OPTIMIZATION METHODS & SOFTWARE
https://doi.org/10.1080/10556788.2026.2647275

Taylor & Francis
Taylor & Francis Group

OPEN ACCESS Check for updates

Self-concordant smoothing in proximal quasi-Newton algorithms for large-scale convex composite optimization

Adeyemi D. Adeoye and Alberto Bemporad

IMT School for Advanced Studies Lucca, Lucca, Italy

ABSTRACT

We introduce a notion of *self-concordant smoothing* for minimizing the sum of two convex functions, one of which is smooth and the other nonsmooth. The key highlight is a natural property of the resulting problem's structure that yields a variable metric selection method and a step length rule especially suited to proximal quasi-Newton algorithms. Also, we efficiently handle specific structures promoted by the nonsmooth term, such as ℓ_1-regularization and group-lasso penalties. A convergence analysis for the class of proximal quasi-Newton methods covered by our framework is presented. In particular, we obtain guarantees, under standard assumptions, for two algorithms: `Prox-N-SCORE` (a proximal Newton method) and `Prox-GGN-SCORE` (a proximal generalized Gauss-Newton method). The latter uses a low-rank approximation of the Hessian inverse, reducing most of the cost of matrix inversion and making it effective for overparameterized machine learning models. Numerical experiments on synthetic and real data demonstrate the efficiency of both algorithms against state-of-the-art approaches. A Julia implementation is publicly available at https://github.com/adeyemiadeoye/SelfConcordantSmoothOptimization.jl.

ARTICLE HISTORY
Received 18 September 2024
Accepted 9 March 2026

KEYWORDS
Nonsmooth optimization; convex optimization; machine learning; regularization; self-concordant functions

2020 MATHEMATICS SUBJECT CLASSIFICATIONS
65K05; 90C06; 49M15

1. Introduction

We consider the composite optimization problem

$$\min_{x\in\mathbb{R}^n} \mathcal{L}(x) := f(x) + g(x), \tag{1}$$

where f is a smooth, convex loss function and g is a closed, proper, convex nonsmooth regularization function. Several optimization problems in engineering, machine learning, and finance can be written in the form (1), including sparse signal recovery, image processing, compressed sensing, and most classification and regression tasks in machine learning. Proximal gradient algorithms are among the most widely used methods for such problems (see [20] and the references therein for a comprehensive treatment). These algorithms handle the nonsmooth term g efficiently by employing its *proximal* operator, which

CONTACT Adeyemi D. Adeoye adeyemi.adeoye@imtlucca.it IMT School for Advanced Studies Lucca, Piazza San Francesco 19, 55100 Lucca, Italy

Supplemental data for this article can be accessed online at https://doi.org/10.1080/10556788.2026.2647275.

© 2026 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.
This is an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. The terms on which this article has been published allow the posting of the Accepted Manuscript in a repository by the author(s) or with their consent.

is typically assumed to be computable at low cost. Among other notable approaches is the *partial smoothing* framework of [9], which reassesses early *full* smoothing methods that iteratively replace g with a smooth approximation, e.g., [12,13], and instead smooths only one component of g while leaving the other unchanged. The setting they consider has the form $g(x) = \mathcal{R}(x) + \Omega(x)$, where $\mathcal{R}$ is typically a scaled ℓ_1-norm ($\beta\|x\|_1$) that promotes sparsity, and Ω encodes additional structure (group, fused, etc.). The main motivation remains to use the proximal operator of the (unchanged) nonsmooth component so that the intended structures are preserved. Notably, *fast* proximal gradient schemes [8,34,35,61] have become standard for solving such problems.

Although such first-order schemes outperform subgradient or bundle methods [9], they often yield only modest solution accuracy [45]. Incorporating second-order information typically improves both convergence speed and solution accuracy. The main drawback is the computational cost associated with full Hessian inverse. Several prior works [10,27,46,52,57,60] therefore incorporate approximate second-order information into proximal gradient schemes to mimic the performance of true proximal Newton methods. To ensure global convergence, many of these schemes rely on line search or trust-region procedures, which introduce additional computational cost. Some other works avoid such safeguards by imposing extra structure on the smooth term f. For instance, in the convex case [60] assumes that f is self-concordant; this assumption yields efficient step-length and correction rules but confines the approach to settings where these conditions hold (see [31,49]). In contrast, we propose a step-length selection rule specifically for proximal quasi-Newton methods; it is derived from a self-concordance-like structure inherent in our scheme and does not demand that f or g be self-concordant.

In particular, we *regularize*[1] problem (1) by a second smooth function g_s, resulting in the following problem:

$$\min_{x\in\mathbb{R}^n} \mathcal{L}_s(x) \coloneqq f(x) + g_s(x;\mu) + g(x), \tag{2}$$

where[2] g_s is a self-concordant, epi-smoothing function for g with a positive smoothing parameter μ (see Definition 3.3). By construction (see Section 3.1), the functions g and g_s do not conflict; therefore, efficient proximal schemes can be used to iteratively solve problem (2) and, for a suitable choice of μ, recover the solution of the original formulation (1) (see Sections 4 and 7). The smooth regularizer g_s serves two main algorithmic purposes in this work. First, it provides an adaptive step-length selection method analogous to the Newton-decrement framework but without requiring any self-concordance information about f. Second, its Hessian has a simple diagonal structure that can be exploited as a variable metric to scale the proximal operator of g efficiently. As a result, the regularization enhances both the solvability of the smooth part and the handling of the nonsmooth component.

While our development does not rely on any particular structure of g, Section 5 shows how known structures can be incorporated, extending the approach to a broader class of structured penalties. For lasso and multi-task regression with structured sparsity, we relate Nesterov's smoothing [35] to our framework and combine the 'prox-decomposition' property of g with the smoothness of g_s, thereby enabling straightforward treatment of such structures.

Most notably, three observations are vital to the development of our algorithmic framework:

(1) For many practical optimization problems, e.g., those that arise in modern machine learning, proximal Newton methods enjoy powerful convergence guarantees but are often computationally prohibitive. This motivates the use of proximal quasi-Newton schemes that use low-rank updates at each iteration (see Section 4.2).
(2) The infimal-convolution smoothing technique employed to construct g_s uncovers a structure that falls within the self-concordant regularization (SCORE) framework of [1,2]. Consequently, we can devise an efficient adaptive step-length rule for proximal quasi-Newton algorithms without requiring the original problem to be self-concordant. In other words, our development extends SCORE so that it accommodates nonsmooth regularizers while preserving problem-specific structure.
(3) The notion of *epi-smoothing functions* introduced in [14] permit a principled combination of the smooth regularizer g_s with proximal algorithms that handles the nonsmooth term g, assuming an efficient method exists for evaluating its proximal operator. Moreover, the diagonal Hessian of g_s serves as a natural variable metric for the resulting scheme, enabling efficient computation of the scaled proximal operator.

Burke and Hoheisel [14,15] developed the notion of *epi-smoothing* for studying several epigraphical convergence (*epi-convergence*) properties for convex composite functions by combining the infimal convolution smoothing framework due to Beck and Teboulle [9] with the idea of *gradient consistency* due to Chen [17]. The key variational analysis tool used throughout their development is the *coercivity* of the class of regularization kernels studied in [9]. In particular, they establish the close connection between epi-convergence of the regularization functions and supercoercivity of the regularization kernel. Then, based on the above observations, we synthesize this idea with the notion of *self-concordant regularization* [1,2] to propose two proximal-type algorithms, viz., `Prox-N-SCORE` (Algorithm 1) and `Prox-GGN-SCORE` (Algorithm 2), for convex composite minimization.

Paper organization. The rest of this paper is organized as follows: In Section 2, we present some notations and background on convex analysis. In Section 3, we establish our self-concordant smoothing notion with some properties and results. We describe our proximal quasi-Newton scheme in Section 4, and present the `Prox-N-SCORE` and `Prox-GGN-SCORE` algorithms. In Section 5, we describe an approach for handling specific structures promoted by the nonsmooth function g in problem (1), and propose a practical extension of the so-called *prox-decomposition* property of g for the self-concordant smoothing framework, which has certain in-built smoothness properties. Convergence properties of the `Prox-N-SCORE` and `Prox-GGN-SCORE` algorithms are studied in Section 6. In Section 7, we present some numerical simulation results for our proposed framework with an accompanying Julia package,[3] and compare the results with other state-of-the-art approaches. Finally, we give a concluding remark and discuss prospects for future research in Section 8.

2. Notation and preliminaries

We denote by $\bar{\mathbb{R}} \coloneqq \mathbb{R} \cup \{-\infty, +\infty\}$ the set of extended real numbers. The sets $\mathbb{R}_+ \coloneqq [0, +\infty[$ and $\mathbb{R}_{++} \coloneqq \mathbb{R}_+ \backslash \{0\}$, respectively, denote the set of nonnegative and positive real numbers. Let $g\colon \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be an extended real-valued function. The *(effective) domain* of g is given by $\operatorname{dom} g \coloneqq \{x \in \mathbb{R}^n \mid g(x) < +\infty\}$ and its *epigraph* (resp., *strict epigraph*) is given by $\operatorname{epi} g \coloneqq \{(x, \gamma) \in \mathbb{R}^n \times \mathbb{R} \mid g(x) \leq \gamma\}$ (resp., $\operatorname{epi_s} g \coloneqq \{(x, \gamma) \in \mathbb{R}^n \times \mathbb{R} \mid g(x) < \gamma\}$). Given $\gamma \in \mathbb{R}_{++}$, the γ-sublevel set of g is $\Gamma_\gamma(g) \coloneqq \{x \in \mathbb{R}^n \colon g(x) \leq \gamma\}$. The standard inner product between two vectors $x, y \in \mathbb{R}^n$ is denoted by $\langle \cdot, \cdot \rangle$, that is, $\langle x, y \rangle \coloneqq x^\top y$, where $x^\top$ is the transpose of x.

For an $n \times n$ matrix H, we write $H \succ 0$ (resp., $H \succeq 0$) to say H is positive definite (resp., positive semidefinite). If all the entries of H are zero, we say that H is null. The sets $\mathcal{S}^n_+$ and $\mathcal{S}^n_{++}$, respectively, denote the set of $n \times n$ symmetric positive semidefinite and symmetric positive definite matrices. The set $\{\operatorname{diag}(v) \mid v \in \mathbb{R}^n\}$, where $\operatorname{diag}\colon \mathbb{R}^n \to \mathbb{R}^{n \times n}$, defines the set of all diagonal matrices in $\mathbb{R}^{n \times n}$. Matrix I_d denotes the $d \times d$ identity matrix. We denote by $\operatorname{card}(\mathcal{G})$, the cardinality of a set $\mathcal{G}$. For any two functions f and g, we define $(f \circ g)(\cdot) \coloneqq f(g(\cdot))$. We denote by $\mathcal{C}^k(\mathbb{R}^n)$, the class of k-times continuously-differentiable functions on $\mathbb{R}^n$, $k \in \mathbb{R}_+$. If the p-th derivatives of a function $f \in \mathcal{C}^k(\mathbb{R}^n)$ is L_f-Lipschitz continuous on $\mathbb{R}^n$ with $p \leq k$, $L_f \in \mathbb{R}_+$, we write $f \in \mathcal{C}^{k,p}_{L_f}(\mathbb{R}^n)$. The notation $\|\cdot\|$ stands for the standard Euclidean (or 2-) norm $\|\cdot\|_2$. We define the weighted norm induced by $H \in \mathcal{S}^n_{++}$ by $\|x\|_H \coloneqq \langle Hx, x \rangle^{\frac{1}{2}}$, for $x \in \mathbb{R}^n$. An Euclidean ball of radius r centered at $\bar{x}$ is denoted by $\mathcal{B}_r(\bar{x}) \coloneqq \{x \in \mathbb{R}^n \mid \|x - \bar{x}\| \leq r\}$. Associated with a given $H \in \mathcal{S}^n_{++}$, the (Dikin) ellipsoid of radius r centred at $\bar{x}$ is defined by $\mathcal{E}_r(\bar{x}) \coloneqq \{x \in \mathbb{R}^n \mid \|x - \bar{x}\|_H \leq r\}$. We define the spectral norm $\|A\| \equiv \|A\|_2$ of a matrix $A \in \mathbb{R}^{m \times n}$ as the square root of the maximum eigenvalue of $A^\top A$, where $A^\top$ is the transpose of A.

A convex function $g\colon \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is said to be *proper* if $\operatorname{dom} g \neq \emptyset$. The function g is said to be lower semicontinuous (lsc) at y if $g(y) \leq \liminf_{x \to y} g(x)$; if it is lsc at every $y \in \operatorname{dom} g$, then it is said to be lsc on $\operatorname{dom} g$. We denote by $\Gamma_0(D)$ the set of proper convex lsc functions from $D \subseteq \mathbb{R}^n$ to $\mathbb{R} \cup \{+\infty\}$. Given $g \in \mathcal{C}^3(\operatorname{dom} g)$, we respectively denote by $g'(t)$, $g''(t)$ and $g'''(t)$ the first, second and third derivatives of g, at $t \in \mathbb{R}$, and by $\nabla_x g(x)$, $\nabla^2{}_x g(x)$, and $\nabla^3{}_x g(x)$ the gradient, Hessian and third-order derivative tensor of g, respectively, at $x \in \mathbb{R}^n$; if the variables with respect to which the derivatives are taken are clear from context, the subscripts are omitted. If $\nabla^2 g(x) \in \mathcal{S}^n_{++}$ for a given $x \in \mathbb{R}^n$, then the *local* norm $\|\cdot\|_x$ with respect to g at x is defined by $\|d\|_x \coloneqq \langle \nabla^2 g(x) d, d \rangle^{1/2}$, the weighted norm of d induced by $\nabla^2 g(x)$. The associated dual norm is denoted $\|v\|^\diamond_x \coloneqq \langle \nabla^2 g(x)^{-1} v, v \rangle^{1/2}$, for $v \in \mathbb{R}^n$. The subdifferential $\partial g\colon \mathbb{R}^n \to 2^{\mathbb{R}^n}$ of a proper function $g\colon \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is defined by $x \mapsto \left\{u \in \mathbb{R}^n \mid (\forall y \in \mathbb{R}^n)\ \langle y - x, u \rangle + g(x) \leq g(y)\right\}$, where $2^{\mathbb{R}^n}$ denotes the set of all subsets of $\mathbb{R}^n$. The function g is said to be subdifferentiable at $x \in \mathbb{R}^n$ if $\partial g(x) \neq \emptyset$; the subgradients of g at x are the members of $\partial g(x)$.

We define set convergence in the sense of Painlevé-Kuratowski. Let $\mathbb{N}$ denote the set of natural numbers. Let $\{C_k\}_{k \in \mathbb{N}}$ be a sequence of subsets of $\mathbb{R}^n$. The outer limit of $\{C_k\}_{k \in \mathbb{N}}$ is the set

$$\limsup_{k \to \infty} C_k \coloneqq \left\{x \in \mathbb{R}^n \mid \exists \{k_j\}_{j \in \mathbb{N}}, \exists \{x_j\}_{j \in \mathbb{N}}\ \forall j, x_k \in C_k, \{x_k\} \to x\right\},$$

and its inner limit is

$$\liminf_{k\to\infty} C_k \coloneqq \left\{x \in \mathbb{R}^n \mid \exists x_k \in C_k \colon \{x_k\} \to x, \forall k \in \mathbb{N}\right\}.$$

The limit C of $\{C_k\}_{k\in\mathbb{N}}$ exists if its outer and inner limits coincide, and we write

$$C = \lim_{k\to\infty} C_k \coloneqq \limsup_{k\to\infty} C_k = \liminf_{k\to\infty} C_k.$$

We say that a function $g\colon \mathbb{R}^n \to \mathbb{R}\cup\{+\infty\}$ is coercive if $\liminf_{\|x\|\to\infty} g(x) = +\infty$, and supercoercive if $\liminf_{\|x\|\to\infty} \frac{g(x)}{\|x\|} = +\infty$. The sequence $\{g_k\}$ of functions $g_k\colon \mathbb{R}^n \to \bar{\mathbb{R}}$ is said to epi-converge to the function $g\colon \mathbb{R}^n \to \bar{\mathbb{R}}$ if $\lim_{k\to\infty} \operatorname{epi} g_k = \operatorname{epi} g$; it is said to continuously converge to g if for all $x \in \mathbb{R}^n$ and $\{x_k\} \to x$, we have $\lim_{k\to\infty} g_k(x_k) = g(x)$; and it converges pointwise to g if for all $x \in \mathbb{R}^n$, $\lim_{k\to\infty} g_k(x) = g(x)$. Epi-convergence, continuous convergence, and pointwise convergence of $\{g_k\}$ to g are respectively denoted by $\text{e-}\lim g_k = g$ (or $g_k \xrightarrow{e} g$), $\text{c-}\lim g_k = g$ (or $g_k \xrightarrow{c} g$), and $\text{p-}\lim g_k = g$ (or $g_k \xrightarrow{p} g$).

The conjugate (or Fenchel conjugate, or Legendre transform, or Legendre-Fenchel transform) $g^\star\colon \mathbb{R}^n \to \mathbb{R}\cup\{+\infty\}$ of a function $g\colon \mathbb{R}^n \to \mathbb{R}\cup\{+\infty\}$ is the mapping $y \mapsto \sup_{x\in\mathbb{R}^n}\{\langle x, y\rangle - g(x)\}$, and its biconjugate is $g^{\star\star} = (g^\star)^\star$.

3. Self-concordant regularization

This section introduces the concept of self-concordant smoothing, which provides structures that can be exploited in composite optimization problems. We begin by presenting the definition of generalized self-concordant functions, as given in [56].

Definition 3.1 (Generalized self-concordant function on $\mathbb{R}$): A univariate convex function $g \in \mathcal{C}^3(\operatorname{dom} g)$, with $\operatorname{dom} g$ open, is said to be (M_g, ν)-generalized self-concordant, with $M_g \in \mathbb{R}_+$ and $\nu \in \mathbb{R}_{++}$, if

$$\left|g'''(t)\right| \le M_g g''(t)^{\frac{\nu}{2}}, \quad \forall\, t \in \operatorname{dom} g.$$

Definition 3.2 (Generalized self-concordant function on $\mathbb{R}^n$ of order ν): A convex function $g \in \mathcal{C}^3(\operatorname{dom} g)$, with $\operatorname{dom} g$ open, is said to be (M_g, ν)-generalized self-concordant of order $\nu \in \mathbb{R}_{++}$, with $M_g \in \mathbb{R}_+$, if $\forall x \in \operatorname{dom} g$

$$\left|\left\langle \nabla^3 g(x)[v]u, u\right\rangle\right| \le M_g \|u\|_x^2 \|v\|_x^{\nu-2} \|v\|^{3-\nu}, \quad \forall\, u, v \in \mathbb{R}^n,$$

where $\nabla^3 g(x)[v] \coloneqq \lim_{t\to 0}\{(\nabla^2 g(x+tv) - \nabla^2 g(x))/t\}$ is the third directional derivative of g.

Note that for an (M_g, ν)-generalized self-concordant function g defined on $\mathbb{R}^n$, the univariate function $\varphi\colon \mathbb{R} \to \mathbb{R}\cup\{+\infty\}$ defined by $\varphi(t) \coloneqq g(x+tv)$ is (M_g, ν)-generalized self-concordant for every $x, v \in \operatorname{dom} g$ and $x + tv \in \operatorname{dom} g$. This provides an alternative definition for the generalized self-concordant function on $\mathbb{R}^n$.

A key observation from Definitions 3.1 and 3.2 is the possibility to extend the theory beyond the case $\nu = 3$ and $u = \nu$ originally presented in [38]. This observation, for instance, allowed the authors in [4] to introduce a *pseudo* self-concordant framework, in which $\nu = 2$, for the analysis of logistic regression. In a recent development, the authors in [41] identified a new class of pseudo self-concordant functions and showed how these functions may be slightly modified to make them *standard* self-concordant (i.e. where $M_g = 2, \nu = 3, u = \nu$), while preserving desirable structures. With such generalizations, and stemming from the idea of *Newton decrement* [38], we propose new step-length selection techniques for proximal quasi-Newton methods from the self-concordant regularization framework of this section. We denote by $\mathcal{F}_{M_g,\nu}$ the class of (M_g, ν)-generalized self-concordant functions, with generalized self-concordant parameters $M_g \in \mathbb{R}_+$ and $\nu \in \mathbb{R}_{++}$.

Definition 3.3 (Self-concordant smoothing function): We say that the parameterized function $g_s\colon \mathbb{R}^n \times \mathbb{R}_{++} \to \mathbb{R}$ is a self-concordant smoothing function for $g \in \Gamma_0(\mathbb{R}^n)$ if the following two conditions are satisfied:

SC.1 $\text{e-}\lim_{\mu \downarrow 0} g_s(x;\mu) = g(x)$.
SC.2 $g_s(x;\mu) \in \mathcal{F}_{M_g,\nu}$.

We denote by $\mathcal{S}^{\mu}_{M_g,\nu}$ the set of self-concordant smoothing functions for a function $g \in \Gamma_0(\mathbb{R}^n)$, that is, $\mathcal{S}^{\mu}_{M_g,\nu} \coloneqq \left\{ g_s\colon \mathbb{R}^n \times \mathbb{R}_{++} \to \mathbb{R} \,|\, g_s \xrightarrow{e} g,\ g_s \in \mathcal{F}_{M_g,\nu} \right\}$.

3.1. Self-concordant regularization via infimal convolution

Next, we present key elements of smoothing through infimal convolution, which includes the Moreau-Yosida regularization process as a special case in defining the (scaled) proximal operator.

Definition 3.4 (Infimal convolution): Let g and h be two functions from $\mathbb{R}^n$ to $\mathbb{R} \cup \{+\infty\}$. The infimal convolution (or 'inf-convolution' or 'inf-conv')[4] of g and h is the function $g \square h\colon \mathbb{R}^n \to \bar{\mathbb{R}}$ defined by

$$(g \square h)(x) = \inf_{w \in \mathbb{R}^n} \left\{ g(w) + h(x - w) \right\}. \tag{3}$$

The infimal convolution of g with h is said to be *exact at* $x \in \operatorname{dom} g$ if the infimum (3) is attained. It is *exact* if it is exact at each $x \in \operatorname{dom} g$, in which case we write $g \boxdot h$. Of utmost importance about the inf-conv operation in this paper is its use in the approximation of a function $g \in \Gamma_0(\mathbb{R}^n)$; that is, the approximation of g by its infimal convolution with a member $h_\mu(\cdot)$ of a parameterized family $\mathcal{H} \coloneqq \{h_\mu \mid \mu \in \mathbb{R}_{++}\}$ of (regularization) kernels. In more formal terms, we recall the notion of inf-conv regularization in Definition 3.5. For $h \in \Gamma_0(\mathbb{R}^n)$ and $\mu \in \mathbb{R}_{++}$, we define the function $h_\mu\colon \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ by the *epi-multiplication* operation[5]

$$h_\mu(\cdot) \coloneqq \mu h\left(\frac{\cdot}{\mu}\right), \quad \mu \in \mathbb{R}_{++}. \tag{4}$$

Definition 3.5 (Inf-conv regularization): Let g be a function in $\Gamma_0(\mathbb{R}^n)$. Define

$$\mathcal{H} \coloneqq \left\{(x,w) \mapsto h_\mu(x-w) \mid \mu \in \mathbb{R}_{++}\right\}$$

a parameterized family of regularization kernels. The inf-conv regularization process of g with $h_\mu \in \mathcal{H}$ is given by $(g\square h_\mu)(x)$, for any $x \in \mathbb{R}^n$.

The operation of the inf-conv regularization generalizes the Moreau-Yosida regularization process in which case, $h_\mu(\cdot) = \|\cdot\|^2/(2\mu)$ or, with a scaled norm, $h_\mu(\cdot) = \|\cdot\|_Q^2/(2\mu)$ for some $Q \in \mathcal{S}_{++}^n$. The Moreau–Yosida regularization process provides the value function of the proximal operator associated with a function $g \in \Gamma_0(\mathbb{R}^n)$. This leads us to the definition of the scaled proximal operator.

Definition 3.6 (Scaled proximal operator): The scaled proximal operator of a function $g \in \Gamma_0(\mathbb{R}^n)$, written $\operatorname{prox}_{\alpha g}^Q(\cdot)$, for $\alpha \in \mathbb{R}_{++}$ and $Q \in \mathcal{S}_{++}^n$, is defined as the unique point in dom g that satisfies

$$(g\square\psi_\alpha)(x) = g(\operatorname{prox}_{\alpha g}^Q(x)) + \psi_\alpha(x - \operatorname{prox}_{\alpha g}^Q(x)),$$

where $\psi_\alpha(\cdot) \coloneqq \|\cdot\|_Q^2/(2\alpha)$. That is, $\operatorname{prox}_{\alpha g}^Q(x) \coloneqq \operatorname*{argmin}_{w\in\mathbb{R}^n}\{g(w) + \psi_\alpha(x-w)\}$.

A key property of the scaled proximal operator is its *nonexpansiveness*; that is, the property that (see, e.g., [47] and [60, Lemma 2])

$$\left\|\operatorname{prox}_{\alpha g}^Q(x) - \operatorname{prox}_{\alpha g}^Q(y)\right\|_Q \le \|x-y\|_{Q^{-1}}, \tag{5}$$

for all $x, y \in \mathbb{R}^n$.

In the sequel, we assume that the regularization kernel function h is of the form

$$h(x) = \sum_{i=1}^n \phi(x^{(i)}), \tag{6}$$

where ϕ is a univariate *potential function*. We are now left with the question of what properties we need to hold for ϕ such that $g\square h_\mu$ produces g_s satisfying the self-concordant smoothing conditions SC.1–SC.2. To this end, we assume that ϕ satisfies the following:

K.1 ϕ is supercoercive.
K.2 $\phi \in \mathcal{F}_{M_\phi,\nu}$.

Many functions that appear in different settings naturally exhibit the structures in conditions K.1–K.2. For example, the ones belonging to the class of *Bregman/Legendre functions* introduced by Bauschke and Borwein [5] (see also [21] for a related characterization of the class of *Bregman functions*). In the context of proximal gradient algorithms for solving (1), the recent paper [7] enlists these functions as satisfying the new descent lemma (a.k.a *descent lemma without Lipschitz gradient continuity*) which the paper introduced. We summarize examples of these regularization kernel functions on different domains

Table 1. Examples of regularization kernel functions for self-concordant smoothing, and their generalized self-concordant parameters M_ϕ and ν (see Definition 3.1).

$\phi(t)$	dom ϕ	M_ϕ	ν	Remark
$\frac{1}{p}\sqrt{1+p^2\lvert t\rvert^2}-1, p\in\mathbb{R}_{++}$	$\mathbb{R}$	2	2.6	$p=1$
$\frac{1}{2}\left[\sqrt{1+4t^2}-1+\log\left(\frac{\sqrt{1+4t^2}-1}{2t^2}\right)\right]$	$\mathbb{R}$	$2\sqrt{2}$	3	Ostrovskii & Bach [41]
$\frac{1}{2}t^2$	$\mathbb{R}$	0	3	'Energy'
$\frac{1}{p}\lvert t\rvert^p, p\in(1,2)$	$\mathbb{R}_+$	4	6	$p=1.5$
$\log(1+\exp(t))$	$\mathbb{R}$	1	2	'Logistic'
$t\log t-t$	$[0,+\infty]$	1	4	'Boltzmann–Shannon'
$\begin{cases}\frac{1}{2}(t^2-4t+3), & \text{if } t\le 1\\ -\log t, & \text{otherwise}\end{cases}$	$\mathbb{R}$	4	3	De Pierro & Iusem [21]

in Table 1. We extract practical examples on $\mathbb{R}$ for the smoothing of the 1-norm and the indicator functions.

Remark 3.1: Suppose that dom h is a nonempty bounded subset of $\mathbb{R}^n$, for example, if $\phi\in\Gamma_0(\text{dom }\phi)$, then since we have that $g\in\Gamma_0(\text{dom } g)$ is bounded below as it possesses a continuous affine minorant (in view of [6, Theorem 9.20]), the less restrictive condition that ϕ is coercive sufficiently replaces the condition K.1. In other words, the key convergence notion presented later holds similarly for the resulting function $g\square h_\mu$ in this case. Particularly, we get that $g\square h_\mu$ in this case is exact, finite-valued and locally Lipschitz continuous (see, e.g., [15, Proposition 3.6]) making it fit into our algorithmic framework.

Remark 3.2: Whenever the supercoercivity condition is difficult to check (and the condition in Remark 3.1 does not hold), two possibilities exist according to [15, Proposition 3.9]: (1) If $h\in\Gamma_0(\mathbb{R}^n)$ is such that $g\square h_\mu \xrightarrow{e} g$, and $g\in\Gamma_0(\mathbb{R}^n)$ is supercoercive, then h is necessarily supercoercive; (2) If, however, $g\in\Gamma_0(\mathbb{R}^n)$ is not supercoercive, then we can find some $h\in\Gamma_0(\mathbb{R}^n)$ that is not supercoercive but for which $g\square h_\mu \xrightarrow{e} g$.

In light of Remarks 3.1 and 3.2, our examples in Table 1 include both coercive and supercoercive functions. In either case, we have $\phi\in\mathcal{F}_{M_\phi,\nu}$. We keep the supercoercivity condition to emphasize other realizable properties of $g\square h_\mu$.

Examples. For some functions g and h_μ, there exists a closed form solution to $g\square h_\mu$. On the other hand, if one gets that $g\square h_\mu = g\boxdot h_\mu\in\Gamma_0(\mathbb{R}^n)$, e.g., as a result of Proposition 3.13(i), then knowing in this case that

$$g\square h_\mu = (g^\star + h_\mu^\star)^\star, \tag{7}$$

we can efficiently estimate $g\square h_\mu$ using *fast* numerical schemes (see, e.g. [30]). The structure of h implies g_s can be expressed in terms of a corresponding univariate function

$\varphi : \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ by defining $\varphi_s(t;\mu) \coloneqq (\varphi \square h_\mu)(t)$, and then

$$g_s(x;\mu) = \sum_{i=1}^{n} \varphi_s(x^{(i)};\mu).$$

In the following, we provide examples of such φ_s for some $\phi \in \mathcal{F}_{M_\phi,\nu}$.

Infimal convolution of $\|\cdot\|_1$ with h_μ. In the first two examples, we consider $g(x) = \|x\|_1$ (Figure 1).

Example 3.7: Let $p=1$ in $\phi(t) = \frac{1}{p}\sqrt{1+p^2|t|^2} - 1$, with dom $\phi = \mathbb{R}$. Then,

$$\varphi_s(t;\mu) = \frac{\mu^2 - \mu\sqrt{\mu^2+t^2} + t^2}{\sqrt{\mu^2+t^2}}.$$

Example 3.8: $\phi(t) = \frac{1}{2}[\sqrt{1+4t^2} - 1 + \log(\frac{\sqrt{1+4t^2}-1}{2t^2})]$, with dom $\phi = \mathbb{R}$:

$$\varphi_s(t;\mu) = \frac{\sqrt{\mu^2+4t^2}}{2} - \frac{\mu}{2}\left[1 + \log(2) - \log\left(\frac{2t - \sqrt{\mu^2+4t^2} + \mu}{t}\right)\right.$$
$$\left. - \log\left(\frac{2t + \sqrt{\mu^2+4t^2} - \mu}{t}\right)\right].$$

Infimal convolution of $\delta_C(x)$ with h_μ. In the next example, we consider $g(x) = \delta_C(x)$, where $C \coloneqq \{x \in \mathbb{R}^n \mid l \le x \le u\}$ and

$$\delta_C(x) \coloneqq \begin{cases} 0, & \text{if } x \in C, \\ +\infty, & \text{otherwise.} \end{cases}$$

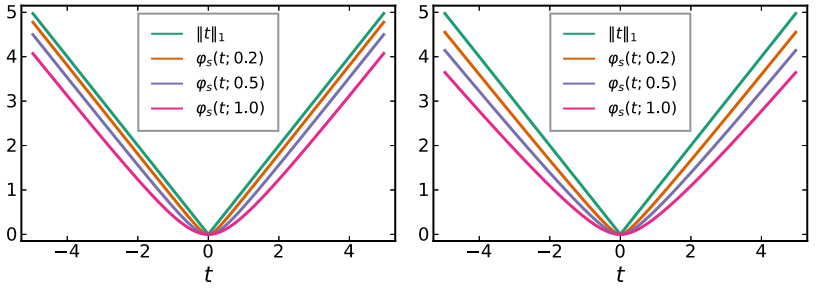

Figure 1. Generalized self-concordant smoothing of $\|\cdot\|_1$ with $\phi(t) = \sqrt{1+|t|^2} - 1$ (left) and $\phi(t) = \frac{1}{2}[\sqrt{1+4t^2} - 1 + \log(\frac{\sqrt{1+4t^2}-1}{2t^2})]$ (right). The smooth approximation is shown for $\mu = 0.2, 0.5, 1.0$.

Example 3.9: Let $g(x) = \delta_C(x)$, and consider

$$\phi(t) = \begin{cases} \frac{1}{2}(t^2 - 4t + 3), & \text{if } t \leq 1 \\ -\log t, & \text{otherwise,} \end{cases}$$

with $\operatorname{dom} \phi = \mathbb{R}$. We have

$$\varphi_s(t;\mu) = \begin{cases} \frac{1}{2\mu}\left(l - t + 3\mu\right)\left(l - t + \mu\right), & \text{if } l \geq t - \mu \\ \mu\log(\mu) - \mu\log(t - l), & \text{otherwise.} \end{cases}$$

The next two results characterize the functions h and h_μ defined by supercoercive and generalized self-concordant kernel functions.

Lemma 3.10: *Let $\phi \in \Gamma_0(\mathbb{R})$ be a function from $\mathbb{R}$ to $\mathbb{R} \cup \{+\infty\}$, and let the function $h\colon \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be defined by $h(x) \coloneqq \sum_{i=1}^n \lambda_i \phi(x^{(i)})$ with $x^{(i)} \in \operatorname{dom} \phi$, $\lambda_i \in \mathbb{R}_{++}$, $i = 1, 2, \ldots, n$. Then the following properties hold:*

(i) $h \in \Gamma_0(\mathbb{R}^n)$.
(ii) *h is supercoercive if and only if ϕ is supercoercive on its domain.*
(iii) *If $\phi \in \mathcal{F}_{M_\phi,\nu}$, where $M_\phi \in \mathbb{R}_+$ and $\nu \geq 2$, then $h(x)$ is well-defined on* $\operatorname{dom} h = \{\operatorname{dom} \phi\}^n$, *and $h(x) \in \mathcal{F}_{M_h,\nu}$, with $M_h \coloneqq \max\{\lambda_i^{1-\frac{\nu}{2}} M_\phi \mid 1 \leq i \leq n\} \in \mathbb{R}_+$.*

Proof: (i) This statement is a direct consequence of [6, Corollary 9.4, Lemma 1.27 and Proposition 8.17].
(ii) Follows directly from the definition of supercoercivity.
(iii) $h(\cdot) \in \mathcal{F}_{M_h,\nu}$ with $M_h \coloneqq \max\{\lambda_i^{1-\frac{\nu}{2}} M_\phi \mid 1 \leq i \leq n\} \in \mathbb{R}_+$ follows from [56, Proposition 1]. ■

Proposition 3.11 (Self-concordance of h_μ): *Suppose the conditions of Lemma 3.10 hold such that the function $h\colon \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ defined by (6) is (M_h, ν)-generalized self-concordant. Let $A \in \mathbb{R}^{n\times n}$ be a diagonal matrix defined by $A \coloneqq \operatorname{diag}(\frac{1}{\mu})$ such that $h(\frac{x}{\mu}) \equiv h(Ax)$ is an affine transformation of $h(x)$. Then the following properties hold:*

(i) *If $\nu \in (0, 3]$, then $h_\mu \in \mathcal{F}_{M,\nu}$ with $M = n^{\frac{3-\nu}{2}} \mu^{\frac{\nu}{2}-2} M_h$.*
(ii) *If $\nu > 3$, then $h_\mu \in \mathcal{F}_{M,\nu}$ with $M = \mu^{4-\frac{3\nu}{2}} M_h$.*

Proof: (i) We have $\|A\| = \frac{\sqrt{n}}{\mu}$. By [56, Proposition 2(a)], $h(\frac{x}{\mu}) \in \mathcal{F}_{M,\nu}$ with $M = \|A\|^{3-\nu} M_h$. In view of Lemma 3.10(iii), the scaling $h(\frac{\cdot}{\mu}) \mapsto \mu h(\frac{\cdot}{\mu})$ gives $M \mapsto \mu^{1-\frac{\nu}{2}} M$. The result follows.
(ii) The value $\mu^2 \in \mathbb{R}_{++}$ corresponds to the unique eigenvalue of $A^\top A$. By [56, Proposition 2(b)], $h(\frac{x}{\mu}) \in \mathcal{F}_{M,\nu}$ with $M = \mu^{3-\nu} M_h$. The result follows as in Item (i) above. ■

The next result concerns the epi-convergence of smoothing via infimal convolution under the condition of supercoercive regularization kernels in $\Gamma_0(\mathbb{R}^n)$.

Lemma 3.12 ([15, Theorem 3.8]): *Let $g, h \in \Gamma_0(\mathbb{R}^n)$ with h supercoercive and $0 \in \operatorname{dom} h$. Let h_μ be defined as in (4). Then the following hold:*

(i) $e\text{-}\lim_{\mu \downarrow 0}\{g^\star + \mu H_\star\} = g^\star$.
(ii) $e\text{-}\lim_{\mu \downarrow 0}\{g \square h_\mu\} = g$.
(iii) *If $h(0) \leq 0$, we have* $p\text{-}\lim_{\mu \downarrow 0}\{g \square h_\mu\} = g$.

The main argument for the notion of epi-convergence in optimization problems is that when working with functions that may take infinite values, it is necessary to extend traditional convergence notions by applying the theory of *set convergence* to epigraphs in order to adequately capture local properties of the function (through a resulting calculus of smoothing functions), which on the other hand may be challenging due to the *curse of differentiation* associated with nonsmoothness. We refer the interested reader to [48, Chapter 7] for further details on the notion of epi-convergence, and to [14,15,54] for extended results on epi-convergent smoothing via infimal convolution.

In the following, we highlight key properties of the infimal convolution of $g \in \Gamma_0(\mathbb{R}^n)$ with h_μ satisfying $h \in \mathcal{F}_{M_h,\nu}$.

Proposition 3.13: *Let $g, h \in \Gamma_0(\mathbb{R}^n)$. Suppose further that h is (M_h, ν)-generalized self-concordant and supercoercive, and define $g_s \coloneqq g \square h_\mu$ for all $\mu \in \mathbb{R}_{++}$. Then:*

(i) $g \square h_\mu = g \boxdot h_\mu \in \Gamma_0(\mathbb{R}^n)$.
(ii) *if $g^\star$ is such that $\nabla^2 g^\star[v]$ is null, it holds that $g_s \in \mathcal{S}^\mu_{M_g,\nu}$ with*

$$M_g = \begin{cases} n^{\frac{3-\nu}{2}} \mu^{\frac{\nu}{2}-2} M_h, & \text{if } \nu \in (0, 3], \\ \mu^{4-\frac{3\nu}{2}} M_h, & \text{if } \nu > 3. \end{cases}$$

(iii) *g_s is locally Lipschitz continuous.*

Proof: First, as an immediate consequence of [6, Lemma 1.28, Lemma 1.27 and Proposition 8.17], we have $h_\mu \in \Gamma_0(\mathbb{R}^n)$.

(i) Follows immediately from [6, Proposition 12.14].
(ii) By Item (i), $g_s = g \boxdot h_\mu \in \Gamma_0(\mathbb{R}^n)$. As a consequence of [6, Proposition 12.14], we have

$$g_s(x, \mu) = \min_{w \in \mathbb{R}^n} \left\{ g(w) + h_\mu(x - w) \right\},$$

and $g_s \xrightarrow{e} g$ (by [48, Theorem 11.34]). In view of [48, Proposition 7.2], for $x \in \operatorname{dom} g$ and

$$w_\mu(x) \in \operatorname*{argmin}_{w \in \mathbb{R}^n} \left\{ g(w) + h_\mu(x - w) \right\} \neq \emptyset,$$

$g_s \xrightarrow{e} g$ implies that $g_s(x,\mu) \to g(x)$ for at least one sequence $w_\mu(x) \to x$. Hence, we have

$$(g \square h_\mu)(x) = g(w_\mu(x)) + h_\mu(x - w_\mu(x)).$$

And, given $h \in \mathcal{F}_{M_h,\nu}$, we have by Proposition 3.11 that h_μ is (M_g,ν)-generalized self-concordant, where M_g is given by

$$M_g = \begin{cases} n^{\frac{3-\nu}{2}} \mu^{\frac{\nu}{2}-2} M_h, & \text{if } \nu \in (0,3], \\ \mu^{4-\frac{3\nu}{2}} M_h, & \text{if } \nu > 3. \end{cases}$$

Hence, $h_\mu \in \mathcal{C}^3(\operatorname{dom} g)$, and by [6, Proposition 18.7/Corollary 18.8], noting that higher-order derivatives are defined inductively in this sense [6, Definition 2.54, Remark 2.55], we deduce from the assumption on $g^\star$ that

$$\left|\left\langle \nabla^3 (g \square h_\mu)(x)[v]u, u \right\rangle\right| = \left|\left\langle \nabla^3 h_\mu(x - w_\mu(x))[v]u, u \right\rangle\right|, \quad \forall\, u, v \in \operatorname{dom} g,$$

and similarly for the second-order directional derivatives. By definition, the univariate function

$$\varphi(t) \coloneqq h_\mu(u_1 + t v_1), \tag{8}$$

is (M_g,ν)-generalized self-concordant, for every $u_1, v_1 \in \operatorname{dom} g$. That is, $\forall t \in \mathbb{R}$,

$$\left|\varphi'''(t)\right| \le M_g \varphi''(t)^{\frac{\nu}{2}},$$

which concludes the proof with $u_1 \equiv x$, $v_1 \equiv w(\frac{x}{\mu})$ and $t \equiv -\mu$ in (8).

(iii) Following the arguments in Items (i) and (ii) above, w_μ (and hence g_s) is finite-valued (see also [14, Lemma 4.2]). Then the Lipschitz continuity of g_s near some $\bar{x} \in \operatorname{dom} g$ follows from the convexity of g_s (see [48, Example 9.14]; see also [15, Proposition 3.6]).

■

4. A proximal quasi-Newton scheme

Our notion of self-concordant smoothing developed in the previous section is motivated by algorithmic purposes. Notably, we have discussed the epi-convergence of $g_s \in \mathcal{F}_{M_g,\nu}$ to $g \in \Gamma_0(\mathbb{R}^n)$ under suitable conditions, which plays a critical role in the optimization problem (2) in a global sense. We next characterize the optimal solution set of (2) using the notion of ε-optimality with respect to (1). We define ε-argmin $g \coloneqq \{x \mid g(x) \le \inf g + \varepsilon\}$ to be the set of points that minimize the function g up to a tolerance $\varepsilon \in \mathbb{R}_+$. For our approach, it suffices to state the following about the set of minimizers of g_s.

Proposition 4.1: *Fix any $\mu \in \mathbb{R}_{++}$. Suppose $g \in \Gamma_0(\mathbb{R}^n)$ and $g_s \in \mathcal{S}^\mu_{M_g,\nu}$. Then a minimizer of g_s is ε^μ-optimal for g with $\varepsilon^\mu \in \mathbb{R}_+$.*

Proof: From Proposition 3.13(iii), we have that, for any $\bar{x} \in \operatorname{dom} g$, $g_s \xrightarrow{e} g$ implies there is at least one sequence $w_\mu(\bar{x}) \to \bar{x}$. By the (super)coercivity of g_s, the level set $\{x \in \mathbb{R}^n \,|\, g_s(x;\mu) \le \hat{\alpha}\}$ at $\hat{\alpha} \in \mathbb{R}$ is bounded and contained in a compact set C such that $w_\mu(\bar{x}) \in C$. Let $w_\mu(\bar{x}) \in \varepsilon^\mu\text{-}\operatorname{argmin} g_s \subseteq C$ (with $\mu \in \mathbb{R}_{++}$ fixed). Then, since $g_s \xrightarrow{e} g$, we get from [48, Theorem 7.31(b)] that $g(\bar{x}) \le \inf g + \varepsilon^\mu$. Hence, $\bar{x} \in \varepsilon^\mu\text{-}\operatorname{argmin} g$. Finally, $w_\mu(\bar{x}) \in \varepsilon^\mu\text{-}\operatorname{argmin} g$ necessarily follows from [48, Theorem 7.33]. ■

Proposition 4.1, along with the observation in [48, Theorem 7.37], suggests that a proximal algorithm can provide a solution to (2), which also solves (1) with a high accuracy. Hence, the proximal method effectively handles the nonsmooth part of the problem, while our regularization approach enhances both the solvability of the smooth part of the original problem and improves the handling of the nonsmooth part through the choice of the variable metric. For the optimization problem (2), we assume the following:

P.1 f is convex and $f \in \mathcal{C}^{2,2}_{L_f}(\mathbb{R}^n)$.

P.2 $\rho_1 I_n \le \nabla^2 f(x^\star) \le L_1 I_n$, $\rho_2 I_n \le \nabla^2 g_s(x^\star) \le L_2 I_n$ at a locally optimal solution $x^\star$ of (2) with $L_1 \ge \rho_1 \in \mathbb{R}_{++}$ and $L_2 \ge \rho_2 \in \mathbb{R}_{++}$.

P.3 $g \in \Gamma_0(\mathbb{R}^n)$.

P.4 $g_s \in \mathcal{S}^\mu_{M_g,\nu}$.

In particular, we consider $g_s(x;\mu) \coloneqq g \square h_\mu$, where h is a suitable regularization kernel for self-concordant smoothing of g in the sense of Section 3.

Proximal quasi-Newton algorithms for solving (2) consist in minimizing a sequence of *upper approximation* of $\mathcal{L}_s$ obtained by summing the nonsmooth part $g(x_k)$ and a local quadratic model of the smooth part $q(x_k) \coloneqq f(x_k) + g_s(x_k)$ near x_k. That is, for $x \in \operatorname{dom} \mathcal{L} \equiv \operatorname{dom} f \cap \operatorname{dom} g$, we iteratively define

$$\hat{q}_k(x) \coloneqq q(x_k) + \langle \nabla q(x_k), x - x_k \rangle + \frac{1}{2}\|x - x_k\|^2_Q, \tag{9a}$$

$$\hat{m}_k(x) \coloneqq \hat{q}_k(x) + g(x), \tag{9b}$$

where $Q \in \mathcal{S}^n_{++}$, and then solve the subproblem

$$\delta_k \in \operatorname*{argmin}_{d \in \mathbb{R}^n} \hat{m}_k(x_k + d), \tag{10}$$

for a proximal quasi-Newton search direction δ_k. Our characterization of the optimality conditions for (2) in this section, particularly the flexibility in the choice of the variable metric Q, is well-motivated by the class of *cost approximation (CA) methods* [44]. This leads to a novel approach for selecting $\{x_k\}$ from the sequence of iterates $\{\delta_k\}$. The necessary optimality conditions for (2) are defined by

$$0 \in \nabla q(x^\star) + \partial g(x^\star), \tag{11}$$

for $x^\star \in \operatorname{dom} \mathcal{L}$. To find points $x^\star$ satisfying (11), CA methods, as the name implies, iteratively approximate $\nabla q(x_k)$ by a *cost approximating mapping* $\Phi\colon \mathbb{R}^n \to \mathbb{R}^n$, taking into

account the fixed approximation error term $\Phi(x_k) - \nabla q(x_k)$. That is, a point d is sought satisfying

$$0 \in \Phi(d) + \partial g(d) + \nabla q(x_k) - \Phi(x_k). \tag{12}$$

Let Φ be the gradient mapping of a continuously differentiable convex function $\psi : \mathbb{R}^n \to \mathbb{R}$. A CA method iteratively solves the subproblem

$$\min_{d\in\mathbb{R}^n} \left\{ \psi(d) + q(x_k) + g(d) - \psi(x_k) + \left\langle \nabla q(x_k) - \nabla \psi(x_k), d - x_k \right\rangle \right\}. \tag{13}$$

A step is then taken in the direction $\delta_k - x_k$, namely

$$x_{k+1} = x_k + \alpha_k(\delta_k - x_k), \tag{14}$$

where δ_k solves (13) and $\alpha_k \in \mathbb{R}_{++}$ is a step-length typically computed via a line search such that an appropriately selected *merit function* is sufficiently decreased along the direction $\delta_k - x_k$.

Remark 4.1: Evaluating the merit function too many times can be impractical. One way to mitigate this issue for large-scale problems is to incorporate 'predetermined step-lengths' into the solution scheme of (13). This allows us to update x_k as $x_{k+1} \equiv \delta_k$. However, methods that use this approach do not generally yield a monotonically decreasing sequence of objective values. Instead, convergence is characterized by a metric that measures the distance from iteration points to the set of optimal solutions [43].

We discuss next a new proximal quasi-Newton scheme that compromises between minimizing the objective values and decreasing the distance from iteration points to the set of optimal solutions as specified by a curvature-exploiting variable metric.

4.1. Variable metric and adaptive step-length selection

A very nice feature of the CA framework is that it can help, for instance, through the specific choice of Φ, to efficiently utilize the original problem's structure—a practice which is particularly useful when solving medium- to large-scale problems. This feature fits directly into our self-concordant smoothing framework. We notice that (13) gives (10) with the following choice of ψ:

$$\psi(\cdot) = \frac{1}{2}\|\cdot\|_Q^2, \quad Q \in \mathcal{S}_{++}^n. \tag{15}$$

In this case, the optimality conditions and our assumptions give

$$(Q - \nabla q)(x_k) \in (Q + \partial g)(d), \tag{16}$$

which leads to

$$\delta_k = \operatorname{prox}_g^Q(x_k - Q^{-1}\nabla q(x_k)). \tag{17}$$

In the proximal quasi-Newton scheme, Q may be the Hessian of $q(x_k)$ or its (low-rank) approximation. Although a diagonal structure of Q is often desired in this case due to its

ease of implementation, we most likely discard relevant curvature information, especially when q is not assumed to be separable. Our consideration in this work entails the following characterization of the optimality conditions:

$$(H_k - \nabla q)(x_k) \in (Q_k + \partial g)(d), \tag{18}$$

where H_k may be the Hessian, $\nabla^2 q(x_k) \equiv H_k^f + H_k^g$, of q or its approximation, where $H_k^f \equiv \nabla^2 f(x_k)$, $H_k^g \equiv \nabla^2 g_s(x_k;\mu)$, and $Q_k \in \mathcal{S}_{++}^n$. Specifically, we set $Q_k = H_k^g$ in (18) and propose the following step update formula:

$$x_{k+1} = \operatorname{prox}_{\alpha_k g}^{H_k^g}(x_k - \bar{\alpha}_k H_k^{-1} \nabla q(x_k)), \tag{19}$$

where $\bar{\alpha}_k \in \mathbb{R}_{++}$ results from *damping* the quasi-Newton steps.

Algorithm 1 `Prox-N-SCORE` (A proximal Newton algorithm)

Require: $x_0 \in \mathbb{R}^n$, problem functions f, g, self-concordant smoothing function $g_s \in \mathcal{S}_{M_g,\nu}^{\mu}$, $\alpha \in (0,1]$

1: **for** $k = 0, \ldots$ **do**
2: $\quad \text{grad}_k \leftarrow \nabla f(x_k) + \nabla g_s(x_k)$
3: $\quad H_k^g \leftarrow \nabla^2 g_s(x_k)$; $\eta_k \leftarrow \left\| \nabla g_s(x_k) \right\|_{H_k^{g^{-1}}}$ ▷ Note: H_k^g is diagonal
4: $\quad \bar{\alpha}_k = \frac{\alpha}{1+M_g \eta_k}$
5: $\quad H_k \leftarrow \nabla^2 f(x_k) + H_k^g$; Solve for Δ_k: $H_k \Delta_k = \text{grad}_k$
6: $\quad x_{k+1} \leftarrow \operatorname{prox}_{\alpha g}^{H_k^g}(x_k - \bar{\alpha}_k \Delta_k)$
7: **end for**

The validity of this procedure in the present scheme may be seen in the interpretation of the proximal operator $\operatorname{prox}_g(x^+)$ for some $x^+ \in \operatorname{dom} g$ as compromising between minimizing the function g and staying close to x^+ (see [42, Chapter 1]). When scaled by, say, H_k^g, 'closeness' is quantified in terms of the metric induced by H_k^g, and we want the proximal steps to stay close (as much as possible) to the Newton iterates relative to, say, $\|\cdot\|_{H_k^g}$. To see this, we note that in view of the fixed-point characterization (13) via CA methods, we may interpret proximal quasi-Newton algorithms as a fixation of the error term $\nabla\psi - \nabla q$ at some point in $\operatorname{dom} q \cap \operatorname{dom} g$. Let us fix some $\bar{x} \in \operatorname{dom} q \cap \operatorname{dom} g$ and introduce the operator $E_{\bar{x}}$ defined by

$$E_{\bar{x}}(z) \coloneqq \nabla^2 q(\bar{x}) z - \bar{\alpha} \nabla q(z), \tag{20}$$

where $0 < \bar{\alpha} \leq \alpha \leq 1$. Set $Q = Q_k \in \mathcal{S}_{++}^n$ arbitrary in (15). We aim to exploit the structure in g_s (and $\nabla^2 g_s$), so we define an operator $\xi_{\bar{x}}(Q_k, \cdot)$ to quantify the error between $\nabla^2 g_s$ and Q_k as follows:

$$\xi_{\bar{x}}(Q_k, z) \coloneqq (\nabla^2 g_s(\bar{x}) - Q_k)(z - x_k). \tag{21}$$

We provide a local characterization of the optimality conditions for (13) in terms of $E_{\bar{x}}$ and $\xi_{\bar{x}}$ in the next result.

Proposition 4.2: *Let the operators $E_{\bar{x}}$ and $\xi_{\bar{x}}(Q_k,\cdot)$ be defined by (20) and (21), respectively. Then the optimality conditions for (13) with $\psi(\cdot)=\frac{1}{2}\|\cdot\|^2_{Q_k}$ are locally characterized in terms of $E_{\bar{x}}$ and $\xi_{\bar{x}}(Q_k,\cdot)$ by*

$$E_{\bar{x}}(x_k)+\xi_{\bar{x}}(Q_k,d)\in\nabla^2 g_s(\bar{x})d+\alpha\partial g(d). \tag{22}$$

More precisely, (18) holds with $Q_k=\nabla^2 g_s(\bar{x})$ whenever $\bar{x}$ is the unique optimizer satisfying (22) at a local solution d of (13).

Proof: As g_s satisfies the property in SC.1, it holds that [14, Lemma 3.4]

$$\limsup_{\substack{x\to\bar{x}\\ \mu\downarrow 0}}\nabla g_s(x;\mu)=\partial g(\bar{x}). \tag{23}$$

Hence, by Lemma 3.12 and [48, Theorem 13.2], there exists $v_g\in\mathbb{R}^n$, in the *extended sense* of differentiability (see [48, Definition 13.1]), such that

$$\limsup_{\substack{x\to\bar{x}\\ \mu\downarrow 0}}\nabla g_s(x)=\partial g(\bar{x})=\{v_g\}, \tag{24a}$$

$$\emptyset\neq\partial g(d)\subset v_g+\nabla^2 g_s(\bar{x})(d-\bar{x})+o(\|d-\bar{x}\|)\mathcal{E}_r(\bar{x}). \tag{24b}$$

Let x_k be in some neighbourhood of $\bar{x}$ and let $\{x_k\}\to\bar{x}$ be generated by an iterative process. By assumption, the differentiable terms in (24b) are convex and the differential operators are monotone. It then holds that

$$\partial g(d)\subset v_g+\nabla^2 g_s(\bar{x})(d-x_k)+o(\|d-\bar{x}\|)\mathcal{E}_r(\bar{x}), \tag{25}$$

for all x_k in the neighbourhood of $\bar{x}$. Since differentiability in the extended sense is necessary and sufficient for differentiability in the *classical sense* (see [48, Definition 13.1 and Theorem 13.2]), it holds for some $\mu\in\mathbb{R}_{++}$ that $v_g\equiv\nabla g_s(\bar{x})$ which is defined through:

$$\nabla g_s(d)=\nabla g_s(\bar{x})+\nabla^2 g_s(\bar{x})(d-\bar{x})+o(\|d-\bar{x}\|). \tag{26}$$

Consequently, using (12) (with $\Phi=\nabla\psi$), and defining the Dikin ellipsoid $\mathcal{E}_r(\bar{x})$ in terms of g_s for r small enough, we deduce from (25), (26) that $Q_k(x_k-d)+\nabla^2 g_s(\bar{x})(x_k-d)-\bar{\alpha}\nabla q(x_k)\in\bar{\alpha}\nabla g_s(\bar{x})$ for $0<\bar{\alpha}\leq 1$. We assert $\nabla^2 f(\bar{x})(d-\bar{x})\in\mathcal{E}_r(\bar{x})$ at a local solution d of (13), and then deduce again from (25), (26) that $\bar{\alpha}\nabla g_s(\bar{x})+\nabla^2 g_s(\bar{x})(d-x_k)+\nabla^2 f(\bar{x})x_k\in\alpha\partial g(d)$ holds for $0<\bar{\alpha}\leq\alpha\leq 1$ near $\bar{x}$, whenever $\bar{x}$ is the unique solution $x^\star$ of (2). As a result, using $q:=f+g_s$, we get

$$(\nabla^2 q(\bar{x})-\bar{\alpha}\nabla q)x_k-\nabla^2 g_s(\bar{x})x_k\in Q_k(d-x_k)+\alpha\partial g(d). \tag{27}$$

In terms of $E_{\bar{x}}$ and $\xi_{\bar{x}}(Q_k,\cdot)$, (27) may be written as (22), which exactly gives (18) with the choice $Q_k=\nabla^2 g_s(\bar{x})$. ■

We consider *damping* the quasi-Newton steps such that

$$\bar{\alpha}_k = \frac{\alpha_k}{1 + M_g \eta_k}, \tag{28}$$

where M_g is given by P.4 and $\eta_k \coloneqq \left\| \nabla g_s(x_k) \right\|_{x_k}^{\diamond}$ is the dual norm of $\nabla g_s(x_k)$ with respect to $g_s(x_k)$. Note that the above choice for $\bar{\alpha}_k$, in the context of minimizing generalized self-concordant functions, assumes $\nu \geq 2$ (see, e.g. [56, Equation 12]). Suppose for example $\alpha_k = 1$ is fixed and $\nu = 3$, then (27) leads to the standard damped-step proximal quasi-Newton method in the framework of Newton decrement (cf. [56,60]).

By (6), H_k^g has a desirable diagonal structure and hence can be cheaply updated from iteration to iteration. This structure provides an efficient way to compute the scaled proximal operator $\operatorname{prox}_g^{H_k^g}$, e.g., via the proximal calculus presented in [11] (see Section 7 for two practical examples). Overall, by exploiting the structure of the problem, precisely

(i) taking adaptive steps that properly capture the curvature of the objective functions, and
(ii) scaling the proximal operator of g by a variable metric H_k^g which has a simple, diagonal structure,

we can adapt to an affine-invariant structure due to the quasi-Newton steps and ensure we remain close to them towards convergence.

If we choose $H_k \equiv \nabla^2 q(x_k)$ in (19), we obtain a proximal Newton step (see Algorithm 1):

$$x_{k+1} = \operatorname{prox}_{\alpha_k g}^{H_k^g}(x_k - \bar{\alpha}_k \nabla^2 q(x_k)^{-1} \nabla q(x_k)). \tag{29}$$

However, H_k may be any approximation of the Hessian of q at x_k. In view of (22), this corresponds to replacing the Hessian term $\nabla^2 q(\bar{x})$ in (20) by the approximating matrix evaluated at $\bar{x}$.

Algorithm 2 `Prox-GGN-SCORE` (A proximal generalized Gauss-Newton algorithm)

Require: $x_0 \in \mathbb{R}^n$, problem functions f, g, self-concordant smoothing function $g_s \in \mathcal{S}^{\mu}_{M_g,\nu}$, model $\mathcal{M}$, input-output pairs $\{u^{(i)}, y^{(i)}\}_{i=1}^m$ with $y^{(i)} \in \mathbb{R}^{n_y}$, $\alpha \in (0, 1]$

for $k = 0, \ldots$ **do**
  $H_k^g \leftarrow \nabla^2 g_s(x_k)$; $\eta_k \leftarrow \left\| \nabla g_s(x_k) \right\|_{H_k^{g^{-1}}}$ ▷ Note: H_k^g is diagonal
  $\bar{\alpha}_k \leftarrow \frac{\alpha}{1+M_g \eta_k}$
  if $m + n_y \leq n$ **then**
    Compute δ_k^{ggn} via (34)
  else
    Compute δ_k^{ggn} via (33)
  end if
  $x_{k+1} \leftarrow \operatorname{prox}_{\alpha g}^{H_k^g}(x_k + \bar{\alpha}_k \delta_k^{\text{ggn}})$
end for

4.2. A proximal generalized Gauss-Newton algorithm

In describing the proximal GGN algorithm, consider first the simple case $g \equiv 0$. Then (19) with $\bar{\alpha}_k = 1$ gives exactly the pure Newton direction

$$\delta_k^{\text{ggn}} = -H_k^{-1} \nabla q(x_k). \tag{30}$$

Now suppose that the function f quantifies a data-misfit or loss between the outputs[6] $\hat{y}^{(i)}$ of a model $\mathcal{M}(\cdot; x)$ and the expected outputs $y^{(i)}$, for $i = 1, 2, \ldots, m$, as in a typical machine learning problem, and that $g \neq 0$. Precisely, let $\hat{y}^{(i)} \coloneqq \mathcal{M}(u^{(i)}; x)$, and suppose that f can be written as

$$f(x) = \sum_{i=1}^{m} \ell(y^{(i)}, \hat{y}^{(i)}), \tag{31}$$

where $\ell \colon \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is a loss function. Define an 'augmented' Jacobian matrix $J_k \in \mathbb{R}^{(m+1)\times n}$ by [1,2]

$$J_k^\top \coloneqq \begin{bmatrix} \nabla_{x_k} \hat{y}^{(1)} & \nabla_{x_k} \hat{y}^{(2)} & \cdots & \nabla_{x_k} \hat{y}^{(m)} & \nabla g_s(x_k) \end{bmatrix}. \tag{32}$$

Then GGN approximation of the Newton direction (30) gives

$$\delta_k^{\text{ggn}} = -(H_k^f + H_k^g)^{-1} \nabla q \approx -(J_k^\top V_k J_k + H_k^g)^{-1} J_k^\top e_k, \tag{33}$$

where the vector $e_k \coloneqq [l'_{\hat{y}^{(1)}}(y^{(1)}, \hat{y}^{(1)}), \ldots, l'_{\hat{y}^{(m)}}(y^{(m)}, \hat{y}^{(m)}), 1]^\top \in \mathbb{R}^{m+1}$ defines an augmented 'residual' term, and $V_k \coloneqq \operatorname{diag}(v_k)$, with $v_k \coloneqq [l''_{\hat{y}^{(1)}}(y^{(1)}, \hat{y}^{(1)}), \ldots, l''_{\hat{y}^{(m)}}(y^{(m)}, \hat{y}^{(m)}), 0]^\top \in \mathbb{R}^{(m+1)}$. If $m+1 < n$ (possibly $m \ll n$), that is, when the model is *overparameterized*, the following equivalent formulation of (33) provides a computationally efficient way to compute the GGN search direction [1,2]:

$$\delta_k^{\text{ggn}} = -H_k^{g^{-1}} J_k^\top (I_m + V_k J_k H_k^{g^{-1}} J_k^\top)^{-1} e_k. \tag{34}$$

Note that in case the function g (and hence g_s) is scaled by some (nonnegative) constant, only the identity matrix I_m may be scaled accordingly. Now using $H_k \equiv J_k^\top V_k J_k + H_k^g$ in (19) gives the proximal GGN update (see Algorithm 2):

$$x_{k+1} = \operatorname{prox}_{\alpha_k g}^{H_k^g}(x_k + \bar{\alpha}_k \delta_k^{\text{ggn}}), \tag{35}$$

where $\bar{\alpha}_k$ is as defined in (28).

5. Structured penalties

As we have noted, more general nonsmooth problems impose certain structures on the variables that must be handled explicitly by the algorithm. Such situations arise in some Lasso and multi-task regression problems, where problem (1) takes the form

$$\min_{x \in \mathbb{R}^n} f(x) + \underbrace{\mathcal{R}(x) + \Omega(Cx)}_{g(x)}, \tag{36}$$

where, in addition to $\mathcal{R}(x)$, the function (cf. [18,19])

$$\Omega(Cx) \coloneqq \max_{u\in\mathcal{Q}} \langle u, Cx\rangle, \tag{37}$$

enforces a desired structure of the solution estimates. Here $C : \mathbb{R}^n \to \mathbb{V}$ is a linear map into a finite-dimensional vector space $\mathbb{V}$, and $\mathcal{Q} \subseteq \mathbb{V}^\star$ is a closed, convex subset of the dual space $\mathbb{V}^\star$.

For example, in the sparse group lasso problem [23,51], $\Omega(Cx) = \gamma \sum_{j\in\mathcal{G}} \omega_j \|x^{(j)}\|$ induces group level sparsity on the solution estimates and $\mathcal{R}(x) = \beta\|x\|_1$ promotes the overall sparsity of the solution, so that the optimization problem is written as

$$\min_{x\in\mathbb{R}^n} f(x) + \beta\|x\|_1 + \beta_{\mathcal{G}} \sum_{j\in\mathcal{G}} \omega_j \|x^{(j)}\|, \tag{38}$$

where $\beta \in \mathbb{R}_{++}$, $\beta_{\mathcal{G}} \in \mathbb{R}_{++}$, $\mathcal{G} = \{j_k, \ldots, j_{n_g}\}$ is the set of variables groups with $n_g = \text{card}(\mathcal{G})$, $x^{(j)} \in \mathbb{R}^{n_j}$ is the subvector of x corresponding to variables in group j and $\omega_j \in \mathbb{R}_{++}$ is the group penalty parameter. Another example is the graph-guided fused lasso for multi-task regression problems [26], where the function $\Omega(Cx) = \beta_{\mathcal{G}} \sum_{e=(r,s)\in E, r<s} \tau(\omega_{rs}) \left| x^{(r)} - \text{sign}(\omega_{rs}) x^{(s)} \right|$ encourages a fusion effect over variables $x^{(r)}$ and $x^{(s)}$ shared across tasks through a graph $G \equiv (V, E)$ of relatedness, where $V = \{1, \ldots, n\}$ denotes the set of nodes and E the edges; $\beta_{\mathcal{G}} \in \mathbb{R}_{++}$, $\tau(\omega_{rs})$ is a fusion penalty function, and $\omega_{rs} \in \mathbb{R}$ is the weight of the edge $e = (r, s) \in E$. Here, with $\mathcal{R}(x) = \beta\|x\|_1$, $\beta \in \mathbb{R}_{++}$, the optimization problem is written as

$$\min_{x\in\mathbb{R}^n} f(x) + \beta\|x\|_1 + \beta_{\mathcal{G}} \sum_{e=(r,s)\in E, r<s} \tau(\omega_{rs}) \left| x^{(r)} - \text{sign}(\omega_{rs}) x^{(s)} \right|. \tag{39}$$

In both examples, C is defined so as to encode these additional structures. See Section 7.2 for an illustration involving the sparse group lasso.

5.1. Structure reformulation for self-concordant smoothing

The key observation in problems of the form (36) is that the function $\Omega(Cx)$ belongs to the class of nonsmooth convex functions that is well-structured for Nesterov's smoothing [35] in which a smooth approximation Ω_s of Ω has the form[7]

$$\Omega_s(Cx; \mu) = \max_{u\in\mathcal{Q}} \left\{ \langle u, Cx\rangle - \mu d(u) \right\}, \quad \mu \in \mathbb{R}_{++}, \tag{40}$$

where d is a *prox-function*[8] of the set $\mathcal{Q}$. Note that Nesterov's smoothing approach assumes the knowledge of the exact structure of C. In the sequel, we shall write $\Omega^C(x) \equiv \Omega(Cx)$ or $\Omega_s^C(x; \mu) \equiv \Omega_s(Cx)$, with the superscript '$C$' to indicate the function is *structure-aware* via C.

Proposition 5.1: *Let C: $\mathbb{R}^n \to \mathbb{R}^n$ be a linear map and let ω be a continuous convex function defined on a closed and convex set $Q \subseteq \text{dom}\ \omega \subseteq \mathbb{R}^n$. Further, define*

$$\tilde{\Omega}(x) \coloneqq \max_{u\in\mathcal{Q}} \left\{ \langle u, Cx\rangle - \omega(u) \right\},$$

and let $d \coloneqq h^\star$, *where* $h\colon \mathbb{R}^n \to \mathbb{R}$ *satisfies* $\nabla^2 h \in \mathcal{S}^n_{++}$ *and is of the form* (6) *with* ϕ *satisfying* K.1–K.2 *so that* $h \in \mathcal{F}_{M_h,\nu}$ *with* $\nu \in [3,6)$ *if* $n > 1$ *and with* $\nu \in (0,6)$ *if* $n = 1$. *Then the function*

$$\Omega_s(x;\mu) = \max_{u\in\mathcal{Q}} \left\{ \langle u, Cx\rangle - \omega(u) - \mu d(u) \right\}, \quad \mu \in \mathbb{R}_{++}, \tag{41}$$

is a self-concordant smoothing function for $\tilde{\Omega}(x)$.

Proof: We follow the approach in [9, Section 4]. First note that we can write $\tilde{\Omega}(x) = \Omega(Cx)$, where

$$\Omega \coloneqq (\omega + \delta_Q)^\star.$$

Now, let $\tilde{d} \coloneqq d + \delta_Q$. In view of [56, Proposition 6], we have $d, \tilde{d} \in \mathcal{F}_{M_d,\nu_d}$ where $M_d = M_h$ and $\nu_d = 6 - \nu$. Next, define $\tilde{h} \coloneqq (\tilde{d})^\star$. We have

$$\begin{aligned}(\Omega^\star + \tilde{h}^\star_\mu)^\star(x) &= (\omega + \delta_Q + \mu\tilde{d})^\star(x)\\ &= \max_{u\in\mathcal{Q}} \left\{ \langle u, x\rangle - \omega(u) - \mu d(u) \right\},\end{aligned}$$

which is precisely $(\tilde{\Omega}\square h^\star_\mu)(x)$ according to [9, Theorem 4.1(a)] (cf. (7)). Now, since $d \coloneqq h^\star \in \mathcal{F}_{M_d,\nu_d}$, the result follows from Proposition 3.11 and Proposition 3.13(ii). ■

Under the assumptions of Proposition 5.1, $\Omega^C_s(x;\mu)$ provides a self-concordant smooth approximation of $\Omega(x)$ with $\mathbb{V} \equiv \mathbb{R}^n$. In this case, $\omega = 0$ in Proposition 5.1 and the prox-function d in (40) is given by $h^\star$, the dual of $h \in \mathcal{F}_{M_h,\nu}$.

5.2. Prox-decomposition and smoothness properties

An important property of the function $g = \mathcal{R} + \Omega^C$ we want to infer here is its prox-decomposition property [63] in which the (unscaled) proximal operator of g satisfies

$$\operatorname{prox}_g = \operatorname{prox}_{\Omega^C} \circ \operatorname{prox}_{\mathcal{R}}. \tag{42}$$

Under our assumptions on g and h, this property extends for the inf-conv regularization (and hence the self-concordant smoothing framework) .[9] To see this, let $\mathbb{V} \equiv \mathbb{R}^n$, and note the following equivalent expression for the definition of inf-convolution (3):

$$(\mathcal{R}\square h_\mu)(x) = \inf_{\substack{(u,v)\in\mathbb{R}^n\times\mathbb{R}^n\\ u+v=x}} \left\{ \mathcal{R}(u) + h_\mu(v) \right\}.$$

Define also the function $r_s\colon \mathbb{R}\times\mathbb{R} \to \mathbb{R}$ such that

$$(\mathcal{R}\square h_\mu)(x) \equiv \sum_{i=1}^n r_s(x^{(i)};\mu).$$

The next result follows, highlighting what we propose as the *inf-decomposition* property.

Proposition 5.2: *Let $g \in \Gamma_0(\mathbb{R}^n)$ be given as the sum $g(x) = \mathcal{R}(x) + \Omega^C(x)$. Suppose that the function $h \in \Gamma_0(\mathbb{R}^n)$ is supercoercive and define $z \coloneqq [r_s(x^{(1)};\mu), \ldots, r_s(x^{(n)};\mu)]^\top$. Then the regularization process $(g\square h_\mu)(x)$, for all $\mu \in \mathbb{R}_{++}$, is given by the composition*

$$(g\square h_\mu)(x) = (\Omega^C \square h_\mu)(z). \tag{43}$$

Proof: The exactness of the inf-conv regularization process by Proposition 3.13(i) allows to infer

$$\begin{aligned}(\Omega^C\square h_\mu)(z) &= \inf_{\substack{(u,v)\in\mathbb{R}^n\times\mathbb{R}^n\\ u+v=z}} \left\{\Omega^C(u) + h_\mu(v)\right\}\\ &= \inf_{\substack{(u,v)\in\mathbb{R}^n\times\mathbb{R}^n\\ 2u+v=x}} \left\{\mathcal{R}(u) + \Omega^C(u) + h_\mu(v)\right\}\\ &= ((\mathcal{R}+\Omega^C)\square h_\mu)(x) = (g\square h_\mu)(x).\end{aligned}$$

■

Given the smoothness properties of $\Omega^C\square h_\mu$ and $\mathcal{R}\square h_\mu$, we can apply the chain rule to obtain the derivatives of their composition $g\square h_\mu$. Precisely, [55, Lemma 2.1] provides sufficient conditions for the validity of the derivatives obtained via the chain rule for composite functions, which are indeed satisfied for $g\square h_\mu$ by our assumptions.

6. Convergence analysis

We analyse the convergence of Algorithms 1 and 2 under the proposed smoothing framework. In view of the numerical examples considered in Section 7, we restrict our analysis to the case $2 \le \nu \le 3$. However, similar convergence properties are expected to hold for the general case $\nu \in \mathbb{R}_{++}$, as the key bounds describing generalized self-concordant functions hold similarly for all of these cases (see, e.g., the Section 2 and concluding remark of [56]). We define the following metric term, taking the local norm $\|\cdot\|_x$ with respect to g_s:

$$d_\nu(x,y) \coloneqq \begin{cases} M_g\|y-x\| & \text{if } \nu = 2,\\ \left(\dfrac{\nu}{2}-1\right) M_g \|y-x\|_2^{3-\nu}\|y-x\|_x^{\nu-2} & \text{if } \nu > 2.\end{cases} \tag{44}$$

We introduce the notations $H_\star^g \equiv \nabla^2 g_s(x^\star)$, $H_\star^f \equiv \nabla^2 f(x^\star)$ and $H_\star \equiv \nabla^2 q(x^\star)$. Recall also the notations $H_k^g \equiv \nabla^2 g_s(x_k)$, $H_k^f \equiv \nabla^2 f(x_k)$ and $H_k \equiv \nabla^2 q(x_k)$ at x_k. Furthermore, we define the following matrices associated with any given twice differentiable function f:

$$\Sigma_f^{x,y} \coloneqq \int_0^1 \left(\nabla^2 f(x+\tau(y-x)) - \nabla^2 f(x)\right) \mathrm{d}\tau, \tag{45a}$$

$$\Upsilon_f^{x,y} \coloneqq \nabla^2 f(x)^{-1/2}\,\Sigma_f^{x,y}\,\nabla^2 f(x)^{-1/2}. \tag{45b}$$

We begin by stating some useful preliminary results. The following result provides bounds on the function g_s in (2).

Lemma 6.1 ([56, Proposition 10]): *Suppose that* P.3–P.4 *hold. Then, given any* $x, y \in \operatorname{dom} g$, *we have*

$$\omega_\nu(-d_\nu(x,y))\|y-x\|_x^2 \le g_s(y) - g_s(x) - \langle \nabla g_s(x), y-x\rangle \le \omega_\nu(d_\nu(x,y))\|y-x\|_x^2, \quad (46)$$

in which, if $\nu > 2$, *the right-hand side inequality holds if* $d_\nu(x,y) < 1$, *and*

$$\omega_\nu(\tau) \coloneqq \begin{cases} \dfrac{\exp(\tau)-\tau-1}{\tau^2} & \text{if } \nu = 2, \\ \dfrac{-\tau-\ln(1-\tau)}{\tau^2} & \text{if } \nu = 3, \\ \dfrac{(1-\tau)\ln(1-\tau)+\tau}{\tau^2} & \text{if } \nu = 4, \\ \left(\dfrac{\nu-2}{4-\nu}\right)\dfrac{1}{\tau}\left[\dfrac{\nu-2}{2(3-\nu)\tau}\left((1-\tau)\dfrac{2(3-\nu)}{2-\nu}-1\right)-1\right] & \text{otherwise.} \end{cases} \quad (47)$$

The next two lemmas are instrumental in our convergence analysis, and are immediate consequences of the (local) Hessian regularity of the smooth functions f and g_s in (2).

Lemma 6.2 ([37, Lemma 1.2.4]): *For any given* $x, y \in \operatorname{dom} f$, *we have*

$$\left\|\nabla f(y) - \nabla f(x) - \nabla^2 f(x)(y-x)\right\| \le \frac{L_f}{2}\|y-x\|^2, \quad (48)$$

$$\left|f(y) - f(x) - \langle \nabla f(x), y-x\rangle - \frac{1}{2}\langle \nabla^2 f(x)(y-x), y-x\rangle\right| \le \frac{L_f}{6}\|y-x\|^3. \quad (49)$$

Lemma 6.3 ([56, Lemma 2]): *For any given* $x, y \in \operatorname{dom} g$, $\Upsilon_{g_s}^{x,y}$ *satisfies*

$$\|\Upsilon_{g_s}^{x,y}\| \le R_\nu(d_\nu(x,y))d_\nu(x,y),$$

where, for $\tau \in [0,1)$, $R_\nu(\tau)$ *is defined by*

$$R_\nu(\tau) \coloneqq \begin{cases} \left(\dfrac{3}{2}+\dfrac{\tau}{3}\right)\exp(\tau) & \text{if } \nu = 2, \\ \dfrac{1-(1-\tau)^{\frac{4-\nu}{\nu-2}} - \left(\frac{4-\nu}{\nu-2}\right)\tau\,(1-\tau)^{\frac{4-\nu}{\nu-2}}}{\left(\frac{4-\nu}{\nu-2}\right)\tau^2\,(1-\tau)^{\frac{4-\nu}{\nu-2}}} & \text{if } \nu \in (2,3]. \end{cases} \quad (50)$$

Global convergence. We establish a global convergence result for the proximal quasi-Newton scheme (19). Specifically, we show that the iterates produced by this scheme decrease the objective function value in (1) when the step lengths are chosen according to (28) with $\alpha_k \in (0,1]$. Consequently, global convergence follows.

Let us define the following mapping:

$$G_{\alpha_k g}(x_k) \coloneqq \frac{1}{\bar{\alpha}_k} H_k\left(x_k - \operatorname{prox}_{\alpha_k g}^{H_k^g}(x_k - \bar{\alpha}_k H_k^{-1}\nabla q(x_k))\right). \quad (51)$$

Clearly, (19) is equivalent to

$$x_{k+1} = x_k - \bar{\alpha}_k H_k^{-1} G_{\alpha_k g}(x_k). \tag{52}$$

Using (18) with $Q_k = H_k^g$ and the definition of the (scaled) proximal operator, $G_{\alpha_k g}(x_k)$ satisfies

$$G_{\alpha_k g}(x_k) \in \nabla q(x_k) + \partial g(x_k - \bar{\alpha}_k H_k^{-1} G_{\alpha_k g}(x_k)). \tag{53}$$

Moreover, $G_{\alpha_k g}(\bar{x}) = 0$ if and only if $\bar{x}$ solves problem (2).

Proposition 6.4: *Suppose that* P.1, P.3 *and* P.4 *hold for* (2). *Let* $\{x_k\}$ *be the sequence generated by scheme* (19) *for problem* (2) *and satisfying* $\omega_v(d_v(x_{k+1}, x_k)) \le \frac{1}{2}$, *where* ω_v *and* d_v *are respectively defined by* (47) *and* (44). *Define* $\varepsilon_k^{\mu}(y) \coloneqq (L_f/6)\|y - x_k\|^3$, *and let* $\bar{\alpha}_k$ *be specified by* (28) *with* $\alpha_k \in (0, 1]$. *Then* $\{x_k\}$ *satisfies*

$$\mathcal{L}(x_{k+1}) \le \mathcal{L}(x_k) - \varepsilon_k^{\mu}(x_{k+1}). \tag{54}$$

***Proof*:** Letting $y = x_k - \bar{\alpha}_k H_k^{-1} G_{\alpha_k g}(x_k)$ and $x = x_k$ in Lemma 6.2, where $G_{\alpha_k g}$ is defined by (51), we have

$$\begin{aligned} f(x_{k+1}) \le & f(x_k) - \bar{\alpha}_k (H_k^{-1} \nabla f(x_k))^{\top} G_{\alpha_k g}(x_k) + \frac{\bar{\alpha}_k^2}{2} \left\| H_k^{-1} G_{\alpha_k g}(x_k) \right\|_{H_k^f}^2 \\ & + \frac{\bar{\alpha}_k^3 L_f}{6} \left\| H_k^{-1} G_{\alpha_k g}(x_k) \right\|^3. \end{aligned} \tag{55}$$

Using $\mathcal{L}(x_{k+1}) \coloneqq f(x_{k+1}) + g(x_{k+1})$ and (55), we get

$$\begin{aligned} \mathcal{L}(x_{k+1}) \le & f(x_k) - \bar{\alpha}_k (H_k^{-1} \nabla f(x_k))^{\top} G_{\alpha_k g}(x_k) + \frac{\bar{\alpha}_k^2}{2} \left\| H_k^{-1} G_{\alpha_k g}(x_k) \right\|_{H_k^f}^2 \\ & + \frac{\bar{\alpha}_k^3 L_f}{6} \left\| H_k^{-1} G_{\alpha_k g}(x_k) \right\|^3 + g(x_k - \bar{\alpha}_k H_k^{-1} G_{\alpha_k g}(x_k)) \\ \overset{\text{Lemma 6.2}}{\le} & f(z) - \langle \nabla f(x_k), z - x_k \rangle - \frac{1}{2}\|z - x_k\|_{H_k^f}^2 + \frac{L_f}{6}\|z - x_k\|^3 \\ & - \bar{\alpha}_k (H_k^{-1} \nabla f(x_k))^{\top} G_{\alpha_k g}(x_k) + \frac{\bar{\alpha}_k^2}{2} \left\| H_k^{-1} G_{\alpha_k g}(x_k) \right\|_{H_k^f}^2 \\ & + \frac{\bar{\alpha}_k^3 L_f}{6} \left\| H_k^{-1} G_{\alpha_k g}(x_k) \right\|^3 + g(x_k - \bar{\alpha}_k H_k^{-1} G_{\alpha_k g}(x_k)). \end{aligned} \tag{56}$$

In the above, we used the lower bound in Lemma 6.2 on $f(z)$. By the convexity of g, we have $g(z) - g(x_{x_{k+1}}) \ge v^{\top}(z - x_{k+1})$ for all $v \in \partial g(x_{k+1})$. Now since from (53), we have $G_{\alpha_k g}(x_k) - \nabla q(x_k) \in \partial g(x_k - \bar{\alpha}_k H_k^{-1} G_{\alpha_k g}(x_k))$, and noting that $\nabla q - \nabla f = \nabla g_s$, (56) gives

$$\mathcal{L}(x_{k+1}) \le f(z) + g(z) - \langle \nabla f(x_k), z - x_k \rangle - \frac{1}{2}\|z - x_k\|_{H_k^f}^2 + \frac{L_f}{6}\|z - x_k\|^3$$

$$
\begin{aligned}
&- \bar{\alpha}_k (H_k^{-1} \nabla f(x_k))^\top G_{\alpha_k g}(x_k) + \frac{\bar{\alpha}_k^2}{2} \left\| H_k^{-1} G_{\alpha_k g}(x_k) \right\|_{H_k^f}^2 \\
&+ \frac{\bar{\alpha}_k^3 L_f}{6} \left\| H_k^{-1} G_{\alpha_k g}(x_k) \right\|^3 \\
&- (G_{\alpha_k g}(x_k) - \nabla q(x_k))^\top (z - x_k + \bar{\alpha}_k H_k^{-1} G_{\alpha_k g}(x_k)) \\
&\leq \mathcal{L}(z) - \langle \nabla f(x_k), z - x_k \rangle - \frac{1}{2} \|z - x_k\|_{H_k^f}^2 - \bar{\alpha}_k (H_k^{-1} \nabla f(x_k))^\top G_{\alpha_k g}(x_k) \\
&+ \frac{\bar{\alpha}_k^2}{2} \left\| H_k^{-1} G_{\alpha_k g}(x_k) \right\|_{H_k^f}^2 + \frac{L_f}{6} \|z - x_k\|^3 + \frac{\bar{\alpha}_k^3 L_f}{6} \left\| H_k^{-1} G_{\alpha_k g}(x_k) \right\|^3 \\
&- G_{\alpha_k g}(x_k)^\top (z - x_k) - \frac{\bar{\alpha}_k^2}{2} \langle H_k^{-1} G_{\alpha_k g}(x_k), G_{\alpha_k g}(x_k) \rangle \\
&- \nabla q(x_k)^\top (z - x_k + \bar{\alpha}_k H_k^{-1} G_{\alpha_k g}(x_k)) \\
&= \mathcal{L}(z) + G_{\alpha_k g}(x_k)^\top (x_k - z) + \frac{\bar{\alpha}_k^2}{2} \langle H_k^{-1} (H_k^f H_k^{-1} - I_n) G_{\alpha_k g}(x_k), G_{\alpha_k g}(x_k) \rangle \\
&+ \nabla g_s(x_k)^\top (z - x_k) + \bar{\alpha}_k (H_k^{-1} \nabla g_s(x_k))^\top G_{\alpha_k g}(x_k) - \frac{1}{2} \|z - x_k\|_{H_k^f}^2 \\
&+ \frac{L_f}{6} \|z - x_k\|^3 + \frac{\bar{\alpha}_k^3 L_f}{6} \left\| H_k^{-1} G_{\alpha_k g}(x_k) \right\|^3, \qquad (57)
\end{aligned}
$$

where the second inequality results from the fact that $\langle H_k^{-1} G_{\alpha_k g}(x_k), G_{\alpha_k g}(x_k) \rangle \in \mathbb{R}_+$ and $\bar{\alpha}_k \geq \bar{\alpha}_k^2$ for $0 < \bar{\alpha}_k \leq 1$. Now set $z = x_k$ in (57) and use the following relations from (52):

$$
\bar{\alpha}_k H_k^{-1} G_{\alpha_k g}(x_k) = x_k - x_{k+1}, \quad G_{\alpha_k g}(x_k) = \frac{1}{\bar{\alpha}_k} H_k (x_k - x_{k+1}).
$$

We get

$$
\begin{aligned}
\mathcal{L}(x_{k+1}) &\leq \mathcal{L}(x_k) + \frac{\bar{\alpha}_k^2}{2} \langle H_k^{-1} (H_k^f H_k^{-1} - I_n) G_{\alpha_k g}(x_k), G_{\alpha_k g}(x_k) \rangle \\
&\quad + \bar{\alpha}_k (H_k^{-1} \nabla g_s(x_k))^\top G_{\alpha_k g}(x_k) + \frac{\bar{\alpha}_k^3 L_f}{6} \left\| H_k^{-1} G_{\alpha_k g}(x_k) \right\|^3 \\
&= \mathcal{L}(x_k) - \Bigg[\langle \nabla g_s(x_k), x_{k+1} - x_k \rangle + \frac{1}{2} \langle H_k^g (x_{k+1} - x_k), x_{k+1} - x_k \rangle \\
&\quad + \frac{L_f}{6} \|x_{k+1} - x_k\|^3 \Bigg]. \qquad (58)
\end{aligned}
$$

Now, let us define the following cubic-regularized upper quadratic model of g_s near x_k (cf. [39]):

$$
\hat{g}_s(y) \coloneqq g_s(x_k) + \langle \nabla g_s(x_k), y - x_k \rangle + \frac{1}{2} \langle H_k^g (y - x_k), y - x_k \rangle + \frac{L_f}{6} \|y - x_k\|^3,
$$

for $y \in \mathbb{R}^n$ and L_f given by P.1. Then, using Lemma 6.1 with $x = x_k$, we have

$$
g_s(y) - \hat{g}_s(y) \leq \omega_\nu (d_\nu(y, x_k)) \|y - x_k\|_x^2 - \frac{1}{2} \langle H_k^g (y - x_k), y - x_k \rangle - \frac{L_f}{6} \|y - x_k\|^3. \quad (59)
$$

Next, using (59) with $y = x_{k+1}$, (58) gives

$$\begin{aligned}\mathcal{L}(x_{k+1}) &\leq \mathcal{L}(x_k) + g_s(x_{k+1}) - \hat{g}_s(x_{k+1}) \\ &\leq \mathcal{L}(x_k) + \left(\omega_\nu(d_\nu(x_{k+1}, x_k)) - \frac{1}{2}\right)\|x_{k+1} - x_k\|_x^2 - \frac{L_f}{6}\|x_{k+1} - x_k\|^3,\end{aligned}$$

which proves the result. ■

A straightforward implication of Proposition 6.4 is that the sequence $\{\mathcal{L}(x_k)\}$ is monotonically decreasing if $\bar{\delta}_k := x_{k+1} - x_k \neq 0$. Consider the set of indices

$$\mathcal{K}_S := \left\{k \text{ such that } x_k \in S \text{ and } S \text{ is a subsequence of } \{x_k\}\right\}. \tag{60}$$

Then, for all $k_j \in \mathcal{K}_S$, $\{x_{k_j}\}$ converges to some $x^\star$.

Lemma 6.5: *Let an iterate x_k be generated by the scheme (19) for problem (2). Then, x_k is a stationary point of $\mathcal{L}$ if and only if $\bar{\delta}_k = 0$.*

Proof: The statement holds true by our characterization of the optimiality conditions in (18) with $Q_k = H_k^g$. ■

Theorem 6.6: *Let $\{x_k\} \subset \mathbb{R}^n$ in Proposition 6.4. Then every limit point $x^\star$ of $\{x_k\}$ at which (18) holds with $Q_k = H_k^g$ is a stationary point of the objective function $\mathcal{L}$ in problem (1).*

Proof: Proposition 6.4 implies $\{\mathcal{L}(x_k)\}$ is nonincreasing and bounded below. Hence, it converges to a finite value $\mathcal{L}^\star$. Consequently (and from the proof of Proposition 4.1), the sequence of iterates $\{x_k\}$ generated from (19) is bounded, and every limit point exists. Let $x^\star$ be a limit point of $\{x_k\}$, and now consider all $k_j \in \mathcal{K}_S$ with $\{x_{k_j}\} \to x^\star$, where $\mathcal{K}_S$ is defined by (60). The relation in (23) implies inclusion in both directions, and hence since $g_s \xrightarrow{e} g$, if $\{x_{k_j}\}$ is such that

$$\limsup_{\substack{x_{k_j} \to x^\star \\ \mu \downarrow 0}} \nabla g_s(x_{k_j}; \mu) \to 0, \tag{61}$$

one finds $x^\star$ is a stationary point of g [14]. For any suitably chosen fixed $\mu \in \mathbb{R}_{++}$, it suffices that both properties (23) and (61) hold only approximately with respect to Proposition 4.1 as they pertain only to the smooth part of the problem. Taking the limit of (18) as $k_j \to \infty$ with $Q_k = H_k^g$, the result follows from Lemma 6.5. Precisely, $\bar{\delta}_{k_j} \to 0$, and hence all the limit points of $\{x_k\}$ are stationary points of $\mathcal{L}$. ■

How to choose α_k. In previous results, we did not specify a particular way to choose α_k. Our algorithms converge for any value of $\alpha_k \in (0, 1]$. Compared to the step-length selection rule proposed in [56], for instance, our approach and analysis do not directly rely on the actual

value of ν in the choice of both $\bar{\alpha}_k$ and α_k. Indeed, in the context of minimizing a function $g_s \in \mathcal{F}_{M_g,\nu}$, an optimal choice for $\bar{\alpha}_k$, in view of [56], corresponds to setting

$$\alpha_k = \begin{cases} \dfrac{\ln(1+d_k)(1+M\eta_k)}{d_k} & \text{if } \nu = 2, \\ \dfrac{2(1+M_g\eta_k)}{2+M_g\eta_k} & \text{if } \nu = 3, \end{cases}$$

where $d_k \coloneqq M_g \|H_k^{g^{-1}} \nabla g_s(x_k)\|$ and in each case, it can be shown that $\bar{\alpha}_k \in (0,1)$. However, choosing α_k this way does not guarantee certain theoretical bounds in the context of the framework studied in this work, especially for $\nu = 2$. We therefore propose to leave α_k as a hyperparameter that must satisfy $0 < \alpha_k \equiv \alpha \leq 1$. This however provides the freedom to exploit specific properties about the function f, when they are known to hold. One of such properties is the global Lipschitz continuity of ∇f, where supposing the Lipschitz constant L is known, one may set

$$\alpha_k = \min\{1/L, 1\}.$$

Local convergence. We next discuss the local convergence properties of Algorithms 1 and 2. In our discussion, we take the local norm $\|\cdot\|_x$ (and its dual) with respect to g_s, and the standard Euclidean norm $\|\cdot\|$ with respect to the (local) Euclidean ball $\mathcal{B}_{r_0}(\cdot) \subset \mathcal{E}_r(\cdot)$. We also remark that, by definition, ω_ν is a strictly increasing function.

Theorem 6.7: *Suppose that* P.1–P.4 *hold, and let* $x^\star$ *be an optimal solution of* (2). *Let* $\{x_k\}$ *be the sequence of iterates generated by Algorithm 1 and define* $\lambda_k \coloneqq 1 + M_g\omega_\nu(-d_\nu(x^\star, x_k))\|x_k - x^\star\|_{x_k}$, *where* ω_ν *is defined by* (47). *Then starting from a point* $x_0 \in \mathcal{E}_r(x^\star)$, *if* $d_\nu(x^\star, x_k) < 1$ *with* d_ν *defined by* (44), *the sequence* $\{x_k\}$ *satisfies*

$$\|x_{k+1} - x^\star\|_{x^\star} \leq \vartheta_k \|x_k - x^\star\| + R_k \|x_k - x^\star\|_{x^\star} + \frac{L_f}{2\sqrt{\rho_2}} \|x_k - x^\star\|^2, \tag{62}$$

where $\vartheta_k \coloneqq (L_1 + L_2)(\lambda_k - \alpha_k)/(\lambda_k\sqrt{\rho})$, $\alpha_k \in (0,1]$, $R_k \coloneqq R_\nu(d_\nu(x^\star, x_k))d_\nu(x^\star, x_k)$ *with* R_ν *defined by* (50).

Proof: The iterative process of Algorithm 1 is given by

$$x_{k+1} = \operatorname{prox}_{\alpha_k g}^{H_k^g}(x_k - \bar{\alpha}_k \nabla^2 q(x_k)^{-1} \nabla q(x_k)).$$

In terms of $E_{\bar{x}}$ and $\xi_{\bar{x}}(Q_k, \cdot)$ with $Q_k \equiv H_k^g$, and using the definition of q, we have

$$\begin{aligned} \|x_{k+1} - x^\star\|_{x^\star} &= \left\| \operatorname{prox}_{\alpha_k g}^{H_\star^g}(E_{x^\star}(x_k) + \xi_{x^\star}(Q_k, x_{k+1})) - \operatorname{prox}_{\alpha_k g}^{H_\star^g}(E_{x^\star}(x^\star)) \right\|_{x^\star} \\ &\overset{(5)}{\leq} \|E_{x^\star}(x_k) - E_{x^\star}(x^\star) + \xi_{x^\star}(Q_k, x_{k+1})\|_{x^\star}^\diamond \\ &= \|H_\star x_k - \bar{\alpha}_k \nabla q(x_k) - H_\star x^\star + \bar{\alpha}_k q(x^\star)\|_{x^\star}^\diamond \\ &= \|\nabla q(x^\star) - \nabla q(x_k) + (1 - \bar{\alpha}_k)(\nabla q(x_k) - \nabla q(x^\star)) + H_\star(x_k - x^\star)\|_{x^\star}^\diamond \end{aligned}$$

$$
\begin{aligned}
&\leq \left\| \nabla q(x_k) - \nabla q(x^\star) - H_\star(x_k - x^\star) \right\|_{x^\star}^{\diamond} + (1-\bar{\alpha}_k) \left\| \nabla q(x_k) - \nabla q(x^\star) \right\|_{x^\star}^{\diamond} \\
&\leq \left\| \nabla f(x_k) - \nabla f(x^\star) - H_\star^f(x_k - x^\star) \right\|_{x^\star}^{\diamond} \\
&\quad + \left\| \nabla g_s(x_k) - \nabla g_s(x^\star) - H_\star^g(x_k - x^\star) \right\|_{x^\star}^{\diamond} \\
&\quad + (1-\bar{\alpha}_k) \left(\left\| \nabla f(x_k) - \nabla f(x^\star) \right\|_{x^\star}^{\diamond} + \left\| \nabla g_s(x_k) - \nabla g_s(x^\star) \right\|_{x^\star}^{\diamond} \right). \qquad (63)
\end{aligned}
$$

To estimate $\|\nabla f(x_k) - \nabla f(x^\star) - H_\star^f(x_k - x^\star)\|_{x^\star}^{\diamond}$, we note that for $v \in \mathbb{R}^n$, $\|v\|_{x^\star}^{\diamond} \equiv \|H_k^{g^{\star -\frac{1}{2}}} v\|$ since we take the dual norm with respect to g_s. Now, using P.2, we get that the matrix $H_\star^g$ is positive definite and

$$
\|H_k^{g^{\star -\frac{1}{2}}}\| \leq \frac{1}{\sqrt{\rho_2}}. \qquad (64)
$$

Consequently, we have

$$
\begin{aligned}
\left\| \nabla f(x_k) - \nabla f(x^\star) - H_\star^f(x_k - x^\star) \right\|_{x^\star}^{\diamond} &= \left\| H_k^{g^{\star -\frac{1}{2}}} \left(\nabla f(x_k) - \nabla f(x^\star) - H_\star^f(x_k - x^\star) \right) \right\| \\
&\leq \|H_k^{g^{\star -\frac{1}{2}}}\| \left\| \nabla f(x_k) - \nabla f(x^\star) - H_\star^f(x_k - x^\star) \right\| \\
&\overset{\text{Lemma 6.2}}{\leq} \frac{L_f \|x_k - x^\star\|^2}{2\sqrt{\rho_2}}.
\end{aligned}
$$

To estimate $\|\nabla g_s(x_k) - \nabla g_s(x^\star) - H_\star^g(x_k - x^\star)\|_{x^\star}^{\diamond}$, we can apply Lemma 6.3 as in the proof of [56, Theorem 5], and get

$$
\left\| \nabla g_s(x_k) - \nabla g_s(x^\star) - H_\star^g(x_k - x^\star) \right\|_{x^\star}^{\diamond} \leq R_v(d_v(x^\star, x_k)) d_v(x^\star, x_k) \left\| x_k - x^\star \right\|_{x^\star}.
$$

Following [56, p. 195], we can derive the following inequality in a neighbourhood of the sublevel set of $\mathcal{L}_s$ in (2) using Lemma 6.1 and the convexity of g_s:

$$
\|\nabla g_s(x_k)\|_{x_k}^{\diamond} \geq \omega_v(-d_v(x^\star, x_k)) \|x_k - x^\star\|_{x_k}. \qquad (65)
$$

In this regard, (28) gives

$$
1 - \bar{\alpha}_k \leq \frac{\lambda_k - \alpha_k}{\lambda_k}. \qquad (66)
$$

Next, by P.2, we deduce

$$
\left\| \nabla g_s(x_k) - \nabla g_s(x^\star) \right\| \leq L_2 \left\| x_k - x^\star \right\|,
$$

and

$$
\left\| \nabla f(x_k) - \nabla f(x^\star) \right\| \leq L_1 \left\| x_k - x^\star \right\|.
$$

Then, using (64), we get

$$\left\|\nabla g_s(x_k) - \nabla g_s(x^\star)\right\|_{x^\star}^{\diamond} = \left\| H_k^{g^{\star-\frac{1}{2}}} \left(\nabla g_s(x_k) - \nabla g_s(x^\star)\right)\right\| \leq \frac{L_2}{\sqrt{\rho_2}}\left\|x_k - x^\star\right\|.$$

Similarly,

$$\left\|\nabla f(x_k) - \nabla f(x^\star)\right\|_{x^\star}^{\diamond} \leq \frac{L_1}{\sqrt{\rho_2}}\left\|x_k - x^\star\right\|.$$

Finally, putting the above estimates into (63), we obtain (62). ■

To prove the local convergence of Algorithm 2, we need an additional assumption about the behaviour of the Jacobian matrix J_k near $x^\star$. As before, J_k denotes the Jacobian matrix evaluated at x_k; likewise, V_k and e_k. At $x^\star$, we respectively write $J^\star$, $V^\star$ and $u^\star$. We assume the following:

G.1 $\|J_k v\| \geq \beta_1 \|v\|$, $\beta_1 \in \mathbb{R}_{++}$, for all x_k near $x^\star$, and for any $v \in \mathbb{R}^n$.

For f defined by (31), condition G.1 implies that the singular values of J_k are uniformly bounded away from zero, at least locally. Let the unaugmented version of the residual vector e_k be denoted by $\tilde{e}_k$, that is,

$$\tilde{e}_k \coloneqq [l'_{\hat{y}^{(1)}}(y^{(1)}, \hat{y}^{(1)}), \ldots, l'_{\hat{y}^{(m)}}(y^{(m)}, \hat{y}^{(m)})]^\top \in \mathbb{R}^m.$$

Define the following matrix:

$$W_k^\top \coloneqq \begin{bmatrix} \hat{y}^{(1)''}(x^{(1)}) & \hat{y}^{(2)''}(x^{(1)}) & \cdots & \hat{y}^{(m)''}(x^{(1)}) \\ \hat{y}^{(1)''}(x^{(2)}) & \hat{y}^{(2)''}(x^{(2)}) & \cdots & \hat{y}^{(m)''}(x^{(2)}) \\ \vdots & \vdots & & \vdots \\ \hat{y}^{(1)''}(x^{(n)}) & \hat{y}^{(2)''}(x^{(n)}) & \cdots & \hat{y}^{(m)''}(x^{(n)}) \end{bmatrix} \in \mathbb{R}^{n\times m}. \tag{67}$$

We note that the 'full' Hessian matrix H_k can be expressed as

$$H_k \equiv J_k^\top V_k J_k + (\mathbf{1} \otimes (W_k^\top \tilde{e}_k))^\top + H_k^g, \tag{68}$$

where $\mathbf{1} \in \mathbb{R}^{n\times 1}$ is the $n \times 1$ matrix of ones and $\otimes$ denotes the outer product. By P.1, P.2 and the Lipschitz continuity of g_s around $x^\star$ in Proposition 3.13(iii), we have: for r small enough, there exists a constant $\beta_2 \in \mathbb{R}_{++}$ such that $\left\|\tilde{e}_k\right\| \leq \beta_2$ near $x^\star$. Furthermore by our assumptions (see, e.g., [40, Theorem 10.1]), we deduce that there exists $\beta_3 \in \mathbb{R}_{++}$ such that $\|W_k\| \leq \beta_3$ near $x^\star$.

The next result follows. Note that for Algorithm 2, we consider the case where f in problem (2) may, in general, be expressed in the form (31).

Theorem 6.8: *Suppose that* P.1–P.4 *hold, and let* $x^\star$ *be an optimal solution of* (2) *where* f *is defined by* (31). *Additionally, let G.1 hold for the Jacobian matrix* J_k *defined by* (32). *Let* $\{x_k\}$ *be the sequence of iterates generated by Algorithm 2, and define* $\lambda_k := 1 + M_g\omega_\nu(-d_\nu(x^\star, x_k))\|x_k - x^\star\|_{x_k}$, *where* ω_ν *is defined by* (47). *Then starting from a point* $x_0 \in \mathcal{E}_r(x^\star)$, *if* $d_\nu(x^\star, x_k) < 1$ *with* d_ν *defined by* (44), *the sequence* $\{x_k\}$ *satisfies*

$$\|x_{k+1} - x^\star\|_{x^\star} \leq \vartheta_k\|x_k - x^\star\| + R_k\|x_k - x^\star\|_{x^\star} + \frac{L_f}{2\sqrt{\rho_2}}\|x_k - x^\star\|^2, \tag{69}$$

where R_k *is as defined in Theorem* 6.7, $\vartheta_k := (\lambda_k(L_1 + L_2)(\lambda_k - \alpha_k) + \tilde{\beta})/\sqrt{\rho_2}$, $\alpha_k \in (0, 1]$, *and* $\tilde{\beta} := \beta_2\beta_3 \in \mathbb{R}_{++}$.

Proof: Let $\hat{H}_k := J_k^\top V_k J_k + H_k^g$, and consider the iterative process of Algorithm 2 given by

$$x_{k+1} = \operatorname{prox}_{\alpha_k g}^{H_k^g}(x_k - \bar{\alpha}_k\hat{H}_k^{-1}J_k^\top e_k).$$

We first note that $J_k^\top e_k$ is a compact way of writing $\nabla f(x_k) + \nabla g_s(x_k) =: \nabla q(x_k)$, where f is given by (31). Following the proof of Theorem 6.7, we have

$$\begin{aligned}\|x_{k+1} - x^\star\|_{x^\star} &= \left\|\operatorname{prox}_{\alpha_k g}^{H_\star^g}(E_{x^\star}(x_k) + \xi_{x^\star}(Q_k, x_{k+1})) - \operatorname{prox}_{\alpha_k g}^{H_\star^g}(E_{x^\star}(x^\star))\right\|_{x^\star} \\ &\leq \left\|\nabla q(x_k) - \nabla q(x^\star) - \hat{H}_k^\star(x_k - x^\star)\right\|_{x^\star}^\diamond + (1 - \bar{\alpha}_k)\left\|\nabla q(x_k) - \nabla q(x^\star)\right\|_{x^\star}^\diamond.\end{aligned} \tag{70}$$

Let $W^\star$ and $\tilde{u}^\star$ respectively denote expressions for W_k and $\tilde{u}$ evaluated at $x^\star$. Substituting (68) into (70) and using (64) in the estimate

$$\left\|(\mathbf{1} \otimes (W^{\star\top}\tilde{e}_k))^\top(x_k - x^\star)\right\|_{x^\star}^\diamond \leq \left\|H_k^{g^{\star - \frac{1}{2}}}(\mathbf{1} \otimes (W^{\star\top}\tilde{u}^\star))^\top\right\|\|x_k - x^\star\|,$$

where W_k is defined by (67), we get

$$\begin{aligned}\|x_{k+1} - x^\star\|_{x^\star} &\leq \left\|\nabla q(x_k) - \nabla q(x^\star) - H_\star(x_k - x^\star)\right\|_{x^\star}^\diamond + \left\|(\mathbf{1} \otimes (W^{\star\top}\tilde{u}^\star))^\top(x_k - x^\star)\right\|_{x^\star}^\diamond \\ &\quad + (1 - \bar{\alpha}_k)\left\|\nabla q(x_k) - \nabla q(x^\star)\right\|_{x^\star}^\diamond \\ &\leq \left\|\nabla f(x_k) - \nabla f(x^\star) - H_\star^f(x_k - x^\star)\right\|_{x^\star}^\diamond \\ &\quad + \left\|\nabla g_s(x_k) - \nabla g_s(x^\star) - H_\star^g(x_k - x^\star)\right\|_{x^\star}^\diamond \\ &\quad + (1 - \bar{\alpha}_k)\left(\left\|\nabla f(x_k) - \nabla f(x^\star)\right\|_{x^\star}^\diamond + \left\|\nabla g_s(x_k) - \nabla g_s(x^\star)\right\|_{x^\star}^\diamond\right) \\ &\quad + \frac{\tilde{\beta}\|x_k - x^\star\|}{\sqrt{\rho_2}},\end{aligned} \tag{71}$$

where $\tilde{\beta} = \beta_2\beta_3$. Now, using the estimates derived in the proof of Theorem 6.7 in (71) above, we obtain (69). ■

7. Numerical experiments

In this section, we validate the efficiency of the technique introduced in this paper in numerical examples using both synthetic and real datasets from the LIBSVM repository [16]. The approach and algorithms proposed in this paper are implemented in the Julia programming language and are available online as an open-source package.[10] We test the performance of Algorithms 1 and 2 for various fixed values of $\alpha_k \equiv \alpha \in (0, 1]$ (see Figure 2). In the remaining parts, we fix $\alpha_k = 1$ and compare our approach with `PANOC` [53], `ZeroFPR` [57], `OWL-QN` [3], proximal gradient [29], and fast proximal gradient [8] algorithms.[11] In the sparse group lasso experiments, we also compare with the block coordinate descent (`BCD`)[12] algorithm, and the semismooth Newton augmented Lagrangian (`SSNAL`) method [28] which was extended[13] in [64] to solve sparse group lasso problems. `BCD` is known to be an efficient algorithm for general regularized problems [22], and is used as a standard approach for the sparse group lasso problem [23,25,51]. Since the problems considered in our experiments use the ℓ_1 and ℓ_2 regularizers, we use $\phi(t) = \frac{1}{p}\sqrt{1 + p^2|t|^2} - 1$ from Example 3.7, with $p = 1$ and derive g_s in problem (2) accordingly.

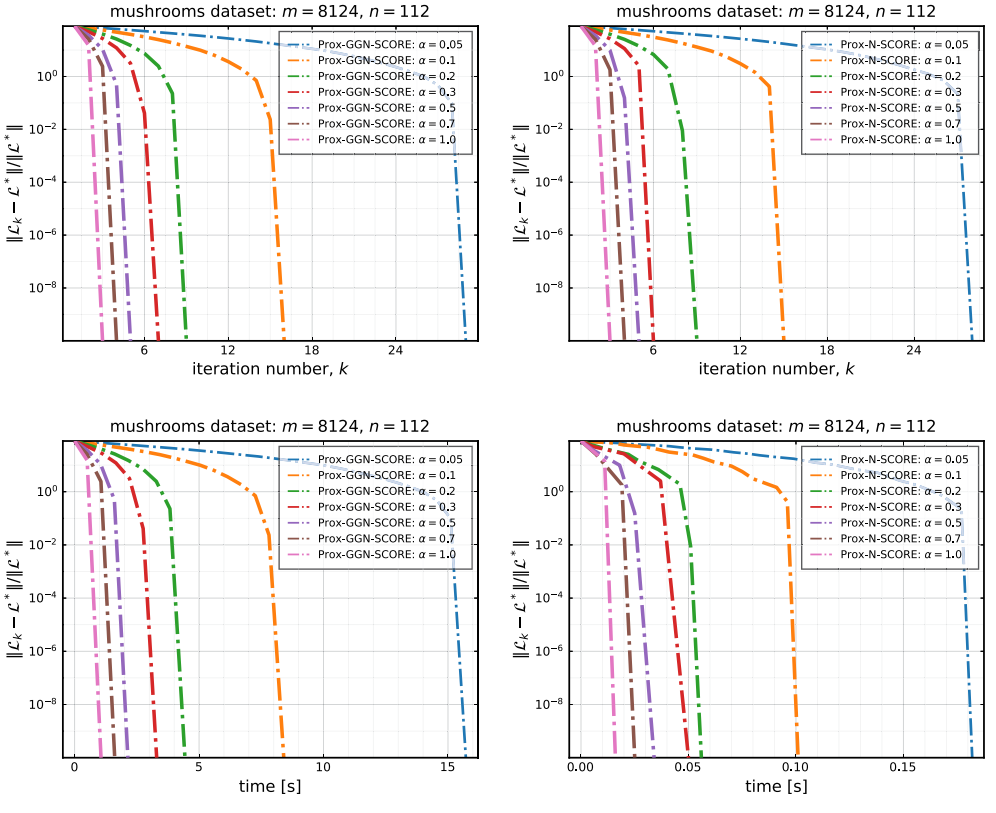

Figure 2. Behaviour of `Prox-N-SCORE` and `Prox-GGN-SCORE` for different fixed values of α_k in problem (72).

For a diagonal matrix $H_k^g \in \mathbb{R}^{n\times n}$, the scaled proximal operator for the 1- and 2-norms are obtained using the proximal calculus derived in [11]. Let $\hat{d}_k \in \mathbb{R}^n$ be the vector containing the diagonal entries of H_k^g, and let $\beta \in \mathbb{R}_{++}$; the components of $\mathrm{prox}^{H_k^g}_{\beta\|\cdot\|_1}$ and $\mathrm{prox}^{H_k^g}_{\beta\|\cdot\|}$ at iteration k are given, respectively, by:

(i) $(\mathrm{prox}^{H_k^g}_{\beta\|\cdot\|_1}(p_k))^{(i)} = \mathrm{sign}(p_k^{(i)})\max\{|p_k^{(i)}| - \beta\hat{d}_k^{(i)}, 0\}$, and

(ii) $(\mathrm{prox}^{H_k^g}_{\beta\|\cdot\|}(p_k))^{(i)} = p_k^{(i)}\max\{1 - \beta\hat{d}_k^{(i)}/\|p_k\|, 0\}$.

We terminate each of the tested algorithms either with the default stopping criterion or when $\frac{\|x_k - x_{k-1}\|}{\max\{\|x_{k-1}\|,1\}} < \varepsilon_{tol}$ with $\varepsilon_{tol} \in \{10^{-6}, 10^{-10}\}$.

All experiments are performed on a laptop with dual (2.30 GHz + 2.30 GHz) Intel Core i7-11800 H CPU and 32 GB RAM.

7.1. Sparse logistic regression

We consider the problem of finding a sparse solution x to the following logistic regression problem

$$\min_{x\in\mathbb{R}^n} \mathcal{L}(x) := \underbrace{\sum_{i=1}^{m} \log\left(1 + \exp(-y^{(i)}\langle a^{(i)}, x\rangle)\right)}_{=:f(x)} + \beta\|x\|_1, \tag{72}$$

where, in view of (1), $g(x) := \beta\|x\|_1$, $\beta \in \mathbb{R}_{++}$, and $a^{(i)} \in \mathbb{R}^n$, $y^{(i)} \in \{-1, 1\}$ form the data. We perform experiments on both randomly generated data and real datasets summarized in Table 2. For the synthetic data, we set $\beta = 0.2$, while for the real datasets, we set $\beta = 1$. We fix $\mu = 1$ in both Algorithms 1 and 2, and set $\alpha_k = 1/L$ for the proximal gradient algorithm, where L is estimated as $L = \lambda_{\max}(A^\top A)$, the columns of $A \in \mathbb{R}^{n\times m}$ are the vectors $a^{(i)}$ and $\lambda_{\max}$ denotes the largest eigenvalue. For the sake of fairness, we provide this value of L to each of `PANOC`, `ZeroFPR`, and fast proximal gradient algorithms for computing their step-lengths in our comparison.

Table 2. Summary of the real datasets used for sparse logistic regression.

Data	m	n	Density
`mushrooms`	8124	112	0.19
`phishing`	11,055	68	0.44
`w1a`	2477	300	0.04
`w2a`	3470	300	0.04
`w3a`	4912	300	0.04
`w4a`	7366	300	0.04
`w5a`	9888	300	0.04
`w8a`	49,749	300	0.04
`a1a`	1605	123	0.11
`a2a`	2265	123	0.11
`a3a`	3185	123	0.11
`a4a`	4781	123	0.11
`a5a`	6414	123	0.11

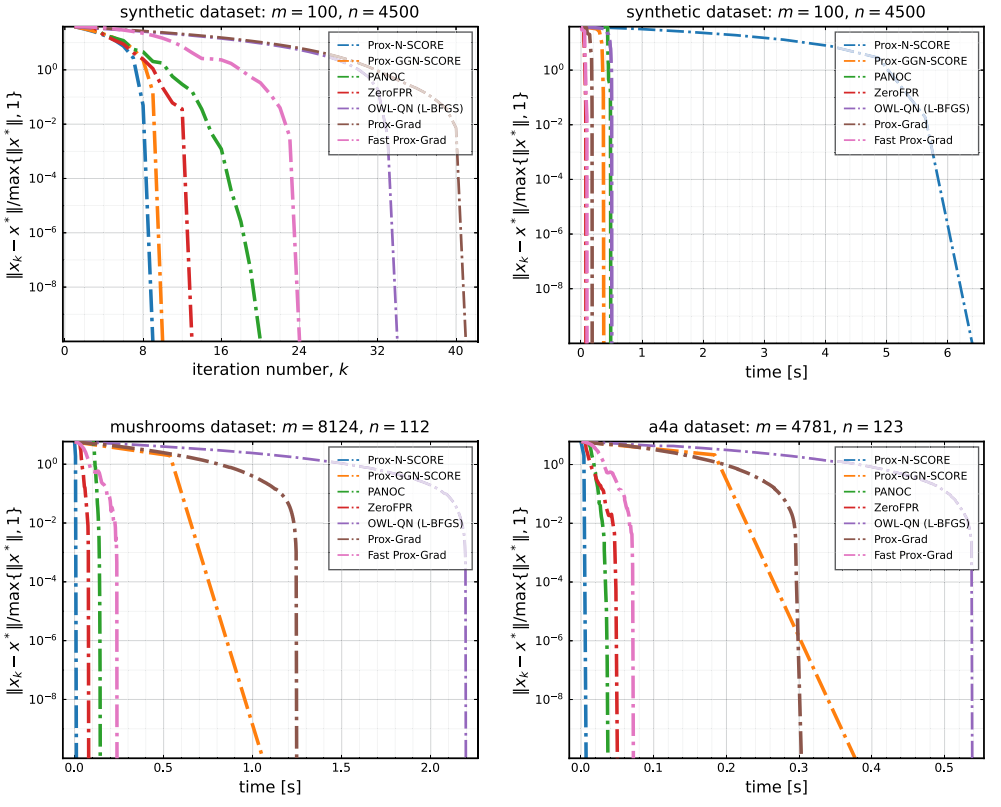

Figure 3. Overparameterized problem (first row) and nonoverparameterized problems (second row) in (72). `Prox-GGN-SCORE` reduces most of the computational burden of `Prox-N-SCORE` if $m + n_y < n$ (or $m \ll n$). However, `Prox-N-SCORE` solves the problem faster, and is more stable, if $n < m + n_y$ (or $n \ll m$).

The results are shown in Figures 2–4. In Figure 3, we observe that `Prox-GGN-SCORE` reduces most of computational burden of the quasi-Newton method when $m + n_y < n$ and makes the method competitive with the first-order methods considered. However, as shown in both Figures 2 and 3, `Prox-GGN-SCORE` is no longer preferred when $n < m + n_y$ and, by our experiments, the algorithm can run into computational issues when $n \ll m$. In this case (particularly for all of the real datasets that we use in this example), `Prox-N-SCORE` would be preferred and, as shown in the performance profile of Figure 4, outperforms other tested algorithms in most cases, especially with $\alpha = 1$.

7.2. Sparse group lasso

In this example, we consider the sparse group lasso problem (38):

$$\min_{x \in \mathbb{R}^n} \mathcal{L}(x) := \underbrace{\frac{1}{2}\left\|Ax - y\right\|^2}_{=:f(x)} + \underbrace{\beta\|x\|_1 + \beta_{\mathcal{G}} \sum_{j \in \mathcal{G}} \omega_j \|x^{(j)}\|}_{=:g(x)}. \tag{73}$$

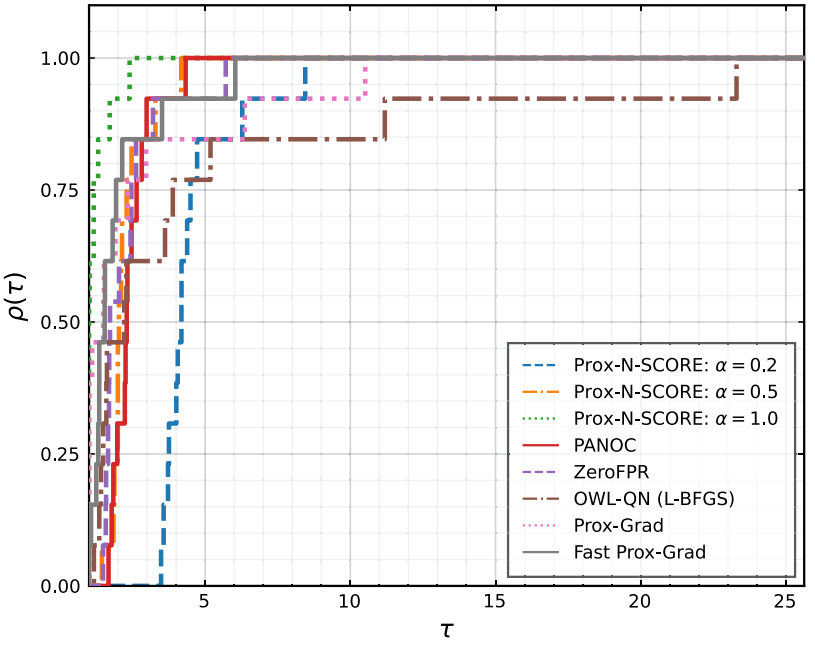

Figure 4. Performance profile (CPU time) for the sparse logistic regression problem (72) using the LIBSVM datasets summarized in Table 2. Here, τ denotes the performance ratio (CPU times in seconds) averaged over 20 independent runs with different random initializations, and $\rho(\tau)$ is the corresponding frequency.

We use the common example used in the literature [58,62], which is based on the model $y = Ax^\star + 0.01\epsilon \in \mathbb{R}^{m\times 1}$, $\epsilon \sim \mathcal{N}(0,1)$. The entries of the data matrix $A \in \mathbb{R}^{m\times n}$ are drawn from the normal distribution with pairwise correlation $\text{corr}(A^{(i)}, A^{(j)}) = 0.5^{|i-j|}$, $\forall (i,j) \in \{1,\ldots,n\}^2$. We generate datasets for different values of m and n with n satisfying $(n \mod n_g) = 0$. In this problem, we want to further highlight the faster computational time achieved by the approximation in `Prox-GGN-SCORE`, so we consider only overparameterized models (i.e., with $m + n_y \leq n$).

In this problem, the matrix C in the reformulation (36) is a diagonal matrix with row indices given by all pairs $(i,j) \in \{(i,j)|i \in j, i \in \{1,\ldots,n_g\}, j \in \mathcal{G}\}$, and column indices given by $k \in \{1,\ldots,n_g\}$. That is,

$$C^{((i,j),k)} = \begin{cases} \beta_{\mathcal{G}}\omega_j & \text{if } i = k, \\ 0 & \text{otherwise.} \end{cases}$$

We construct $x^\star$ in a similar way as [32]: We fix $n_g = 100$ and break n randomly into groups of equal sizes with 0.1 percent of the groups selected to be *active*. The entries of the subvectors in the *nonactive* groups are set to zero, while for the active groups, $\lceil \frac{n}{n_g} \rceil \times 0.1$ of the subvector entries are drawn randomly and set to $\text{sign}(\xi) \times U$ where ξ and U are uniformly distributed in $[0.5, 10]$ and $[-1, 1]$, respectively; the remaining entries are set to zero. For the sake of fair comparison, each data and the associated initial vector x_0 are generated in Julia, and exported for the `BCD` implementation in Python and also for `SSNAL` in MATLAB.

For `Prox-GGN-SCORE`, `Prox-Grad` and `BCD`, we set $\beta = \tau_1 \gamma \|A^\top y\|_\infty$, $\beta_{\mathcal{G}} = (10 - \tau_1)\gamma \|A^\top y\|_\infty$ with $\tau_1 = 0.9$ and $\gamma \in \{10^{-7}, 10^{-8}\}$. `SSNAL` can be made to return a solution estimate that has number of nonzero entries close to that of the true solution with a carefully tuned β and simply setting $\beta_{\mathcal{G}} = \beta$ (cf., [64, Table 1]). However, by our numerical experiments, `SSNAL` can be very sensitive to the choice of β and $\beta_{\mathcal{G}}$ if the goal is to have a reasonable convergence to the true solution with the correct within-group sparsity in the solution estimate. After a careful tuning, and for the sake of fair comparison, we set $\beta = \tau_1 \gamma \|A^\top y\|_\infty$ and $\beta_{\mathcal{G}} = \|A^\top y\|_\infty$ with $\gamma = 10^{-5}$ and $\tau_1 \in \{4, 5, 10, 12\}$ (depending on the problem size) for `SSNAL`. For each group j, the parameter ω_j is set to the standard value $\sqrt{n_j}$ [23,51], where $n_j = \text{card}(j)$. For fairness, the estimate $\alpha_k = 1/L$ with $L = \lambda_{\max}(A^\top A)$ is used in the proximal gradient and `SSNAL` algorithms.

We set μ to 1.2 for $m = 500$, $n = 2000$, 2.0 for $m = 1000$, $n = 12{,}000$, and to 1.6 in the remaining setups. The simulation results are shown in Table 3 and Figure 5. As shown, `Prox-GGN-SCORE` terminates faster than `SSNAL`, `Prox-Grad` and `BCD` algorithms in most cases with the correct number of nonzero entries in its solution estimates. Additionally, the results further highlight the computational benefits of `Prox-GGN-SCORE` for overparameterized problems.

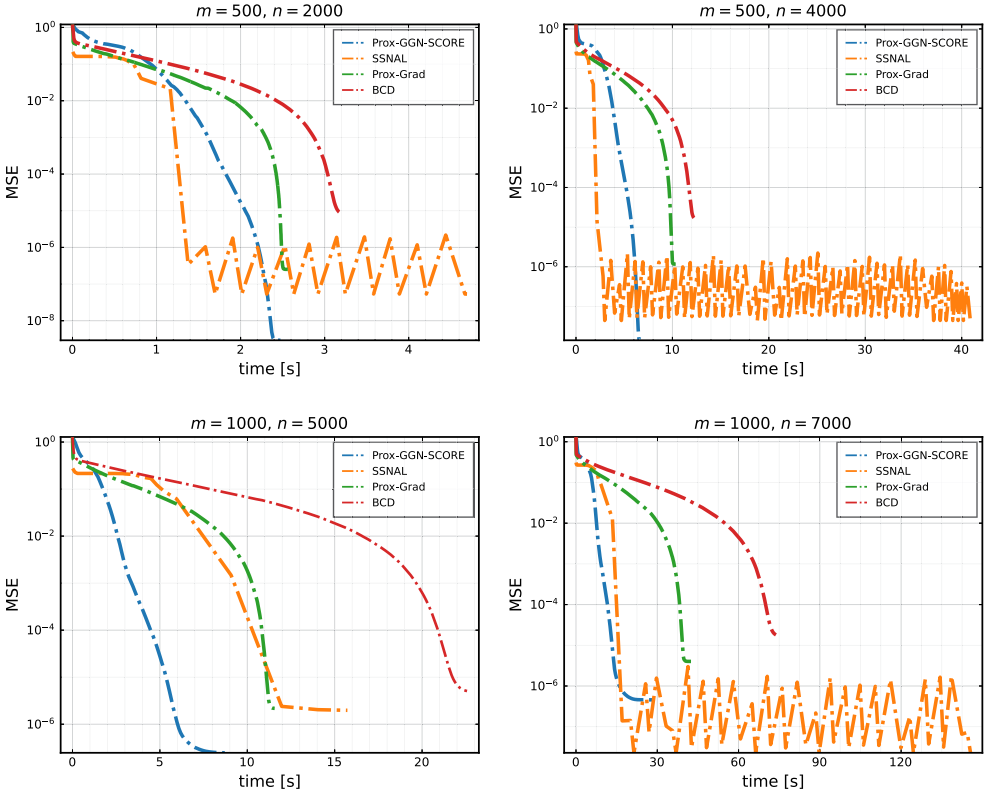

Figure 5. Mean squared error (MSE) between the estimates x_k and the true coefficient $x^\star$ for `Prox-GGN-SCORE`, `SSNAL`, `Prox-Grad` and `BCD` on the sparse group lasso problem (73).

Table 3. Performance of `Prox-GGN-SCORE` (`alg.A`), `SSNAL` (`alg.B`), `Prox-Grad` (`alg.C`) and `BCD` (`alg.D`) on the sparse group lasso problem (73) for different values of m and n.

	nnz				Iteration[14]				Time [s]				MSE			
$(m, n; nnz)$	`alg.A`	`alg.B`	`alg.C`	`alg.D`	`alg.A`	`alg.B`	`alg.C`	`alg.D`	`alg.A`	`alg.B`	`alg.C`	`alg.D`	`alg.A`	`alg.B`	`alg.C`	`alg.D`
(500, 2000; 19)	19	198	19	19	161	62	5904	9690	**2.81**	4.65	13.39	3.55	2.9305E−09	5.3188E−08	2.5189E−07	8.0350E−06
(500, 4000; 36)	36	39	36	36	253	140	10,991	16,790	**8.44**	39.51	51.91	11.60	1.4291E−08	4.3952E−08	1.1653E−06	1.7127E−05
(500, 5000; 45)	45	45	45	45	530	111	13,919	20,830	**16.60**	35.57	90.05	18.53	2.6339E−07	6.0898E−08	2.0121E−06	2.1641E−05
(1000, 5000; 45)	45	82	45	45	112	35	3051	9100	**8.71**	15.73	11.57	22.14	2.4667E−07	1.9757E−06	2.1747E−06	5.0779E−06
(1000, 7000; 65)	65	65	65	65	185	82	7012	20,870	**30.26**	148.07	42.45	70.08	4.5689E−07	2.2847E−08	4.0172E−06	1.8038E−05
(1000, 10,000; 94)	93	94	94	94	497	102	9879	29,330	**53.26**	252.05	90.25	126.17	3.8421E−06	2.8441E−08	3.6320E−06	3.5855E−05
(1000, 12,000; 112)	112	113	113	164	663	68	21,178	59,360	**166.15**	194.40	221.26	373.50	1.5750E−05	4.6965E−08	7.3285E−06	5.9521E−05

Note: nnz stands for the number of nonzero entries of $x^\star$ and of the solutions found by the algorithms. MSE stands for the mean squared error between the true solution $x^\star$ and the estimated solutions.

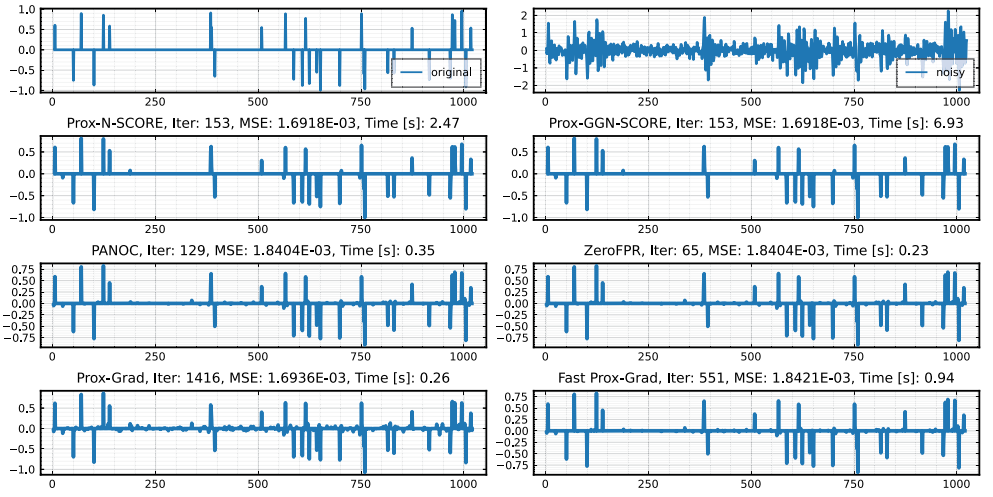

Figure 6. Sparse deconvolution via ℓ_1-regularized least squares (74) using `Prox-N-SCORE`, `Prox-GGN-SCORE`, `PANOC`, `ZeroFPR`, proximal gradient, and fast proximal gradient algorithms with $n = 1024$.

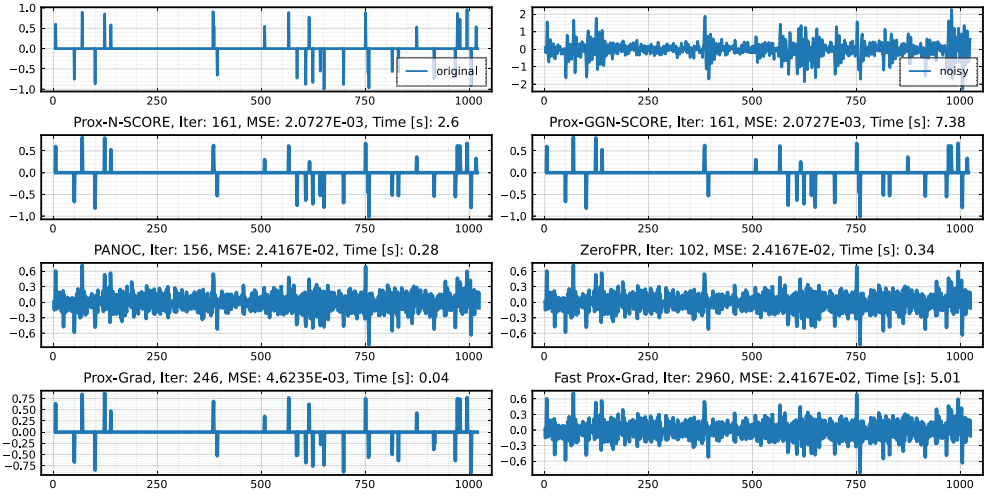

Figure 7. Sparse deconvolution via ℓ_2-regularized least squares (74) using `Prox-N-SCORE`, `Prox-GGN-SCORE`, `PANOC`, `ZeroFPR`, proximal gradient, and fast proximal gradient algorithms with $n = 1024$.

7.3. Sparse deconvolution

In this example, we consider the problem of estimating the unknown sparse input x to a linear system, given a noisy output signal and the system response. That is,

$$\min_{x\in\mathbb{R}^n} \mathcal{L}(x) \coloneqq \underbrace{\frac{1}{2}\|Ax - y\|^2}_{=:f(x)} + \beta\|x\|_p, \tag{74}$$

where $A \in \mathbb{R}^{n\times n}$ and $y \in \mathbb{R}^{n\times 1}$ are given data about the system which we randomly generate according to [50, Example F].

We solve with both ℓ_1 ($p=1$) and ℓ_2 ($p=2$) regularizers, and set $\beta = 10^{-3}$. We set $\mu = 5 \times 10^{-2}$ in the smooth approximation g_s of g. We estimate $L = \lambda_{\max}(A^\top A)$ and set $\alpha_k = 1/L$ in the proximal gradient algorithm. Again, for fairness, we provide this value of L to each of `PANOC`, `ZeroFPR`, and fast proximal gradient procedures in our comparison. The simulation results are displayed in Figures 6 and 7. While `Prox-GGN-SCORE` and `Prox-N-SCORE` sometimes use more computational time in this problem, they provide better solution quality with smaller reconstruction error than the other tested algorithms, which is more desirable for signal reconstruction problems.

8. Conclusion

In this paper, we introduced a self-concordant regularization framework for proximal quasi-Newton methods that solves large-scale convex composite optimization problems while preserving the structure induced by nonsmooth regularizers. Two algorithms are studied: a proximal Newton algorithm (`Prox-N-SCORE`) and a proximal generalized Gauss-Newton algorithm (`Prox-GGN-SCORE`). Both algorithms share an adaptive step-length rule that eliminates the need for line search or trust-region subroutines, and they employ a diagonal variable metric derived from the smooth regularization. These design choices guarantee global convergence and yield favorable local behaviour under standard regularity assumptions. The `Prox-GGN-SCORE` variant relies on a low-rank approximation of the Hessian inverse that exploits the structure of prediction models (e.g., in machine learning). This makes it especially effective for overparameterized regimes where the number of decision variables exceeds the number of observations, allowing the method to scale to high-dimensional problems without forming full matrix inverses. Future work will focus on adaptive selection of the smoothing parameter, a theoretical analysis of how self-concordant smoothing influences optimization dynamics and generalization in scientific machine learning settings, and the derivation of explicit complexity estimates for both algorithms.

Notes

1. In this work, we use 'regularization' and 'smoothing' interchangeably but use 'regularization' to emphasize explicit addition of a smooth function (a smooth approximation of the *nonsmooth part* of the problem) to the *smooth part* of the problem.
2. We occasionally write $g_s(x)$ instead of $g_s(x;\mu)$ to refer to the same function.
3. https://github.com/adeyemiadeoye/SelfConcordantSmoothOptimization.jl
4. Also sometimes called 'epigraphic sum' or 'epi-sum', as its operation yields the (strict) *epigraphic sum* epi$g$ + epi$h$ [24, p. 93].
5. It is easy to show that $h^\star_\mu = \mu h^\star$.
6. Note that for the sake of simplicity, we assume here $y^{(i)} \in \mathbb{R}$, but it is straightforward to extend the approach that follows to cases where $y^{(i)} \in \mathbb{R}^{n_y}$, $n_y > 1$.
7. The reader should not confuse the barrier smoothing technique of, say, [36,59], with the self-concordant smoothing framework of this paper. The self-concordant barrier smoothing techniques, just like Nesterov's smoothing, realize first-order and subgradient algorithms that solve problems of this exact form.
8. A function d_1 is called a *prox-function* of a closed and convex set $\mathcal{Q}_1$ if $\mathcal{Q}_1 \subseteq \operatorname{dom} d_1$, and d_1 is continuous and strongly convex on $\mathcal{Q}_1$ with convexity parameter $\rho_1 \in \mathbb{R}_{++}$ [35].

9. Additional assumptions may be required to hold in order to accurately define this property in our framework, e.g., nonoverlapping groups in case of the sparse group lasso problem, in which case, $\mathbb{V}$ is the space $\mathbb{R}^n$.
10. https://github.com/adeyemiadeoye/SelfConcordantSmoothOptimization.jl. Code to reproduce most of the experiments in this paper can be found in the v0.1.0 release.
11. We use the open-source package `ProximalAlgorithms.jl` for the `PANOC`, `ZeroFPR`, and fast proximal gradient algorithms, while we use our own implementation of the `OWL-QN` (modification of https://gist.github.com/yegortk/ce18975200e7dffd1759125972cd54f4) and proximal gradient methods.
12. We use the `BCD` method of [33] which is efficiently implemented with a gap safe screening rule. The open-source implementation can be found in https://github.com/EugeneNdiaye/Gap_Safe_Rules.
13. We use the freely available implementation provided by the authors in https://github.com/YangjingZhang/SparseGroupLasso.
14. Number of 'outer' iterations is displayed for `SSNAL` (`alg.B`). In an augmented Lagrangian method, most of the computational time is likely spent on the inner iterations.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was funded by the European Union (ERC Advanced Research Grant COMPACT, No. 101141351). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

Data availability statement

The data that support the findings of this study are openly available in LIBSVM [16] at https://www.csie.ntu.edu.tw/cjlin/libsvmtools/datasets/binary.html.

Notes on contributors

Adeyemi D. Adeoye received his B.Sc. degree (Hons.) in Mathematics from the University of Ilorin, Ilorin, Nigeria, in 2016, his M.Sc. degree in Mathematical Sciences from the African Institute for Mathematical Sciences (AIMS) in Limbe, Cameroon, in 2018, and his M.Sc. degree in Machine Intelligence from AIMS Kigali, Rwanda, in 2021. He received his Ph.D. degree in Systems Science at IMT School for Advanced Studies Lucca, Lucca, Italy, in 2025, where he is currently a research fellow. His research interests include mathematical optimization, data-driven control, and neural networks.

Alberto Bemporad received his M.S. degree cum laude in Electrical Engineering (1993) and his Ph.D. in Control Engineering (1997) from the University of Florence, Italy. He held research positions at Washington University in St. Louis (1996-1997) and ETH Zurich (1997-2002). He served at the University of Siena (1999-2009) and at the University of Trento (2010-2011). Since 2011, he has been Full Professor at the IMT School for Advanced Studies Lucca, where he was Director from 2012 to 2015. He co-founded ODYS S.r.l. in 2011, a company specialized in industrial model predictive control systems, and has held visiting appointments at Stanford University, the University of Michigan, and Zhejiang University. He has authored over 400 publications and 21 patents in model predictive control, hybrid systems, optimization, and automotive control, and co-developed the Model Predictive Control Toolbox (MathWorks) for MATLAB. He was an Associate Editor of the IEEE Transactions on Automatic Control during 2001-2004 and Chair of the IEEE CSS Technical Committee on Hybrid Systems during 2002-2010. He is an IEEE Fellow (since 2010) and IFAC Fellow

(since 2025). His awards include the IFAC High-Impact Paper Award (2011-14), IEEE CSS Transition to Practice Award (2019), SAE Environmental Excellence in Transportation Award (2021), the Beale-Orchard-Hays Prize (2024), and an ERC Advanced Grant (2024).

References

[1] A.D. Adeoye and A. Bemporad, *SC-Reg: Training overparameterized neural networks under self-concordant regularization*, Tech. Rep., IMT School for Advanced Studies Lucca, 2021. https://arxiv.org/abs/2112.07344v1.

[2] A.D. Adeoye and A. Bemporad, *SCORE: Approximating curvature information under self-concordant regularization*, Comput. Optim. Appl. 86 (2023), pp. 599–626.

[3] G. Andrew and J. Gao, *Scalable training of ℓ_1-regularized log-linear models*, in *Proceedings of the 24th International Conference on Machine Learning, ICML '07*, Association for Computing Machinery, New York, NY, USA, 2007, pp. 33–40.

[4] F. Bach, *Self-concordant analysis for logistic regression*, Electron. J. Stat. 4 (2010), pp. 384–414.

[5] H.H. Bauschke and J.M. Borwein, *Legendre functions and the method of random Bregman projections*, J. Convex Anal. 4 (1997), pp. 27–67.

[6] H.H. Bauschke and P.L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, CMS Books in Mathematics, Springer, New York, 2017.

[7] H.H. Bauschke, J. Bolte, and M. Teboulle, *A descent lemma beyond Lipschitz gradient continuity: First-order methods revisited and applications*, Math. Oper. Res. 42 (2017), pp. 330–348.

[8] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sci. 2 (2009), pp. 183–202.

[9] A. Beck and M. Teboulle, *Smoothing and first order methods: A unified framework*, SIAM J. Optim. 22 (2012), pp. 557–580.

[10] S. Becker and J. Fadili, *A quasi-Newton proximal splitting method*, in *Advances in neural information processing systems* 25 (2012).

[11] S. Becker, J. Fadili, and P. Ochs, *On quasi-Newton forward-backward splitting: Proximal calculus and convergence*, SIAM J. Optim. 29 (2019), pp. 2445–2481.

[12] A. Ben-Tal and M. Teboulle, A smoothing technique for nondifferentiable optimization problems, in *Optimization: Proceedings of the Fifth French-German Conference*, Springer, Castel Novel, France, 1989. pp. 1–11.

[13] D.P. Bertsekas, Nondifferentiable optimization via approximation, in *Nondifferentiable optimization*, Springer, Berlin, Heidelberg, 2009. pp. 1–25.

[14] J.V. Burke and T. Hoheisel, *Epi-convergent smoothing with applications to convex composite functions*, SIAM J. Optim. 23 (2013), pp. 1457–1479.

[15] J.V. Burke and T. Hoheisel, *Epi-convergence properties of smoothing by infimal convolution*, Set-Valued Var. Anal. 25 (2017), pp. 1–23.

[16] C.C. Chang and C.J. Lin, *LIBSVM: A library for support vector machines*, ACM Trans. Intell. Syst. Technol. 2 (2011), pp. 27:1–27:27. Software available at http://www.csie.ntu.edu.tw/cjlin/libsvm.

[17] X. Chen, *Smoothing methods for nonsmooth, nonconvex minimization*, Math. Program. 134 (2012), pp. 71–99.

[18] X. Chen, S. Kim, Q. Lin, J.G. Carbonell, and E.P. Xing, *Graph-structured multi-task regression and an efficient optimization method for general fused lasso*, preprint (2010), arXiv:1005.3579.

[19] X. Chen, Q. Lin, S. Kim, J.G. Carbonell, and E.P. Xing, *Smoothing proximal gradient method for general structured sparse regression*, Ann. Appl. Stat. 6 (2012), pp. 719–752.

[20] P.L. Combettes and J.C. Pesquet, Proximal splitting methods in signal processing, in *Fixed-point algorithms for inverse problems in science and engineering*, Springer, New York, 2011. pp. 185–212.

[21] A. De Pierro and A. Iusem, *A relaxed version of Bregman's method for convex programming*, J. Optim. Theory Appl. 51 (1986), pp. 421–440.

[22] J. Friedman, T. Hastie, and R. Tibshirani, *Regularization paths for generalized linear models via coordinate descent*, J. Stat. Softw. 33 (2010), pp. 1–22.

[23] J. Friedman, T. Hastie, and R. Tibshirani, *A note on the group lasso and a sparse group lasso*, preprint (2020), arXiv:1001.0736 (2010).
[24] J.B. Hiriart-Urruty and C. Lemaréchal, *Fundamentals of Convex Analysis*Grundlehren Text EditionsSpringer Science & Business Media, Berlin, Heidelberg, 2004.
[25] Y. Ida, Y. Fujiwara, and H. Kashima, *Fast sparse group lasso*, in *Advances in neural information processing systems* 32 (2019).
[26] S. Kim, K.A. Sohn, and E.P. Xing, *A multivariate regression approach to association analysis of a quantitative trait network*, Bioinformatics 25 (2009), pp. i204–i212.
[27] J.D. Lee, Y. Sun, and M.A. Saunders, *Proximal Newton-type methods for minimizing composite functions*, SIAM J. Optim. 24 (2014), pp. 1420–1443.
[28] X. Li, D. Sun, and K.C. Toh, *A highly efficient semismooth Newton augmented Lagrangian method for solving Lasso problems*, SIAM J. Optim. 28 (2018), pp. 433–458.
[29] P.L. Lions and B. Mercier, *Splitting algorithms for the sum of two nonlinear operators*, SIAM J. Numer. Anal. 16 (1979), pp. 964–979.
[30] Y. Lucet, *Faster than the fast Legendre transform, the linear-time Legendre transform*, Numer. Algorithms 16 (1997), pp. 171–185.
[31] K. Mishchenko, *Regularized Newton method with global* $\mathcal{O}(\mathbf{1/k^2})$ *convergence*, SIAM J. Optim. 33 (2023), pp. 1440–1462.
[32] E. Ndiaye, O. Fercoq, A. Gramfort, and J. Salmon, *Gap safe screening rules for sparse-group lasso*, in *Advances in neural information processing systems* 29 (2016).
[33] E. Ndiaye, O. Fercoq, A. Gramfort, and J. Salmon, *Gap safe screening rules for sparsity enforcing penalties*, J. Mach. Learn. Res. 18 (2017), pp. 4671–4703.
[34] Y. Nesterov, *A method for unconstrained convex minimization problem with the rate of convergence O (1/k^2)*, Dokl. Akad. Nauk. SSSR 269 (1983), pp. 543–547.
[35] Y. Nesterov, *Smooth minimization of non-smooth functions*, Math. Program. 103 (2005), pp. 127–152.
[36] Y. Nesterov, *Barrier subgradient method*, Math. Program. 127 (2011), pp. 31–56.
[37] Y. Nesterov, *Lectures on Convex Optimization*, Vol. 137, Springer, Cham, 2018.
[38] Y. Nesterov and A. Nemirovskii, *Interior-point Polynomial Algorithms in Convex Programming*, SIAM, Philadelphia, 1994.
[39] Y. Nesterov and B.T. Polyak, *Cubic regularization of Newton method and its global performance*, Math. Program. 108 (2006), pp. 177–205.
[40] J. Nocedal and S.J. Wright, *Numerical Optimization*, Springer, New York, NY, 1999.
[41] D.M. Ostrovskii and F. Bach, *Finite-sample analysis of M-estimators using self-concordance*, Electron. J. Stat. 15 (2021), pp. 326–391.
[42] N. Parikh and S. Boyd, *Proximal algorithms*, Found. Trends Optim. 1 (2014), pp. 127–239.
[43] M. Patriksson, *A unified framework of descent algorithms for nonlinear programs and variational inequalities*, Ph.D. diss., Linköping University Linköping, Sweden, 1993.
[44] M. Patriksson, *Cost approximation: A unified framework of descent algorithms for nonlinear programs*, SIAM J. Optim. 8 (1998), pp. 561–582.
[45] P. Patrinos and A. Bemporad, *Proximal Newton methods for convex composite optimization*, in *52nd IEEE Conference on Decision and Control*, IEEE, 2013, pp. 2358–2363.
[46] P. Patrinos, L. Stella, and A. Bemporad, *Forward-backward truncated Newton methods for convex composite optimization*, preprint (2014), arXiv:1402.6655.
[47] R.T. Rockafellar, *Monotone operators and the proximal point algorithm*, SIAM J. Control Optim. 14 (1976), pp. 877–898.
[48] R.T. Rockafellar and R.J.B. Wets*Variational Analysis*, Springer Science & Business Media, Berlin, Heidelberg, 2009.
[49] A. Rodomanov and Y. Nesterov, *Greedy quasi-Newton methods with explicit superlinear convergence*, SIAM J. Optim. 31 (2021), pp. 785–811.
[50] I.W. Selesnick and I. Bayram, *Sparse signal estimation by maximally sparse convex optimization*, IEEE Trans. Signal. Process. 62 (2014), pp. 1078–1092.
[51] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani, *A sparse-group lasso*, J. Comput. Graph. Stat.22 (2013), pp. 231–245.

[52] L. Stella, A. Themelis, and P. Patrinos, *Forward–backward quasi-Newton methods for nonsmooth optimization problems*, Comput. Optim. Appl. 67 (2017), pp. 443–487.
[53] L. Stella, A. Themelis, P. Sopasakis, and P. Patrinos, *A simple and efficient algorithm for nonlinear model predictive control*, in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, IEEE, 2017, pp. 1939–1944.
[54] T. Strömberg, *A study of the operation of infimal convolution*, Ph.D. diss., Luleå Tekniska Universitet, 1994.
[55] D. Sun, *The strong second-order sufficient condition and constraint nondegeneracy in nonlinear semidefinite programming and their implications*, Math. Oper. Res. 31 (2006), pp. 761–776.
[56] T. Sun and Q. Tran-Dinh, *Generalized self-concordant functions: A recipe for Newton-type methods*, Math. Program. 178 (2019), pp. 145–213.
[57] A. Themelis, L. Stella, and P. Patrinos, *Forward-backward envelope for the sum of two nonconvex functions: Further properties and nonmonotone linesearch algorithms*, SIAM J. Optim. 28 (2018), pp. 2274–2303.
[58] R. Tibshirani, J. Bien, J. Friedman, T. Hastie, N. Simon, J. Taylor, and R.J. Tibshirani, *Strong rules for discarding predictors in Lasso-type problems*, J. R. Stat. Soc. Ser. B Stat. Methodol. 74 (2012), pp. 245–266.
[59] Q. Tran-Dinh, Y.H. Li, and V. Cevher, *Barrier smoothing for nonsmooth convex minimization*, in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, 2014, pp. 1503–1507.
[60] Q. Tran-Dinh, A. Kyrillidis, and V. Cevher, *Composite self-concordant minimization*, J. Mach. Learn. Res. 16 (2015), pp. 371–416.
[61] P. Tseng. 2008, On accelerated proximal gradient methods for convex-concave optimization, Tech. Rep., Department of Mathematics, University of Washington, Seattle. https://www.mit.edu/ dimitrib/PTseng/papers/apgm.pdf.
[62] J. Wang and J. Ye, *Two-layer feature reduction for sparse-group lasso via decomposition of convex sets*, in *Advances in Neural Information Processing Systems* 27 (2014).
[63] Y.L. Yu, *On decomposing the proximal map*, in *Advances in neural information processing systems* 26 (2013).
[64] Y. Zhang, N. Zhang, D. Sun, and K.C. Toh, *An efficient Hessian based algorithm for solving large-scale sparse group Lasso problems*, Math. Program. 179 (2020), pp. 223–263.