/\phi}$, we arrive at

$$\begin{aligned} &B_{t+1} D(x, v_{t+1}) + A_{t+1} L_{f/\phi} D(x_{t+1}, y_t) - B_t D(x, v_t) - a_{t+1}\sigma_{f/\phi} D(x, y_t) \\ \leq\; &a_{t+1} \nabla f(y_t)^\top (x - v_{t+1}) + a_{t+1}\big(\psi(x) - \psi(v_{t+1})\big). \end{aligned} \tag{30}$$

We then expand the gradient term on the right-hand side of (30) into two parts:

$$a_{t+1} \nabla f(y_t)^\top (x - v_{t+1}) = a_{t+1} \nabla f(y_t)^\top (x - w_t) + a_{t+1} \nabla f(y_t)^\top (w_t - v_{t+1}), \tag{31}$$

where $w_t = (1-\beta_t)v_t + \beta_t y_t$. For the first part,

$$a_{t+1}\nabla f(y_t)^\top (x - w_t) = a_{t+1}\nabla f(y_t)^\top (x - y_t) + \frac{a_{t+1}(1-\alpha_t)}{\alpha_t}\nabla f(y_t)^\top (x_t - y_t)$$
$$\leq a_{t+1}\left(f(x) - f(y_t) - \sigma_{f/\phi} D(x, y_t)\right) + \frac{a_{t+1}(1-\alpha_t)}{\alpha_t}\left(f(x_t) - f(y_t)\right). \tag{32}$$

Notice that

$$a_{t+1}\frac{1-\alpha_t}{\alpha_t} = a_{t+1}\left(\frac{1}{\alpha_t} - 1\right) = a_{t+1}\left(\frac{A_{t+1}}{a_{t+1}} - 1\right) = A_{t+1} - a_{t+1} = A_t.$$

Therefore, Equation (32) becomes

$$a_{t+1}\nabla f(y_t)^\top (x - w_t) \leq a_{t+1} f(x) - A_{t+1} f(y_t) + A_t f(x_t) - a_{t+1}\sigma_{f/\phi} D(x, y_t). \tag{33}$$

For the second part on the right-hand side of (31),

$$a_{t+1}\nabla f(y_t)^\top (w_t - v_{t+1}) = -\frac{a_{t+1}}{\alpha_t}\nabla f(y_t)^\top (x_{t+1} - y_t) = -A_{t+1}\nabla f(y_t)^\top (x_{t+1} - y_t)$$
$$\leq \; -A_{t+1}\left(f(x_{t+1}) - f(y_t) - L_{f/\phi} D(x_{t+1}, y_t)\right), \tag{34}$$

where in the last inequality we used the relative smoothness assumption in (29).

Summing the inequalities (30), (32) and (34), we have

$$B_{t+1}D(x, v_{t+1}) - B_t D(x, v_t) \leq a_{t+1}f(x) - A_{t+1}f(x_{t+1}) + A_t f(x_t) + a_{t+1}(\psi(v_{t+1}) - \psi(x))$$
$$\leq -A_{t+1}\big(f(x_{t+1}) - f(x)\big) + A_t\big(f(x_t) - f(x)\big) + a_{t+1}\big(\psi(x) - \psi(v_{t+1})\big),$$

which is the same as

$$A_{t+1}\big(f(x_{t+1}) - f(x)\big) + B_{t+1}D(x, v_{t+1}) \leq A_t\big(f(x_t) - f(x)\big) + B_t D(x, v_t) + a_{t+1}\big(\psi(x) - \psi(v_{t+1})\big). \tag{35}$$

Finally we consider the term $a_{t+1}\big(\psi(x) - \psi(v_{t+1})\big)$. Using $x_{t+1} = (1-\alpha_t)x_t + \alpha_t v_{t+1}$ and convexity of ψ, we have

$$\psi(x_{t+1}) \leq (1-\alpha_t)\psi(x_t) + \alpha_t \psi(v_{t+1}).$$

Since by definition $\alpha_t = \frac{a_{t+1}}{A_{t+1}}$ and $(1-\alpha_t) = \frac{A_t}{A_{t+1}}$, the above inequality is equivalent to

$$A_{t+1}\psi(x_{t+1}) \leq A_t\psi(x_t) + a_{t+1}\psi(v_{t+1}),$$

which implies (using $A_{t+1} = A_t + a_{t+1}$) that for any $x \in \operatorname{dom}\phi$,

$$A_{t+1}\big(\psi(x_{t+1}) - \psi(x)\big) \leq A_t\big(\psi(x_t) - \psi(x)\big) + a_{t+1}\big(\psi(v_{t+1}) - \psi(x)\big). \tag{36}$$

Summing the inequalities (35) and (36) and using $\Phi = f + \psi$, we have

$$A_{t+1}\big(\Phi(x_{t+1}) - \Phi(x)\big) + B_{t+1}D(x, v_{t+1}) \leq A_t\big(\Phi(x_t) - \Phi(x)\big) + B_t D(x, v_t).$$

This can then be unrolled, and we obtain the desired result by setting $x = x_*$.

Finally, the estimates of A_t and B_t follow from a direct adaptation of the techniques in (Nesterov and Stich, 2017). The only difference is the use of time-varying $\gamma_t = \sqrt{\sigma_{F/\phi}/(L_{F/\phi}G_t)}$ instead of a constant $\gamma = \sqrt{\sigma_{F/\phi}/L_{F/\phi}}$, which does not impact the derivations. □

A.3. Proof of Lemma 2

The analysis in Lemma 1 is very pessimistic, since we use uniform lower and upper bounds for the Hessian of ϕ, whereas what we actually want is to bound is the differences between Hessians. If the Hessian is well-behaved (typically Lipschitz, or if ϕ is self-concordant), we can prove Lemma 2, which leads to a finer asymptotic convergence rate.

We start with the local quadratic representation of Bregman divergence:

$$\begin{aligned}
D_\phi(x_{t+1}, y_t) &= \frac{1}{2}\|x_{t+1} - y_t\|^2_{H(x_{t+1},y_t)} = \frac{\alpha_t^2}{2}\|v_{t+1} - w_t\|^2_{H(x_{t+1},y_t)} \\
&\le \frac{\alpha_t^2}{2}\Big((1-\beta_t)\|v_{t+1} - v_t\|^2_{H(x_{t+1},y_t)} + \beta_t\|v_{t+1} - y_t\|^2_{H(x_{t+1},y_t)}\Big) \\
&\le \frac{\alpha_t^2}{2}\Big((1-\beta_t)\|v_{t+1} - v_t\|^2_{H(v_{t+1},v_t)} + \beta_t\|v_{t+1} - y_t\|^2_{H(v_{t+1},y_t)}\Big) \\
&\quad + \frac{\alpha_t^2}{2}(1-\beta_t)\|H(x_{t+1},y_t) - H(v_{t+1},v_t)\| \cdot \|v_{t+1} - v_t\|^2 \\
&\quad + \frac{\alpha_t^2}{2}\beta_t\|H(x_{t+1},y_t) - H(v_{t+1},y_t)\| \cdot \|v_{t+1} - y_t\|^2.
\end{aligned}$$

Now we use the Lipschitz property of $\nabla^2\phi$ to bound the spectral norms of differences of Hessians:

$$\|H(x_{t+1},y_t) - H(v_{t+1},v_t)\| \le M\|z_{xy} - z_{vv}\|, \qquad \|H(x_{t+1},y_t) - H(v_{t+1},y_t)\| \le M\|z_{xy} - z_{vy}\|,$$

where $z_{vv} \in [v_{t+1}, v_t]$, $z_{xy} \in [y_t, x_{t+1}]$ and $z_{vy} \in [y_t, v_{t+1}]$. Using the triangle inequality of norms, we have

$$\|z_{xy} - z_{vy}\| = \|z_{xy} - y_t + y_t - z_{vy}\| \le \|z_{xy} - y_t\| + \|y_t - z_{vy}\| \le \|x_{t+1} - y_t\| + \|y_t - v_{t+1}\|,$$

and

$$\|z_{vv} - z_{xy}\| \le \|z_{vv} - v_{t+1}\| + \|v_{t+1} - y_t\| + \|y_t - z_{xy}\| \le \|v_t - v_{t+1}\| + \|v_{t+1} - y_t\| + \|y_t - x_{t+1}\|.$$

Therefore, we have

$$d_t \triangleq \max\big\{\|z_{xy} - z_{vv}\|,\ \|z_{xy} - z_{vy}\|\big\} \le \|v_t - v_{t+1}\| + \|v_{t+1} - y_t\| + \|y_t - x_{t+1}\|,$$

and consequently,

$$\begin{aligned}
D_\phi(x_{t+1}, y_t) &\le \frac{\alpha_t^2}{2}\Big((1-\beta_t)\|v_{t+1} - v_t\|^2_{H(v_{t+1},v_t)} + \beta_t\|v_{t+1} - y_t\|^2_{H(v_{t+1},y_t)}\Big) \\
&\quad + \frac{Md_t\alpha_t^2}{2}\Big((1-\beta_t)\|v_{t+1} - v_t\|^2 + \beta_t\|v_{t+1} - y_t\|^2\Big) \\
&\le \frac{\alpha_t^2}{2}\Big((1-\beta_t)\|v_{t+1} - v_t\|^2_{H(v_{t+1},v_t)} + \beta_t\|v_{t+1} - y_t\|^2_{H(v_{t+1},y_t)}\Big) \\
&\quad + \frac{Md_t\alpha_t^2}{2\sigma_\phi}\Big((1-\beta_t)\|v_{t+1} - v_t\|^2_{H(v_{t+1},v_t)} + \beta_t\|v_{t+1} - y_t\|^2_{H(v_{t+1},y_t)}\Big) \\
&= \frac{\alpha_t^2}{2}\left(1 + \frac{Md_t}{\sigma_\phi}\right)\Big((1-\beta_t)\|v_{t+1} - v_t\|^2_{H(v_{t+1},v_t)} + \beta_t\|v_{t+1} - y_t\|^2_{H(v_{t+1},y_t)}\Big) \\
&= \alpha_t^2\left(1 + \frac{Md_t}{\sigma_\phi}\right)\Big((1-\beta_t)D(v_{t+1},v_t) + \beta_t D(v_{t+1},y_t)\Big).
\end{aligned}$$

Combining with Lemma 1, we see that $G_t = \min\{\kappa_\sigma,\ 1 + (M/\sigma_\phi)d_t\}$ satisfies the inequality (19). This finishes the proof of Lemma 2.

Note that this condition is not directly useful. Indeed, x_{t+1} and v_{t+1} depend on G_t. Yet, under the uniform choice of $G_t \le \kappa_\phi$, it can be shown that $d_t \to 0$ at rate $(1 - 1/\sqrt{\kappa_\phi\kappa_{F/\phi}})^t$ because the sequences v_t, x_t and y_t all converge to x^* at this rate in the strongly convex case (Lin and Xiao, 2015, Theorem 1). As a consequence, Algorithm 1 will eventually use $G_t \le 2$, leading to an asymptotic rate of $(1 - 1/\sqrt{\kappa_{F/\phi}})^t$.

B. Concentration of Hessians

In practice, preconditioned gradient methods such as DANE are often used with a step-size of 1. This implies the assumption of $L_{F/\phi} = 1$, which holds if n is sufficiently large with a given μ or if μ is sufficiently large for a given n (but $\mu \leq L_F$ always). Otherwise convergence is not guaranteed (which is why it is sometimes considered as "rather unstable"). If μ is such that $\|H_f(x) - H_F(x)\| \leq \mu$ for all $x \in \mathcal{B}(0, D)$ then $L_{F/\phi} = 1$ can safely be chosen since $H_F(x) - H_f(x) \preceq \mu I_d$. Note that this choice of μ is completely independent of λ. In this case, we use that $H_F(x) - H_f(x) \succeq -\mu I$ to write that

$$H_f(x) + \mu \preceq H_F(x) + 2\mu \preceq (1 + 2\mu H_F^{-1}(x))H_F(x) \preceq \left(1 + \frac{2\mu}{\lambda}\right) H_F(x).$$

These derivations are similar to the ones of Zhang and Xiao (2018, Lemma 3), and so we obtain $\sigma_{F/\phi} = \left(1 + \frac{2\mu}{\lambda}\right)^{-1}$ and the corresponding relative condition number $\kappa_{F/\phi} = 1 + \frac{2\mu}{\lambda}$, as explained in Section 1.1. We see that μ is independent of λ, but the problem is still very ill-conditioned for small values of λ, meaning that acceleration makes a lot of sense. In the quadratic case, tighter relative bounds can be derived.

B.1. The quadratic case

This section is focusing on proving Theorem 2.

Proof of Theorem 2. We consider the random variable a, and $(a_i)_{i\in\{1,\dots,n\}}$ are n i.i.d. variables with the same law as a. We introduce matrices $\hat{H}$ and H such that $H_f = \hat{H} + \lambda I_d$ and $H_F = H + \lambda I_d$. In particular, $H = \mathbb{E}\left[aa^\top\right] = \mathbb{E}\hat{H}$. We define for $\alpha \geq 0$, $\beta > 0$, $H_{\alpha,\beta} = \alpha H + \beta I_d$, and

$$S_i = \frac{1}{n} H_{\alpha,\beta}^{-\frac{1}{2}}(a_i a_i^\top - H)H_{\alpha,\beta}^{-\frac{1}{2}},$$

which is such that $\mathbb{E}[S_i] = 0$. This allows to have bounds of the form $\|\sum_i S_i\| \leq t$ with probability $1 - \delta$ and a spectral bound μ that depends on α, β, δ (and other quantities related to H and $a_i a_i^\top$). We note that

$$\sum_{i=1}^n S_i = H_{\alpha,\beta}^{-\frac{1}{2}}(\hat{H} - H)H_{\alpha,\beta}^{-\frac{1}{2}},$$

and write the concentration bounds on the S_i as $-tH_{\alpha,\beta} \preceq \hat{H} - H \preceq tH_{\alpha,\beta}$ for some $t > 0$, which can be rearranged as:

$$\hat{H} + t\beta I_d \succeq (1 - t\alpha)H$$
$$\hat{H} - t\beta I_d \preceq (1 + t\alpha)H.$$

Using $H_f = \hat{H} + \lambda I_d$ and $H_F = H + \lambda I_d$, the first equation can be rearranged as:

$$H_F \preceq \frac{1}{1 - t\alpha}\left(H_f + t(\beta - \alpha\lambda)I_d\right). \tag{37}$$

The second equation can be written

$$H_f \preceq \left[(1 + t\alpha)I_d + t(\beta - \alpha\lambda)H_F^{-1}\right] H_F,$$

which, by adding $t(\beta - \alpha\lambda)I_d$ on both sides, leads to

$$H_f + t(\beta - \alpha\lambda)I_d \preceq \left[(1 + t\alpha)I_d + 2t(\beta - \alpha\lambda)H_F^{-1}\right] H_F.$$

We let $\mu = t(\beta - \alpha\lambda)$ and use $H_F^{-1} \preceq \lambda^{-1} I_d$ to write that:

$$\left(1 + \alpha t + \frac{2\mu}{\lambda}\right)^{-1}(H_f + \mu I_d) \preceq H_F \preceq \frac{1}{1 - \alpha t}\left(H_f + \mu I_d\right). \tag{38}$$

We then use the fact that $a_i a_i^\top$ and H are positive semidefinite and upper bounded by $R^2 I$ to write that:

$$\|S_i\| \leq \frac{1}{n}\|H_{\alpha,\beta}^{-1}\| \max\{\|aa^\top\|, \|H\|\} \leq \frac{R^2}{\beta n}. \tag{39}$$

Using the fact that $H = \mathbb{E}\left[aa^\top\right]$, we bound the variance as:

$$\begin{aligned}
\left\|\sum_i \mathbb{E}\left[S_i S_i^\top\right]\right\| &= \frac{1}{n}\left\|\mathbb{E}\left[H_{\alpha,\beta}^{-\frac{1}{2}}(aa^\top - H)H_{\alpha,\beta}^{-1}(aa^\top - H)H_{\alpha,\beta}^{-\frac{1}{2}}\right]\right\| \\
&= \frac{1}{n}\left\|H_{\alpha,\beta}^{-\frac{1}{2}}(\mathbb{E}\left[aa^\top H_{\alpha,\beta}^{-1} aa^\top\right] - HH_{\alpha,\beta}^{-1}H)H_{\alpha,\beta}^{-\frac{1}{2}}\right\| \\
&\leq \frac{1}{n}\max\left\{\tilde{R}^2\left\|H_{\alpha,\beta}^{-\frac{1}{2}}\mathbb{E}\left[aa^\top\right]H_{\alpha,\beta}^{-\frac{1}{2}}\right\|,\ \left\|H_{\alpha,\beta}^{-\frac{1}{2}}HH_{\alpha,\beta}^{-1}HH_{\alpha,\beta}^{-\frac{1}{2}}\right\|\right\} \\
&\leq \frac{1}{n}\left\|H_{\alpha,\beta}^{-\frac{1}{2}}HH_{\alpha,\beta}^{-\frac{1}{2}}\right\| \max\left\{\tilde{R}^2,\ \left\|H_{\alpha,\beta}^{-\frac{1}{2}}HH_{\alpha,\beta}^{-\frac{1}{2}}\right\|\right\},
\end{aligned}$$

with $\tilde{R}^2 \geq a^\top H_{\alpha,\beta}^{-1} a$ almost surely. We first notice that $a_i^\top H_{\alpha,\beta}^{-1} a_i \leq \frac{R^2}{\beta}$. Then, we use the positive definiteness of $H_{\alpha,\beta}$ and H and the fact that $\beta H_{\alpha,\beta}^{-1} \preceq I_d$ to show that for $\alpha > 0$:

$$\left\|H_{\alpha,\beta}^{-\frac{1}{2}}HH_{\alpha,\beta}^{-\frac{1}{2}}\right\| = \alpha^{-1}\left\|H_{\alpha,\beta}^{-\frac{1}{2}}(\alpha H + \beta - \beta)H_{\alpha,\beta}^{-\frac{1}{2}}\right\| = \alpha^{-1}\left\|I_d - \beta H_{\alpha,\beta}^{-1}\right\| \leq \alpha^{-1}\left(1 - \frac{\beta}{\alpha L + \beta}\right) = \frac{L}{\alpha L + \beta},$$

where L is the spectral norm of H, i.e., $L = \|H\|$. A quick calculation shows that this formula is also true for $\alpha = 0$. In the case $\alpha = 0$ and $\beta = 1$ (absolute bounds), $H_{\alpha,\beta} = I_d$ and we recover that we can bound the variance by $\frac{LR^2}{n}$, leading to the usual additive bounds.

For $\alpha > 0$, we use the simpler bound $\left\|H_{\alpha,\beta}^{-\frac{1}{2}}HH_{\alpha,\beta}^{-\frac{1}{2}}\right\| \leq \alpha^{-1}$ and $\tilde{R}^2 \leq \beta^{-1}R^2$, leading to

$$\left\|\sum_i \mathbb{E}\left[S_i S_i^\top\right]\right\| \leq \frac{\max(\beta^{-1}R^2, \alpha^{-1})}{n\alpha}.$$

For any $1 > \delta > 0$, we note $c_\delta = \frac{28}{3}\log\left(\frac{2d}{\delta}\right)$. We now set $\alpha = \frac{\beta n}{c_\delta R^2}$, and assume that $n > c_\delta$ (otherwise concentration bounds will be very loose anyway). In this case, $\beta^{-1}R^2 \geq \alpha^{-1}$, meaning that the bound on the variance becomes:

$$\left\|\sum_i \mathbb{E}\left[S_i S_i^\top\right]\right\| \leq \frac{1}{\alpha^2 c_\delta}.$$

Similarly, according to (39), every S_i is almost surely bounded as: $\|S_i\| \leq \frac{1}{\alpha c_\delta}$. We can now use Matrix Bernstein Inequality (Tropp, 2015, Theorem (6.1.1)) to get that with probability $1 - p_\delta$ and for $t \geq 0$,

$$\left\|\sum_{i=1}^n S_i\right\| \leq t,$$

with

$$p_\delta = 2d \cdot \exp\left(-\frac{t^2/2}{(\alpha^2 c_\delta)^{-1} + (\alpha c_\delta)^{-1}t/3}\right).$$

We choose $t = (2\alpha)^{-1}$, which leads to $p_\delta = \delta$. By substituting the expressions of $\alpha t = \frac{1}{2}$ and $\beta t = \frac{R^2 c_\delta}{n}\alpha t$ into Equation (38), we obtain:

$$\left(\frac{3}{2} + \frac{2\mu}{\lambda}\right)^{-1}(\hat{H}_\lambda + \mu I_d) \preceq H_\lambda \preceq 2\left(\hat{H}_\lambda + \mu I_d\right),$$

with

$$\mu = t(\beta - \alpha\lambda) = \frac{1}{2}\left(\frac{28R^2}{3n}\log\left(\frac{2d}{\delta}\right) - \lambda\right).$$

In case β is very small so that $\mu < 0$ then it is always possible to choose $\delta' < \delta$ so that $\mu > 0$. This means that the same bound on μ holds with probability $1 - \delta' > 1 - \delta$. □

B.2. Almost surely bounded a

We first introduce Theorem 6, which proves a general concentration result that implies Theorem 3 as a special case.

Theorem 6. *We consider functions φ_1, φ_2, which are respectively L_1 and L_2 Lipschitz-continuous. We consider two sets $\mathcal{X}$ and $\mathcal{Y}$ which are contained in balls of center 0 and radius D_1 and D_2. We assume that $|\varphi_1(a_i^\top x)| \leq B_1$ and $|\varphi_2(a_i^\top y)| \leq B_2$ almost surely for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. We consider*

$$Y = \sup_{x\in\mathcal{X},\ y\in\mathcal{Y}} \Big\{ \frac{1}{n}\sum_{i=1}^n \varphi_1(a_i^\top x)\varphi_2(a_i^\top y) - \mathbb{E}\varphi_1(a^\top x)\varphi_2(a^\top y)\Big\}.$$

Then, for all $1 \geq \delta > 0$, with probability greater than $1-\delta$:

$$Y \leq \sqrt{4\pi}\frac{(\mathbb{E}\left[\|a\|^2\right])^{\frac{1}{2}}}{\sqrt{n}}(B_2L_1D_1 + B_1L_2D_2) + \frac{2B_1B_2}{\sqrt{2n}}\sqrt{\log\frac{1}{\delta}}.$$

Theorem 3 is then a direct corollary of Theorem 6, as shown below:

Proof of Theorem 3. The result is obtained by applying Theorem 6 with $\varphi_1 = \ell''$ and $\varphi_2 = \frac{1}{2}(\cdot)^2$. This implies that with probability at least $1-\delta$,

$$\sup_{x\in\mathcal{B}(0,D),\ y\in\mathcal{B}(0,1)} y^\top\Big[\frac{1}{n}\sum_{i=1}^n \ell''(a_i^\top x)a_ia_i^\top - \mathbb{E}\ell''(a^\top x)aa^\top\Big]y \leq \mu,$$

where the value of μ can be obtained by letting $B_1 = B_\ell$, $L_1 = M_\ell$, $D_1 = D$, $D_2 = 1$, $B_2 = \sup_{y:\|y\|\leq 1} y^\top a_ia_i^\top y \leq R^2$ and $L_2 = \sup_{y:\|y\|\leq 1} 2\|y^\top a_i\| = 2R$. □

Proof of Theorem 6. If changing any a_i to some a_i', then the deviation in Y is at most (almost surely):

$$\frac{1}{n}\sup_{x\in\mathcal{X},\ y\in\mathcal{Y}}\big|\varphi_1(a_i^\top x)\varphi_2(a_i^\top y)\big| + \sup_{x\in\mathcal{X},\ y\in\mathcal{Y}}\big|\varphi_1(a_i'^\top x)\varphi_2(a_i'^\top y)\big| \leq \frac{2}{n}B_1B_2.$$

Mac-Diarmid's inequality (see, e.g., Vershynin, 2019, Theorem 2.9.1) thus implies that with probability greater than $1-\delta$,

$$Y \leq \mathbb{E}Y + \frac{2B_1B_2}{\sqrt{2n}}\sqrt{\log\frac{1}{\delta}}\,. \tag{40}$$

In order to bound $\mathbb{E}Y$, we first use classical symmetrization property (see, e.g., Vershynin, 2019, Section 6.4)

$$\mathbb{E}Y \leq \sqrt{2\pi}\cdot\mathbb{E}\sup_{x\in\mathcal{X},\ y\in\mathcal{Y}}\frac{1}{n}\sum_{i=1}^n \varepsilon_i\varphi_1(a_i^\top x)\varphi_2(a_i^\top y),$$

where each ε_i is an independent standard normal variable.

Denoting $Z_{x,y} = \frac{1}{n}\sum_{i=1}^n \varepsilon_i\varphi_1(a_i^\top x)\varphi_2(a_i^\top y)$, we have, for any x, y, x', y', assuming the a_i are fixed,

$$\begin{aligned}
\mathbb{E}(Z_{x,y} - Z_{x',y'})^2 &= \frac{1}{n^2}\sum_{i=1}^n\Big(\varphi_1(a_i^\top x)\varphi_2(a_i^\top y) - \varphi_1(a_i^\top x')\varphi_2(a_i^\top y')\Big)^2 \\
&= \frac{1}{n^2}\sum_{i=1}^n\Big(\varphi_1(a_i^\top x)\big[\varphi_2(a_i^\top y) - \varphi_2(a_i^\top y')\big] + \big[\varphi_1(a_i^\top x) - \varphi_1(a_i^\top x')\big]\varphi_2(a_i^\top y')\Big)^2 \\
&\leq \frac{1}{n^2}\sum_{i=1}^n\Big(2\,\varphi_1(a_i^\top x)^2\big[\varphi_2(a_i^\top y) - \varphi_2(a_i^\top y')\big]^2 + 2\,\varphi_2(a_i^\top y')^2\big[\varphi_1(a_i^\top x) - \varphi_1(a_i^\top x')\big]^2\Big) \\
&\leq \frac{1}{n^2}\sum_{i=1}^n\Big(2B_1^2\big[\varphi_2(a_i^\top y) - \varphi_2(a_i^\top y')\big]^2 + 2B_2^2\big[\varphi_1(a_i^\top x) - \varphi_1(a_i^\top x')\big]^2\Big).
\end{aligned}$$

We then have, using Lipschitz-continuity:

$$\begin{aligned}\mathbb{E}(Z_{x,y}-Z_{x',y'})^2 &\leq \frac{1}{n^2}\sum_{i=1}^n\left(2B_1^2L_2^2\left[a_i^\top y-a_i^\top y'\right]^2+2B_2^2L_1^2\left[a_i^\top x-a_i^\top x'\right]^2\right)\\ &= \mathbb{E}(\tilde{Z}_{x,y}-\tilde{Z}_{x',y'})^2,\end{aligned}$$

for

$$\tilde{Z}_{x,y}=\frac{1}{n}\sum_{i=1}^n\left\{\sqrt{2}B_2L_1\tilde{\varepsilon}_{1i}a_i^\top x+\sqrt{2}B_1L_2\tilde{\varepsilon}_{2i}a_i^\top y\right\},$$

with all $\tilde{\varepsilon}_{1i}$ and $\tilde{\varepsilon}_{2i}$ independent standard random variables.

Using Sudakov-Fernique inequality (Vershynin, 2019, Theorem 7.2.11), we get

$$\begin{aligned}\mathbb{E}Y &= \sqrt{2\pi}\mathbb{E}\sup_{x\in\mathcal{X},\,y\in\mathcal{Y}}Z_{x,y}\\ &\leq \sqrt{2\pi}\mathbb{E}\sup_{x\in\mathcal{X},\,y\in\mathcal{Y}}\tilde{Z}_{x,y}\\ &= \sqrt{4\pi}B_2L_1\mathbb{E}\sup_{x\in\mathcal{X}}\frac{1}{n}\sum_{i=1}^n\tilde{\varepsilon}_{1i}a_i^\top x+\sqrt{4\pi}B_1L_2\mathbb{E}\sup_{y\in\mathcal{Y}}\frac{1}{n}\sum_{i=1}^n\tilde{\varepsilon}_{2i}a_i^\top y\\ &\leq \sqrt{4\pi}B_2L_1D_1\mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^n\tilde{\varepsilon}_{1i}a_i\right\|+\sqrt{4\pi}B_1L_2D_2\mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^n\tilde{\varepsilon}_{2i}a_i\right\|\\ &\leq \sqrt{4\pi}B_2L_1D_1\sqrt{\mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^n\tilde{\varepsilon}_{1i}a_i\right\|^2}+\sqrt{4\pi}B_1L_2D_2\sqrt{\mathbb{E}\left\|\frac{1}{n}\sum_{i=1}^n\tilde{\varepsilon}_{2i}a_i\right\|^2}\\ &\leq \sqrt{4\pi}B_2L_1D_1\frac{\left(\mathbb{E}\left[\|a\|^2\right]\right)^{\frac{1}{2}}}{\sqrt{n}}+\sqrt{4\pi}B_1L_2D_2\frac{\left(\mathbb{E}\left[\|a\|^2\right]\right)^{\frac{1}{2}}}{\sqrt{n}}.\end{aligned}$$

Plugging this into Equation (40), we obtain that with probability greater than $1-\delta$,

$$Y\leq\sqrt{4\pi}\frac{\left(\mathbb{E}\left[\|a\|^2\right]\right)^{\frac{1}{2}}}{\sqrt{n}}(B_2L_1D_1+B_1L_2D_2)+\frac{2B_1B_2}{\sqrt{2n}}\sqrt{\log\frac{1}{\delta}}.$$

□

Remark 4 (Relative bounds)**.** In the quadratic case, considering relative bounds allowed to choose smaller values of μ and to tighten the bounds on the relative condition number by a $\sqrt{n}$ factor. Theorem 3 consists in bounding (using the definition of the operator norm)

$$\sup_{x\in\mathcal{B}(0,D),y\in\mathcal{B}(0,1)}\left\{\frac{1}{n}\sum_{i=1}^n\ell''(a_i^\top x)(a_i^\top y)^2-y^\top H(x)y\right\},$$

and heavily relies on the fact that $(a_i^\top y)^2$ is independent of x. The proof needs to be adapted in the case of the relative bounds since this term becomes $(a_i^\top H_{\alpha,\beta}^{-\frac{1}{2}}(x)y)^2$, which now depends on x as well, and thus requires a different control.

B.3. Subgaussian a

We considered in the previous section a splitting of the summands of the Hessians as a product of 2 functions. We now present a different bound that is designed for a product of an arbitrary number of functions $\varphi_1,\dots,\varphi_r:\mathbb{R}\to\mathbb{R}$. This section is devoted to proving Theorem 7, which is based on the chaining argument (Boucheron et al., 2013, Chapter 13), and from which Theorem 4 can be derived directly.

Theorem 7. *Assume that for all i, $\varphi_i(0)=0$ and φ_i is 1-Lipschitz. Assume that a is ρ-subgaussian, and that for all k, $\sup_{x\in\mathcal{B}(0,1)}|\varphi_k(a_i^\top x)|\leq B_k$. Denote $B=\prod_{k=1}^r B_k$. For suitable constant C_r, for all $\gamma>0$, one has that*

$$\mathbb{P}\left(\sup_{x_1,\dots,x_r\in\mathcal{B}(0,1)}\frac{1}{n}\sum_{i\in[n]}\left\{\prod_{k=1}^r\varphi_k(a_i^\top x_k)-\mathbb{E}\prod_{k=1}^r\varphi_k(a^\top x_k)\right\}\geq\rho^rC_r(d+\gamma)\left[\frac{1}{\sqrt{dn}}+\frac{(\rho^{-r}B)^{1-2/r}}{n}\right]\right)\leq r\frac{\pi^2}{6}e^{-\gamma}.$$

We are primarily interested in the case $r = 3$, $\varphi_1 = \varphi_2 = \mathrm{id}$ (the identity mapping) to control distances between Hessians.

Proof. We look for bounds on

$$Y := \sup_{x_1,\dots,x_r \in \mathcal{S}_1} \frac{1}{n} \sum_{i\in[n]} \left\{ \prod_{k=1}^{r} \varphi_k(a_i^\top x_k) - \mathbb{E}_a \prod_{k=1}^{r} \varphi_k(a^\top x_k) \right\}. \tag{41}$$

For all $j \geq 0$, let $\mathcal{N}_j$ be an ϵ-net of $\mathcal{S}_1$ that approximates $\mathcal{S}_1$ to distance 2^{-j}. Then, $\mathcal{N}_j$ can be chosen as $|\mathcal{N}_j| \leq (1 + 2^{j+1})^d$(see, e.g., Vershynin, 2019, Section 4.2). For all $x \in \mathcal{S}_1$, let $\Pi_j(x)$ be some point in $\mathcal{N}_j$ such that $\|x - \Pi_j(x)\| \leq 2^{-j}$. By convention we take $\Pi_0(x) = 0$.

Then for all $(x_1, \dots, x_r) \in \mathcal{S}^r$, using the chaining approach (Boucheron et al., 2013), we write

$$\begin{aligned}
\frac{1}{n} \sum_{i\in[n]} \prod_{k\in[r]} \varphi_k(a_i^\top x_k) &= \sum_{j\geq 0} \frac{1}{n} \sum_{i\in[n]} \left\{ \prod_{k\in[r]} \varphi_k(a_i^\top \Pi_{j+1}(x_k)) - \prod_{k\in[r]} \varphi_k(a_i^\top \Pi_j(x_k)) \right\} \\
&= \sum_{j\geq 0} \sum_{k\in[r]} \frac{1}{n} \sum_{i\in[n]} \prod_{\ell=1}^{k-1} \varphi_\ell(a_i^\top \Pi_{j+1}(x_\ell)) \big[\varphi_k(a_i^\top \Pi_{j+1}(x_k)) - \varphi_k(a_i^\top \Pi_j(x_k))\big] \prod_{\ell=k+1}^{r} \varphi_\ell(a_i^\top \Pi_j(x_\ell)).
\end{aligned}$$

Let $j \geq 0$ and $k \in [r]$ be fixed. Consider a term of the form $Z = \frac{1}{n} \sum_{i\in[n]} Z_i$, with

$$Z_i = \prod_{\ell=1}^{k-1} \varphi_\ell(a_i^\top u_\ell) \big[\varphi_k(a_i^\top u_k) - \varphi_k(a_i^\top v_k)\big] \prod_{\ell=k+1}^{r} \varphi_\ell(a_i^\top v_\ell), \tag{42}$$

where $u_\ell \in \mathcal{N}_j$, $v_\ell \in \mathcal{N}_{j+1}$, and $\|u_k - v_k\| \leq \epsilon_j := 2^{-j+1}$. By the triangle inequality, for all $x_\ell \in \mathcal{S}_1$, letting $u_\ell = \Pi_j(x_\ell)$ and $v_\ell = \Pi_{j+1}(x_\ell)$, these assumptions are satisfied. For each Z_i and $t > 0$, we have:

$$\begin{aligned}
\mathbb{P}(Z_i \geq \epsilon_j \rho^r t) \leq \mathbb{P}\Big(&|\varphi_\ell(a_i^\top u_\ell)| \geq \rho t^{1/r} \text{ for some } \ell < r, \\
&\text{or } |\varphi_k(a_i^\top u_k) - \varphi_k(a_i^\top v_k)| \geq \rho \epsilon_j t^{1/r}, \\
&\text{or } |\varphi_\ell(a_i^\top v_\ell)| \geq \rho t^{1/r} \text{ for some } \ell > k \Big).
\end{aligned}$$

Therefore, we have

$$\mathbb{P}(Z_i \geq \epsilon_j \rho^r t) \leq 2re^{-t^{2/r}/2} \text{ if } t \leq \rho^{-r} P_{j,k} \text{ and}$$
$$\mathbb{P}(Z_i \geq \epsilon_j \rho^r t) = 0 \text{ if } t > \rho^{-r} P_{j,k},$$

where we noted $P_{j,k} := \min\big\{2B/\epsilon_j, 2(B/B_k)R\big\}$. We will also make use of notation $j^*(k) := \lceil \log_2(R/B_k) \rceil$, so that

$$j \leq j^*(k) \Rightarrow P_{j,k} = 2B/\epsilon_j, \qquad j > j^*(k) \Rightarrow P_{j,k} = 2(B/B_k)R.$$

Fixing $j \geq 0$, $k \in [r]$, we write for any $\theta > 0$ (a specific θ will be chosen later):

$$\mathbb{E}\, e^{(\theta/n)\rho^{-r}[Z_i - \mathbb{E}Z_i]/\epsilon_j} = 1 + \left(\frac{\theta}{n}\right)^2 \mathbb{E}\Big[(\epsilon_j^{-1}\rho^{-r}(Z_i - \mathbb{E}Z_i))^2 F((\theta/n)(\epsilon_j^{-1}\rho^{-r}(Z_i - \mathbb{E}Z_i)))\Big],$$

where

$$F(x) := x^{-2}[e^x - x - 1] \leq e^{|x|}.$$

Thus using this bound and the inequality $xy \leq x^2 + y^2$:

$$\mathbb{E}\, e^{(\theta/n)\rho^{-r}[Z_i - \mathbb{E}Z_i]/\epsilon_j} \leq 1 + \left(\frac{\theta}{n}\right)^2 \left[\mathbb{E}((\epsilon_j^{-1}\rho^{-r}(Z_i - \mathbb{E}Z_i))^4 + \mathbb{E}e^{2(\theta/n)\rho^{-r}|Z_i - \mathbb{E}Z_i|/\epsilon_j}\right]. \tag{43}$$

By the sub-gaussian tail assumption, $\mathbb{E}(\epsilon_j^{-1}\rho^{-r}(Z_i - \mathbb{E}Z_i))^4$ is bounded by a constant κ_r dependent on r. We now assume that θ is such that

$$\frac{\theta}{n} \leq \min\left(\frac{(\rho^{-r}P_{j,k})^{2/r-1}}{8}, 1\right),$$

which is equivalent to having $(\theta/n)y \leq y^{2/r}/8$ for $y \in [0, \rho^{-r}P_{j,k}]$ and $r \geq 2$. Then, $\mathbb{E}e^{2(\theta/n)\rho^{-r}|Z_i - \mathbb{E}Z_i|/\epsilon_j}$ is also bounded by another constant κ_r' dependent on r. Indeed, by the sub-gaussian tail assumption, $|\mathbb{E}Z_i| \leq \rho^r \epsilon_j s_r$ for some r-dependent constant, and we can then use the fact that:

$$\mathbb{E}e^{\alpha X} = \int_0^\infty e^{kz}p(z)dz = \int_0^\infty \left(1 + \alpha\int_0^z e^{\alpha y}\right)p(z)dzdy = 1 + \alpha\int_0^\infty\int_y^\infty e^{\alpha y}dyp(z)dz = 1 + \alpha\int_0^\infty e^{\alpha y}p(X \geq y)dy,$$

with $\alpha = 2\theta/n$ and $X = \rho^{-r}|Z_i|\epsilon_j$ to get:

$$\begin{aligned}
\mathbb{E}\, e^{2(\theta/n)\rho^{-r}|Z_i - \mathbb{E}Z_i|/\epsilon_j} &\leq \mathbb{E}\, e^{2(\theta/n)\rho^{-r}(|Z_i| + |\mathbb{E}Z_i|)/\epsilon_j} \\
&\leq e^{2(\theta/n)s_r}\mathbb{E}e^{2\theta/n\rho^{-r}|Z_i|/\epsilon_j} \\
&\leq e^{2(\theta/n)s_r}[1 + \frac{2\theta}{n}\int_0^\infty e^{2(\theta/n)y}[\mathbb{P}(Z_i \geq y\rho^r\epsilon_j) + \mathbb{P}(-Z_i \geq y\rho^r\epsilon_j)]dy] \\
&\leq e^{2(\theta/n)s_r}[1 + \frac{2\theta}{n}2r\int_0^{\rho^{-r}P_{j,k}} e^{2(\theta/n)y - y^{2/r}/2}dy], \\
&\leq e^{2(\theta/n)s_r}[1 + \frac{2\theta}{n}2r\int_0^\infty e^{-y^{2/r}/4}dy] \\
&= e^{2(\theta/n)s_r}[1 + \frac{\theta}{n}c_r].
\end{aligned}$$

We finally use the fact that $\theta/n \leq 1$ to write $\mathbb{E}e^{2(\theta/n)\rho^{-r}|Z_i - \mathbb{E}Z_i|/\epsilon_j} \leq \kappa_r'$, with $\kappa_r' = e^{2s_r}[1 + c_r]$. We write $\kappa_r'' = \kappa_r + \kappa_r'$ and use Equation (43) together with the independence of the Z_i to obtain:

$$\mathbb{E}\, e^{\theta\rho^{-r}\left[\frac{1}{n}\sum_{i=1}^n Z_i - \mathbb{E}Z_i\right]/\epsilon_j} \leq \left(1 + \left(\frac{\theta}{n}\right)^2\kappa_r''\right)^n \leq e^{\frac{\theta^2}{n}\kappa_r''}.$$

Thus, using that $\mathbb{P}(X \geq y) = \mathbb{P}(e^X \geq e^y) \leq e^{-y}\mathbb{E}e^X$ (Markov Inequality), we have that for fixed $u_\ell\, u_\ell, v_\ell,\ \ell \in [r]$ in the suitable ϵ-nets is upper bounded for all $\theta \in [0, \min(n, n(\rho^{-r}P_{j,k})^{2/r-1}/8)]$ as:

$$\mathbb{P}\left(\frac{1}{n}\sum_{i\in[n]} Z_i - \mathbb{E}Z_i \geq \rho^r\epsilon_j t_{j,k}\right) \leq \exp\left((r+1)d\ln(1 + 2^{j+2}) - \theta t_{j,k} + \kappa_r''\theta^2/n\right). \tag{44}$$

We see in Equation (42) that the variables Z_i are built by fixing a specific either u_ℓ for $\ell < k$, v_ℓ for $\ell > k$, and u_k and v_k, meaning that there are actually $r+1$ variables to be fixed in nets of resolution either 2^{-j} or 2^{-j-1}. Note that all Z_i for $i \in \{1, \cdots, n\}$ are constructed with the same choice of u_ℓ and v_ℓ. Therefore, the number of possible choices for $u_\ell \in \mathcal{N}_j$ and $v_\ell \in \mathcal{N}_{j+1}$ involved in the definition of Z_i is upper-bounded by

$$|\mathcal{N}_{j+1}|^{r+1} \leq e^{d(r+1)\ln(1+2^{j+2})}.$$

Combining this with Equation (44), we obtain using a union bound that:

$$\begin{aligned}
\mathbb{P}\left(\sup_{u_\ell, v_\ell}\left\{Z - \mathbb{E}Z\right\} \geq \rho^r\epsilon_j t_{j,k}\right) &= \mathbb{P}\left(\cup_{u_\ell, v_\ell}\left\{Z - \mathbb{E}Z \geq \rho^r\epsilon_j t_{j,k}\right\}\right) \\
&\leq \sum_{u_\ell, v_\ell}\mathbb{P}\left(Z - \mathbb{E}Z \geq \rho^r\epsilon_j t_{j,k}\right) \\
&\leq \exp\left((r+1)d\ln(1 + 2^{j+2}) - \theta t_{j,k} + \kappa_r''\theta^2/n\right).
\end{aligned}$$

Let now $\theta_{j,k} = \min(n, n(\rho^{-r}P_{j,k})^{2/r-1}/8, \sqrt{nd})$, and

$$t_{j,k} = \kappa_r'' \frac{\theta_{j,k}}{n} + \frac{1}{\theta_{j,k}}[d(r+1)\ln(1+2^{j+2}) + \gamma + 2\ln(j+1)],$$

where $\gamma > 0$ is a free parameter. We then use the chaining decomposition of $Y = \sup_x \left\{ \sum_{j\geq 0, k\in[r]} Z - \mathbb{E}Z \right\}$, and another union bound on j and k to write that:

$$\begin{aligned}
\mathbb{P}\left(Y \geq \rho^r \sum_{j\geq 0, k\in[r]} \epsilon_j t_{j,k} \right) &= \mathbb{P}\left(\sup_x \left\{ \sum_{j\geq 0, k\in[r]} Z - \mathbb{E}Z \right\} \geq \rho^r \sum_{j\geq 0, k\in[r]} \epsilon_j t_{j,k} \right) \\
&\leq \mathbb{P}\left(\sum_{j\geq 0, k\in[r]} \sup_x \{Z - \mathbb{E}Z\} \geq \rho^r \sum_{j\geq 0, k\in[r]} \epsilon_j t_{j,k} \right) \\
&\leq \sum_{j\geq 0, k\in[r]} \mathbb{P}\left(\sup_{u_\ell, v_\ell} \{Z - \mathbb{E}Z\} \geq \rho^r \epsilon_j t_{j,k} \right) \\
&\leq \sum_{j\geq 0, k\in[r]} e^{-\gamma - 2\ln(1+j)}.
\end{aligned}$$

In the end, using that $\sum_{j\geq 1} j^{-2} = \pi^2/6$, we obtain:

$$\mathbb{P}\left(Y \geq \rho^r \sum_{j\geq 0, k\in[r]} \epsilon_j t_{j,k} \right) \leq r\frac{\pi^2}{6} e^{-\gamma}.$$

Moreover, one has

$$\epsilon_j t_{j,k} \leq \epsilon_j A_r (1+j)(d+\gamma)\left[\frac{(\rho^{-r}P_{j,k})^{1-2/r}}{n} + \frac{1}{\sqrt{nd}} + \frac{1}{n} \right],$$

for some suitable constant A_r dependent only on r. Fix some $k \in [r]$. Write:

$$\begin{aligned}
\frac{1}{A_r}\sum_{j\geq 0} \epsilon_j t_{j,k} &\leq 4\frac{d+\gamma}{\sqrt{nd}} + \frac{d+\gamma}{n} \sum_{j=0}^{j^*(k)} \epsilon_j (2\rho^{-r}B/\epsilon_j)^{1-2/r}(1+j) \\
&\quad + \frac{d+\gamma}{n} \sum_{j > j^*(k)} \epsilon_j (2\rho^{-r}BR/B_k)^{1-2/r}(1+j) \\
&\leq 4\frac{d+\gamma}{\sqrt{nd}} + \frac{d+\gamma}{n} A_r' (\rho^{-r}B)^{1-2/r} \left\{ 1 + (B_k/R)^{2/r} \ln(R/B_k) \right\},
\end{aligned}$$

where A_r' is another constant depending only on r. Since $B_k \leq R$, then $(B_k/R)^{2/r}\ln(R/B_k)$ is bounded by a (r-dependent) constant. □

We know present Corollary 1, which is a consequence of Theorem 7. We consider again i.i.d. a_i, bounded by R, satisfying the subgaussian tail assumption with parameter ρ, and some function φ that is 1-Lipschitz, and uniformly bounded by B_φ. Writing

$$H(x) = \frac{1}{n}\sum_{i=1}^n a_i a_i^\top \varphi(a_i^\top x), \tag{45}$$

We have the following corollary.

Corollary 1. *Thus for $1 > \delta > 0$, with probability at least $1-\delta$, it holds for some $C > 0$ that*

$$\sup_{x\in\mathcal{S}_1} \|H(x)\|_{op} \leq C'\rho^3(d + \ln(1/\delta) + \ln(5\pi^2/6))[(1+\rho^{-1}B_\varphi)/\sqrt{dn} + (1 + \{\rho^{-3}R^2B_\varphi\}^{1-2/3})/n]. \tag{46}$$

Proof. Let us write $\varphi_3(u) = \varphi(u) - \varphi(0)$. Then φ_3 satisfies our assumptions (1-Lipschitz, $\varphi_3(0) = 0$). Moreover, we can decompose matrix $H(x) - \mathbb{E}H(x)$ into $M(x) + N$, where

$$M(x) = \frac{1}{n}\sum_{i=1}^{n}\left[a_i a_i^\top \varphi_3(a_i^\top x) - \mathbb{E}a_1 a_1^\top \varphi_3(a_1^\top x)\right], \qquad N = \frac{1}{n}\sum_{i=1}^{n}\left[a_i a_i^\top \varphi(0) - \mathbb{E}a_1 a_1^\top \varphi(0)\right].$$

Taking $r = 2$, $\varphi_1 = \varphi_2 = Id$, the Theorem 7 gives us that

$$\mathbb{P}(\|N\|_{op} \geq C_2|\varphi(0)|\rho^2(\gamma + d)(1/\sqrt{dn} + 1/n)) \leq 2\frac{\pi^2}{6}e^{-\gamma}. \tag{47}$$

Taking next $r = 3$, and $B = R^2 B_\varphi$, we obtain

$$\mathbb{P}(\sup_{x\in\mathcal{S}_1}\|M(x)\|_{op} \geq C_3\rho^3(d+\gamma)(1/\sqrt{dn} + [\rho^{-3}R^2B_\varphi]^{1-2/3}/n)) \leq 3\frac{\pi^2}{6}e^{-\gamma}. \tag{48}$$

Combined, these two bounds give us that for all $\gamma > 0$, we have, setting $C = C_2 + C_3$:

$$\mathbb{P}(\sup_{x\in\mathcal{S}_1}\|H(x)\|_{op} \geq C'\rho^3(d+\gamma)[(1+\rho^{-1}B_\varphi)/\sqrt{dn} + (1 + \{\rho^{-3}R^2B_\varphi\}^{1-2/3})/n]) \leq 5\frac{\pi^2}{6}e^{-\gamma}. \tag{49}$$

We finally take $\gamma = -\ln\left(\frac{6\delta}{5\pi^2}\right)$. □

The last step required to prove Theorem 4 is to consider the supremum over $\mathcal{B}(0, D)$ with an arbitrary M_ℓ-Lipschitz function, which can be done by direct reduction:

Proof of Theorem 4. To apply this to ℓ'', defined on $\mathcal{B}(0, D)$ and M_ℓ Lipschitz, we apply Corollary 1 to $\varphi(x) = \frac{1}{M_\ell D}\ell''(Dx)$ (which is 1-Lipschitz on $\mathcal{B}(0, 1)$). Then, $B_\varphi = B_\ell/M_\ell D$ and the right hand side must be multiplied by $M_\ell D$. □

Remark 5. Note that there is a difference in the way Theorem 6 and Theorem 7 are applied to our linear models problem. In particular, Theorem 6 considers $\varphi_1 = \|\cdot\|^2$ and $\varphi_2 = \ell''$, whereas Theorem 7 uses $\varphi_1 = \varphi_2 = Id$ and $\varphi_3 = \ell''$. Theorem 6 can be adapted to work with $r = 3$, but the bound does not improve when splitting $\|\cdot\|^2$ into $Id \times Id$. Similarly, Theorem 7 could be used with $r = 2$ and $\varphi_1 = \|\cdot\|^2/(2R)$ (to respect the 1-Lipschitz assumption), but in this case the bound can only be worse since the main difference is that the ρ^3 factor becomes $R\rho^2$, and ρ is generally smaller than R.

B.4. Tightness of Theorem 7

Consider that the a_i uniformly distributed on the sphere with radius $R = \sqrt{d}$, and take for f_k the identity. Such vectors can be constructed by taking vectors A_i with coordinates $i.i.d.$ standard gaussian, and setting $a_i = \sqrt{d}\|A_i\|^{-1}A_i$. The subgaussianity parameter ρ can then be taken equal to 1.

Then, using known results about maximal correlation between variables with fixed marginals (e.g., Vershynin, 2019, Section 3), the expectation $\mathbb{E}\prod_{k=1}^{r} a_1^\top x_k$ is maximized, over choices $x_k \in \mathcal{S}_1$, by taking $x_1 = \cdots = x_r$. We may choose $x_1 = e_1$, the first unit vector, by rotational invariance, and thus the expectation is upper-bounded as:

$$\mathbb{E}\prod_{k=1}^{r} a_1^\top x_k \leq \mathbb{E}d^{r/2}\mathbb{E}\left[\frac{|A_i(1)|^r}{\|A_i\|^r}\right].$$

This is of order 1, as can be shown using concentration inequalities on the deviations of $\|A_i\|$ from $\sqrt{d}$. Consider then the empirical sum $\frac{1}{n}\sum_{i\in[n]}\prod_{k\in[r]} a_i^\top x_k$. Choose $x_k = d^{-1/2}a_1$ for all $k \in [r]$. Then this empirical sum evaluates to

$$\frac{1}{n}\sum_{i\in[n]}\prod_{k\in[r]} a_i^\top x_k = \frac{1}{n}d^{r/2} + \frac{1}{n}\sum_{i=2}^{n}\prod_{k\in[r]} a_i^\top a_1.$$

The second sum can be shown to be of order 1 (conditioning on a_1, and then using, *e.g.*, Bienaymé-Tchebitchev inequality). Thus, one cannot hope to establish concentration without extra assumptions on the data distribution unless $d^{r/2} = O(n)$.

Contrast this with the result of Theorem 7: for $R = \sqrt{d}$, $B = R^r$ and $\rho = 1$, it gives that $Y \leq O(d^{(r/2)(1-2/r)}d/n) = O(d^{r/2}/n)$. Thus the result is sharp for the particular example we just considered.

C. Experimental setting

Some implementation details are omitted in the main text due to lack of space. To ease the reader's understanding, we provide these details here, along with some additional experimental results. We also provide code for SPAG in supplementary material.

Optimization problem. We used the logistic loss with quadratic regularization, meaning that the function at node i is:

$$f_i : x \mapsto \frac{1}{m}\sum_{j=1}^{m} \log\left(1+\exp(-y_{i,j}x^\top a_j^{(i)})\right) + \frac{\lambda}{2}\|x\|^2,$$

where $y_{i,j} \in \{-1, 1\}$ is the label associated with $a_j^{(i)}$, the j-th sample of node i. The local datasets are constructed by shuffling the LibSVM datasets, and then assigning a fixed portion to each worker. Then, the server subsamples n points from its local dataset to construct the preconditioning dataset. To assess the suboptimality, we let the best algorithm run for more time in order to get a good approximation of the minimum error. Then, we subtract it to the running error of an algorithm to get the suboptimality at each step.

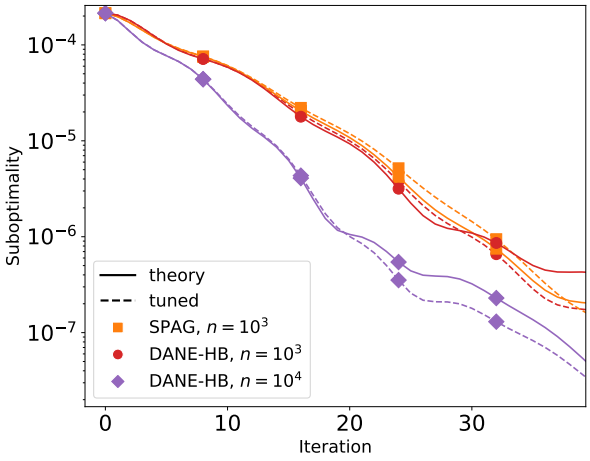
Adjusting α_t and β_t. We found that choosing $A_0 = 0$ and $B_0 = 1$ for SPAG is usually not the best choice. Indeed, rates are asymptotic and sequences α_t and β_t converge very slowly when $\sigma_{F/\phi}$ is small, whereas we typically rarely use more than about 100 iterations of SPAG. Therefore, we start the algorithm with A_{t_0} and B_{t_0} with $t_0 > 0$ instead. We used $t_0 = 50$, but SPAG is not very sensitive to this choice.

Tuning the momentum. Figure 5(a) evaluates the relevance of tuning the parameters controlling the momentum of SPAG and HB-DANE. To do so, we compare the default values of $\beta = (1-(1+2\mu/\lambda)^{-1/2})^2$ (for HB-DANE) and $\sigma_{F/\phi} = 1/(1+2\mu/\lambda)$ (for SPAG) to values obtained through a grid search on the KDD2010 dataset with $\lambda = 10^{-7}$. We tune HB-DANE by using a grid-search of resolution 0.05 to test the values between 0.5 and 1. For $n = 10^3$, theory predicts a momentum of $\beta = 0.86$ and the grid search gives $\beta = 0.85$. For $n = 10^4$, theory predicts $\beta = 0.81$ and the grid search gives $\beta = 0.8$. For SPAG, we test $\sigma_{F/\phi} = 10^{-2},\ 3\times 10^{-3},\ 10^{-3}$ and so on until $\sigma_{F/\phi} = 10^{-5}$ (so roughly divided by 3 at each step). For $n = 10^3$, theory predicts $\sigma_{F/\phi} = 0.005$ and the tuning yields $\sigma_{F/\phi} = 0.006$. For $n = 10^4$, theory predicts $\sigma_{F/\phi} = 0.0099$ and the grid-search leads to $\sigma_{F/\phi} = 0.01$. We do not display the curves in this case ($n = 10^4$) since they are nearly identical. Therefore, the grid-search always obtains the value on the grid that is closest to the theoretical value of the parameter, and the difference in practice is rather small, as can be seen in Figure 5(a). This is why we use default values in the main text.

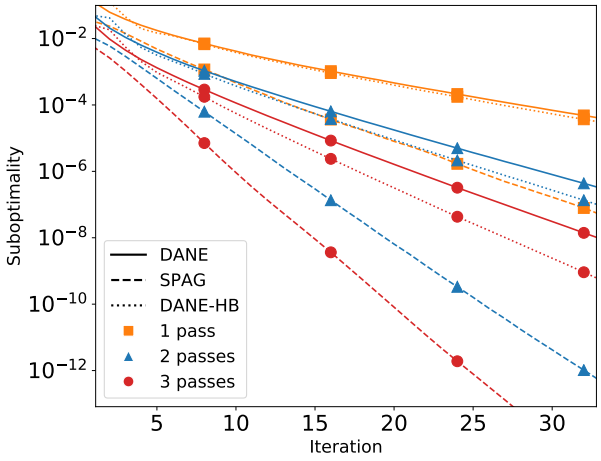
Local subproblems. Local problems are solved using a sparse implementation of SDCA (Shalev-Shwartz, 2016). In practice, the ill-conditionned regime is very hard, especially when μ is small. Indeed, the local subproblems are very hard to solve, and it should be beneficial to use accelerated algorithms to solve the inner problems. In our experiments, we warm-start the local problems (initializing on the solution of the previous one), and keep doing passes over the preconditioning dataset until $\|\nabla V_t(x_t)\| \leq 10^{-9}$ (checked at each epoch). This threshold is important because it greatly affects the performances of preconditioned gradient methods. Figure 5(b) compares the performances of SPAG, DANE and HB-DANE for different number of passes on the inner problems for the RCV1 dataset for $n = 10^4$ and $\lambda = 10^{-5}$. We use $\mu = 2\times 10^{-5}$ and a step-size of 1 for all algorithms. We first see that increasing the number of passes significantly improves the convergence speed of all algorithms. Besides, heavy-ball acceleration does not seem very efficient when local problems are not solved accurately enough. On the contrary, SPAG seems to enjoy faster rates than DANE nevertheless. It would be interesting to understand these different behaviours more in details.

Gain far from the optimum. So far, we have presented experiments with good initializations (solution for the local dataset), and argued why G_t was very small in this case. Because of Lemma 2, one would expect that G_t could be large when x_t is very far from x_*. Yet, We see in the proof of Lemma 2 that the Lipschitz constant of the Hessian only needs to be considered for any convex set that contains x_{t+1}, v_{t+1}, y_t and v_t. In the case of logistic regression, the third derivative decreases very fast when far from 0, meaning that the local Lipschitz constant of the Hessian is small when the iterates are far from 0. In other words, the Hessian changes slowly when far from the optimum (at least for logistic regression).

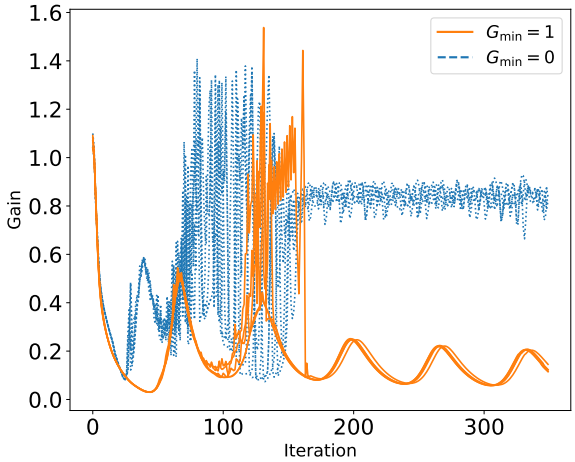
We believe that this is the reason why G_t can always be chosen of order 1 (smaller than 2) in our experiments, and that this holds regardless of the initialization. To support this claim, we plot in Figure 5(c) the values of the gain for the RCV1 dataset with $\lambda = 10^{-7}$ and 5 different x_0 sampled from $\mathcal{N}(0, 10^3)$, the normal law centered at 0 with variance 10^3. We use a step-size of 0.9 and $\mu = 2\times 10^{-5}$. We first see that for $G_{\min} = 1$, the gain is always very low, and actually increases at some point instead of becoming lower and lower, so the fact that we were able to choose G_t of order 1 in the other



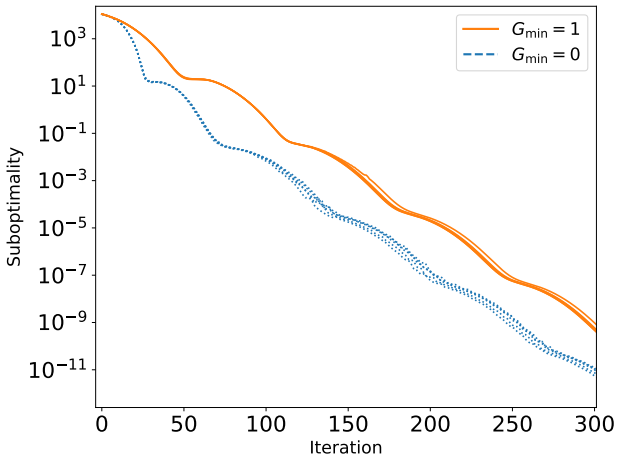
(a) Impact of parameters tuning (KDD2010, $\lambda = 10^{-7}$).



(b) Impact of inaccurate solving of the inner problems.



(c) Impact of $G_{\min}$ on the gain.



(d) Impact of $G_{\min}$ on the suboptimality.

Figure 5. Impact of several implementation details.

experiments is not linked to the good initialization. We had to choose a slightly higher μ than in the other experiments in order to satisfy the relative smoothness condition, which was not satisfied at each iteration otherwise. Since G_t is small in practice and the smaller the G_t the better the rate, we test SPAG with no minimum value for the gain G_t. The curve for the gain in this case is shown by $G_{\min} = 0$, and we see that the true gain stabilizes to a higher value, since updates are more agressive. We discuss the efficiency of this version in the next paragraph. Note that the oscillations are not due to numerical instability or inaccurate solving of the inner problems, but rather to the fact that the step-size is slightly too big so sometimes the smoothness inequality is not verified. Yet, this does not affect the convergence of SPAG, as shown in Figure 5(d).

Line Search with no minimum value. Since the gain is almost always smaller than 1, the line-search in SPAG generally only consists in checking that $G_t = 1$ works, which can be done locally. Therefore, there is no added communication cost. As discussed earlier, it is possible to allow $G_t < 1$ when performing line search, which makes SPAG slightly more adaptative at the cost of a few more line-search loops. Figure 5(d) presents the difference between SPAG using a line search with $G_{\min} = 0$ and $G_{\min} = 1$. The curves show the suboptimality for the runs used to generate Figure 5(c). Note that we omit the cost of line search in the iteration cost (we still count in terms of number of iterations, even though more communication rounds are actually needed when $G_{\min} = 0$). We see that setting $G_{\min} = 0$ is initially slightly faster but that the rate is very similar, so that using $G_{\min} = 0$ may slightly improve iteration complexity but is not worth doing in this case. Note that suboptimality curves for different initializations are almost indistinguishable, which can be explained by the fact that the quadratic penalty term dominates and that all initializations have roughly the same norm (since d is high).