General Proximal Gradient Method: A Case for Non-Euclidean Norms

Marwa El Halabi, Ya-Ping Hsieh, Bang Vu, Quang Nguyen, and Volkan Cevher

Laboratory for Information and Inference Systems (LIONS), EPFL

August 31, 2017

Abstract

In this paper, we consider composite convex minimization problems. We advocate the merit of considering *Generalized Proximal gradient Methods (GPM)* where the norm employed is not Euclidean. To that end, we show the tractability of the *general proximity operator* for a broad class of structure priors by proposing a polynomial-time approach to approximately compute it. We also identify a special case of regularizers whose proximity operator admits an efficient greedy algorithm. We then introduce a proximity/projection-free accelerated variant of GPM. We illustrate numerically the benefit of non-Euclidean norms, on the estimation quality of the Lasso problem and on the time-complexity of the latent group Lasso problem.

1 Introduction

Composite convex minimization with the following template [21] is prevalent in machine learning:

$$F^\star = \min_{\mathbf{x}\in\mathbb{R}^p} \Big\{ F(\mathbf{x}) := f(\mathbf{x}) + g(\mathbf{x}) \Big\}, \tag{1}$$

where f is a smooth convex loss function, often representing the empirical estimate of some risk, and g is a non-smooth regularizer, which acts as a structure prior.

The proximal gradient method and its accelerated variant [6, 21] are the methods of choice for (1), whenever the proximal operator of g can be computed efficiently:

$$\operatorname{prox}_g^{\ell_2}(\mathbf{u}, \mathbf{z}, L_2) := \arg\min_{\mathbf{x}\in\mathbb{R}^p} \mathbf{z}^T\mathbf{x} + \frac{L_2}{2}\|\mathbf{x}-\mathbf{u}\|_2^2 + g(\mathbf{x}), \tag{2}$$

where the constant L_2 is typically chosen as the Lipschitz constant of ∇f with respect to the ℓ_2-norm, and $\mathbf{z}$ is the gradient of f at the current iterate $\mathbf{u}$ within the proximal gradient method.

However, the proximal operator of g (2) is intractable or too expensive in several important problems (c.f., [14, 17, 12, 15]). Generalized conditional gradient (GCG) (a.k.a. Frank-Wolfe (FW)) provides an alternative optimization framework to the (accelerated) proximal gradient by using the tractable (or cheap) linear minimization oracles (LMO), albeit at slower convergence rates [14, 23].

This work considers a General Proximal gradient Method (GPM), using a different operator:

$$\operatorname{prox}_{\mathrm{g}}(\mathbf{u}, \mathbf{z}, L) \in \arg\min_{\mathbf{x}\in\mathbb{R}^p} \mathbf{z}^T\mathbf{x} + \frac{L}{2}\|\mathbf{x}-\mathbf{u}\|^2 + g(\mathbf{x}), \tag{3}$$

where $\|\cdot\|$ is any norm and L is typically chosen as the Lipschitz constant of ∇f with respect to the chosen norm. In stark comparison to the unique solutions to (2), note that (3) can be set valued.

The interest in this generalization stems from the benefit it can entail on the convergence, as already observed in the context of (projected) gradient descent method, for e.g, in [16, 20, 7, 10], affine invariance (c.f., [10]), estimation quality and time-complexity of proximal methods. We derive important cases where (3) is tractable with the proper choice of the norm, whereas (2) is not known to be tractable. In addition, we propose the first—to our knowledge—accelerated proximal optimization framework that handles the composite case (1) with the general proximal operator in (3). Our specific contributions can be summarized as follows:

- We introduce a tractable method, which performs a logarithmic number of linear optimization steps, to approximately compute (3) for a broad class of structure priors g (*cf.*, Sect. 3.1).
- We identify a special class of functions g for which we design an efficient greedy algorithm to compute (3) exactly (c.f., Sect. 3.2). The resulting iterates, in this approach, form a convex combination of only few "atoms", which is a desirable property in several applications.
- We propose an accelerated variant of GPM, accGPM, where we introduce a new type of estimate sequences which allow us to avoid the computation of a proximity/projection operation (c.f., Sect. 4).
- We illustrate our results on the Lasso problem, for which GPM in the ℓ_1-norm yields better estimation quality in a sparse setup and on ℓ_∞-latent group Lasso [25], for which we provide the first efficient proximity operator.

1.1 Preliminaries and notation

We use the set Γ_0 to denote all proper lower semi-continuous convex functions on $\mathbb{R}^p$. We consider problems of the form (1), whose set of minimizers $\mathscr{X}^\star$ is assumed to be non-empty with $f, g \in \Gamma_0$.

We further assume that the gradient of f is L-Lipschitz continuous with respect to $\|\cdot\|$, i.e., $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_* \leq L\|\mathbf{x}-\mathbf{y}\|$, where $\|\cdot\|_*$ is the dual norm of $\|\cdot\|$. This property implies the following majorizer for any $\gamma \in (0, 1/L]$:

$$f(\mathbf{x}) \leq f(\mathbf{y}) + \langle \mathbf{x}-\mathbf{y}, \nabla f(\mathbf{y})\rangle + \frac{1}{2\gamma}\|\mathbf{x}-\mathbf{y}\|^2. \tag{4}$$

A function f is μ-strongly convex with respect to $\|\cdot\|$ if, $\forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^p, \forall \mathbf{p} \in \partial f(\mathbf{y})$, it holds that

$$f(\mathbf{x}) \geq f(\mathbf{y}) + \langle \mathbf{x}-\mathbf{y}, \mathbf{p}\rangle + \frac{\mu}{2}\|\mathbf{x}-\mathbf{y}\|^2. \tag{5}$$

Throughout, $\iota_{\mathcal{X}}$ denotes the indicator function over the set $\mathcal{X}$, where $\iota_{\mathcal{X}}(\mathbf{x}) = 0$, if $\mathbf{x} \in \mathcal{X}$, and ∞ otherwise. The symbol $\circ$ denotes the coordinate-wise multiplication and $\mathbf{A}_S$ denotes the submatrix of $\mathbf{A}$ that corresponds to the columns indexed by S. We use $\operatorname{sign}(\alpha) = \pm 1$ to denote the sign of α, and $\operatorname{sign}(0) = 0$. For a function $f : \mathbb{R}^p \to R \cup \{+\infty\}$, we will denote by f^* its Fenchel conjugate.

2 Generalized proximal gradient method: Warm-up

The general proximal gradient method (GPM) in non-Euclidean norms is the iterative scheme where $\mathbf{x}^{k+1} \in \operatorname{prox}_{\mathrm{g}}(\mathbf{x}^k, \nabla f(\mathbf{x}^k), L)$. For completeness, we state below its basic convergence result.

Theorem 1. *The iterates* $\mathbf{x}^k$ *of GPM satisfy* $\forall k \in \mathbb{N}$*:*

$$F(\mathbf{x}^k) - F^\star \leq \frac{2\max\{\mathcal{R}(\mathbf{x}^0), F(\mathbf{x}^0) - F^\star\}}{k}$$

where $\mathcal{R}(\mathbf{x}^0) = \max_{\{\mathbf{x}:F(\mathbf{x})\leq F(\mathbf{x}^0)\}} \max_{\mathbf{x}^\star\in\mathcal{X}^*} L\|\mathbf{x}-\mathbf{x}^\star\|^2$*. If, in addition,* $f(\mathbf{x})$ *is* μ*-strongly convex w.r.t. norm* $\|\cdot\|$*, then GPM satisfies* $F(\mathbf{x}^k) - F^\star \leq (1-\frac{\mu}{L})^k\big(F(\mathbf{x}^0) - F^\star\big)$.

The convergence rate of GPM depends on the choice of norm, where choosing a non-Euclidean norm can lead in some cases to smaller Lipschitz constant L and level set radius $\mathcal{R}(\mathbf{x}^0)$, as well as larger (restricted) strong convexity constant μ (c.f., Sect. 5.1 and [16, 20, 7]), thus yielding faster convergence.

Theorem 1 is not new; GPM has been analyzed in the context of randomized coordinate descent [26]. However, the primary interest of [26] is the weighted ℓ_2-norm, and the broader tractability question of the non-Euclidean norm choices is not addressed. We fill this gap in Section 3.

3 Tractability of the generalized proximity operator

To our knowledge, the computation of (3) for non-Euclidean norms is not addressed so far, except for the special case where g is the standard simplex constraint and the chosen norm is the ℓ_1-norm [20].

Section 3.1 shows that prox_g can be approximated in polynomial time, for the class of *polyhedral* functions g, if the norm is chosen to be an *atomic* norm $\|\cdot\|_{\mathcal{A}}$ [8]. In Section 3.1, we propose an efficient greedy algorithm to compute prox_g exactly, in the special case where g corresponds to an atomic norm, with *linear independent atoms* and the norm in prox_g is chosen to be the same.

We now introduce a Moreau-like decomposition which relates, as in the Euclidean case, prox_g to the proximity operator of the Fenchel conjugate g^* w.r.t. to the dual norm $\|\cdot\|_*$, denoted by $\text{prox}^*_{g^*}$.

Proposition 1. *Generalized Moreau's decomposition Given* $g \in \Gamma_0$ *and its Fenchel* g^**, we have*

$$\mathbf{p}-\mathbf{z} \in \text{prox}^*_{g^*}(-\mathbf{z}, -\mathbf{u}, 1/L) \text{ and } \mathbf{x}^\star - \mathbf{u} \in -\partial\big(\frac{L}{2}\|\cdot\|_*^2\big)(\mathbf{p}) \cap \big(-\mathbf{u} + \partial g(\mathbf{p}-\mathbf{z})\big) \tag{6}$$

where $\mathbf{p} \in -\partial\big(\frac{L}{2}\|\cdot\|^2\big)(\mathbf{x}^\star - \mathbf{u}) \cap \big(\mathbf{z} + \partial g(\mathbf{x}^\star)\big)$ *and* $\mathbf{x}^\star \in \text{prox}_{\text{g}}(\mathbf{u}, \mathbf{z}, L)$.

The tractability of prox_{g} implies then the tractability of $\text{prox}^*_{\text{g}^*}$ whenever finding an element in the intersection of the two subdifferential sets (6) is easy. Such operation is also required in the acceleration of GPM. Section 4 describes how to find such an element for some examples of interest.

A simple but key observation to our proposed framework is given below:

Lemma 1. *Let* $h(t) = \min_{\|\mathbf{x}-\mathbf{u}\|\leq t} \mathbf{z}^T\mathbf{x} + g(\mathbf{x})$ *and* $t^\star \in \frac{\partial h(t^\star)}{L}$ *then*

$$\mathbf{x}^\star \in \text{prox}_{\text{g}}(\mathbf{u}, \mathbf{z}, L) \Leftrightarrow \mathbf{x}^\star \in \operatorname*{arg\,min}_{\|\mathbf{x}-\mathbf{u}\|\leq t^\star} \mathbf{z}^T\mathbf{x} + g(\mathbf{x}).$$

Computing prox_{g} can be seen then as computing the Fenchel conjugate of g at $-\mathbf{z}$ *locally*, by restricting $\mathbf{x}$ in the norm ball of radius $t^\star$ around $\mathbf{u}$. Hence, we denote this operator by

$$\text{lconj}_{\text{g}}(\mathbf{u}, \mathbf{z}, t) := \operatorname*{arg\,min}_{\|\mathbf{x}-\mathbf{u}\|\leq t} \mathbf{z}^T\mathbf{x} + g(\mathbf{x}).$$

Here, we note a close connection between our local conjugate operator lconj_{g} and the local linear oracle proposed by [12], which corresponds to a relaxation of lconj_{g}, with $g = \iota_P$ for a polytope P.

3.1 Atomic proximity operator of polyhedral functions

In this section, we propose a polynomial time approach to approximately compute prox_g for any *polyhedral* function g, i.e., $P_g := \text{epi}(g)$ is a polytope. Examples where g is a polyhedral functions are abundant, including structure sparsity-inducing norms [4, 24], totally unimodular structure sparsity penalties [11], and atomic norms [8]. For further examples, see [14, 12, 17].

We choose the norm in prox_g to be any atomic norm, i.e., $\|\mathbf{x}\|_{\mathcal{A}} = \inf_{t>0}\{t : \mathbf{x} \in t\ \text{conv}(\mathcal{A})\}$, where the atomic set $\mathcal{A}$ is centrally symmetric with finitely many atoms [8]. We denote the polytope $P_{\mathcal{A}} := \text{conv}(\mathcal{A})$ and the resulting proximity operator by $\text{prox}_g^{\mathcal{A}}$.

Our choice of the atomic norm is motivated by the following observation.

$$h(t) = \min_{\|\mathbf{x}-\mathbf{u}\|_{\mathcal{A}} \leq t} \mathbf{z}^T\mathbf{x} + g(\mathbf{x}) = \min_{\substack{\mathbf{x}-\mathbf{u}\in tP_{\mathcal{A}} \\ (\mathbf{x},y)\in P_g}} \mathbf{z}^T\mathbf{x} + y. \tag{7}$$

Hence, h is a non-increasing piecewise linear function and $h(t)$ can be computed, for any t, by a linear program (LP). We will assume P_g and $P_{\mathcal{A}}$ are solvable polytopes, i.e., they each have a polynomial time seperation oracle.[1] Hence, the LP (7) can be solved in polynomial time. Note that any polytope with a polynomial time LMO also admits a polynomial time separation oracle.

Since $h(t)$ is a non-increasing piecewise-linear function, its subdifferential can be approximated by $\partial h(t) \simeq [\frac{h(t)-h(t+\epsilon)}{\epsilon}, \frac{h(t-\epsilon)-h(t)}{\epsilon}]$ for a small enough $\epsilon > 0$. If t is a differentiable point of $h(t)$, the interval would correspond to a unique value. The optimal $t^\star$ can then be obtained via binary search over the interval $t^\star \in [t_{\min}, t_{\max}]$ where $t_{\min} = \min_{(\mathbf{x},y)\in P_g} \|\mathbf{x}-\mathbf{u}\|_{\mathcal{A}}$ and $t_{\max} = \|\mathbf{x}_{\min}-\mathbf{u}\|_{\mathcal{A}}$ where $\mathbf{x}_{\min} \in \arg\min_{\mathbf{x}} \mathbf{z}^T\mathbf{x} + g(\mathbf{x})$. By Lemma 1, we reach the optimal $t^\star$ when $t^\star \in \frac{\partial h(t)}{L}$. Algorithm 3, given in the Appendix, provides a pseudocode for this approach.

The binary search approach provides us a simple strategy to compute $\text{prox}_g^{\mathcal{A}}$ approximately by a logarithmic number of LPs, for any polyhedral function g, including examples where the standard $\text{prox}_g^{\ell_2}$ is costly. One such prominent example is the ℓ_∞-latent group Lasso for which existing approaches, to our knowledge, to compute $\text{prox}_g^{\ell_2}$ are inefficient.

Note that the convergence analysis we provide in Sect. 2 and Sect. 4 holds only for exact proximity operators. While the study of inexact GPM is straightforward (the gradient method is known to forgive inexact proximal operator calculations), the inexactness must be controlled for its acceleration, which is already a well-studied topic. We will ignore these issues in the sequel.

3.2 Proximity operator of atomic norms with linearly independent atoms

In this section, we consider the special case of polyhedral functions where g is the indicator function of an atomic norm with *linearly independent atoms*, i.e., $g = \iota_{\|\cdot\|_{\mathcal{A}}\leq\lambda}$, where $\mathcal{A} := \{\mathbf{a}_1, \cdots, \mathbf{a}_{2m}\}$, $(\mathbf{a}_i)_1^m$'s are linearly independent and $\mathbf{a}_i = -\mathbf{a}_{m+i}, \forall i = 1, \cdots, m$. To simplify the notation, we use cyclic indexing, i.e., $\mathbf{a}_{2m+i} = \mathbf{a}_i$. For example, for the ℓ_1-norm, $(\mathbf{a}_i)_1^m$ are the standard basis vectors.

We choose the matching norm in prox_g, i.e., $\|\cdot\| = \|\cdot\|_{\mathcal{A}}$. In this case, computing $h(t)$ corresponds to solving a LP over the intersection of the polytope $P_{\mathcal{A}} = \text{conv}(\mathcal{A})$ and its (scaled) translation by $\mathbf{u}$:

$$h(t) = \min_{\|\mathbf{x}-\mathbf{u}\|_{\mathcal{A}} \leq t} \mathbf{z}^T\mathbf{x} + g(\mathbf{x}) = \min_{\substack{\mathbf{x}-\mathbf{u}\in tP_{\mathcal{A}} \\ \mathbf{x}\in\lambda P_{\mathcal{A}}}} \mathbf{z}^T\mathbf{x}. \tag{8}$$

[1] For an input $\mathbf{x}$, a seperation oracle of P either certifies $\mathbf{x} \in P$ or outputs a hyperplane seperating $\mathbf{x}$ from P.

By the definition, we can represent $\mathbf{x} = \sum_{i=1}^{2m} c_i^x \mathbf{a}_i$, where $\mathbf{c}^x \geq 0$ such that $\sum_{i=1}^{2m} c_i^x = \|\mathbf{x}\|_{\mathcal{A}}$.

Lemma 2 shows that only linearly independent atoms are active in such a *unique* decomposition. We call this then a "minimal representation" decomposition and denote it by $\mathbf{c}^x = \mathrm{MR}(\mathbf{x})$.

Lemma 2. *Given* $\mathbf{x} = \sum_{i=1}^{2m} c_i^x \mathbf{a}_i, \mathbf{c}^x \geq 0$, *then* $\sum_{i=1}^{2m} c_i^x = \|\mathbf{x}\|_{\mathcal{A}} \Leftrightarrow \forall i, c_i^x = 0$ *or* $c_{i+m}^x = 0$.

Representing vectors in this fashion allows us to make the following key observation.

Lemma 3. *Given* $\mathbf{x}, \mathbf{y} \in \mathbb{R}^p$, *s.t* $\mathbf{c}^x = \mathrm{MR}(\mathbf{x}), \mathbf{c}^y = \mathrm{MR}(\mathbf{y})$, *we have* $\|\mathbf{x} - \mathbf{y}\|_{\mathcal{A}} = \|\mathbf{c}^x - \mathbf{c}^y\|_1$.

Based on these observations, we present a fast greedy algorithm 1 that computes $\mathrm{prox}_g^{\mathcal{A}}(\mathbf{u}, \mathbf{z}, L)$ exactly and which only requires access to a linear minimization oracle $\mathrm{LMO}_{\mathcal{A}}(\mathbf{z}) \in \arg\min_{\mathbf{a} \in \mathcal{A}} \mathbf{z}^T \mathbf{a}$.

Note first that computing $t_{\min}$ and $t_{\max}$ is easy in this case: $t_{\min} = \min_{\mathbf{x} \in \lambda P_{\mathcal{A}}} \|\mathbf{x} - \mathbf{u}\|_{\mathcal{A}} = \min_{\|\mathbf{c}^x\|_1 \leq \lambda} \|\mathbf{c}^x - \mathbf{c}^u\|_1 = \max\{\|\mathbf{u}\|_{\mathcal{A}} - \lambda, 0\}$ (by lemma 3) and $t_{\max} = \max\{-\mathbf{z}^T \mathbf{a}_{i_{\min}}/L, t_{\min}\}$ where $\mathbf{a}_{i_{\min}} := \mathrm{LMO}_{\mathcal{A}}(\mathbf{z})$ and $-\mathbf{z}^T \mathbf{a}_{i_{\min}}/L$ corresponds to the largest slope of $h(t)$. For simplicity, Algorithm 1 presented here assumes the input is feasible, i.e., $\mathbf{u} \in \lambda P_{\mathcal{A}}$ and $t_{\min} = 0$. This is true for the iterates of GPM, but not for accGPM. The general algorithm is presented in the Appendix.

At a high level, Algorithm 1 acts the following way: Assuming the optimal $t^\star$ is known, the algorithm starts at $\mathbf{u}$ and moves in the direction of the best atom $\mathbf{a}_{i_{\min}}$, i.e., the one with the smallest product $\mathbf{z}^T \mathbf{a}$ (c.f., line 3), until it hits the boundary of one the two polytopes (c.f., lines 6 - 7). If the boundary reached is of $t^* P_{\mathcal{A}} + \mathbf{u}$, we are done. Otherwise, we are at the boundary of $\lambda P_{\mathcal{A}}$.

The algorithm then improves on the solution by moving the largest amount of weight, which will not violate the constraints, from other active atoms to $\mathbf{a}_{i_{\min}}$, starting from the least benefecial active atom in terms of their product with $\mathbf{z}$. The algorithm stops when it runs out of active atoms or it reaches $\|\mathbf{x} - \mathbf{u}\|_{\mathcal{A}} = t^\star$ (c.f., lines 10 -15). Note the similarity with Away step FW [17], which only reduces the weight of the worst active atom.

Note that Algorithm 1 actually minimizes the objective along the path of possible values of $t^* = \|\mathbf{x}^* - \mathbf{u}\|_{\mathcal{A}}$ from $t = 0$ to $t = t_{\max}$. Indeed, the iterates satisfy $\mathbf{x}^k \in \mathrm{lconj_g}(\mathbf{u}, \mathbf{z}, t_u^k), \forall k$, where t_u^k (budget used) and t_l^k (budget left) keep track, respectively, of how far we are from $\mathbf{u}$, $\|\mathbf{x}^k - \mathbf{u}\|_{\mathcal{A}} = t_u^k$ and how far we "guess" we are from the boundary of $t^* P_{\mathcal{A}} + \mathbf{u}$, where the guess of Lt^* corresponds to the current slope of $h(t_u^k)$. Unlike the general case where we are computing $h(t)$ using a black box optimizer, we actually can compute explicitly the slopes of the different pieces of $h(t)$, given by $\mathbf{z}^T \mathbf{a}_{i_{\min}}, 0.5(\mathbf{z}^T \mathbf{a}_{i_{\min}} - \mathbf{z}^T \mathbf{a}_{j_1}), 0.5(\mathbf{z}^T \mathbf{a}_{i_{\min}} - \mathbf{z}^T \mathbf{a}_{j_2}), \cdots, 0.5(\mathbf{z}^T \mathbf{a}_{i_{\min}} - \mathbf{z}^T \mathbf{a}_{j_p})$.

Proposition 2. *Algorithm 1 returns* $\mathbf{x} \in \mathrm{prox}_g^{\mathcal{A}}(\mathbf{u}, \mathbf{z}, L)$ *in* $O(p\mathcal{T} + p \log p)$ *time, where* $\mathcal{T}$ *is the time to compute* $\mathbf{z}^T \mathbf{a}$ *for any atom* $\mathbf{a} \in \mathcal{A}$.

Sketch of Proof Assuming $t^\star$ is guessed correctly, then if the maximal feasible step $\delta_0 = t^\star$, $\mathbf{x}^0$ is optimal. Otherwise $\|x^0\|_{\mathcal{A}} = \lambda$ and there exists an optimal solution $\mathbf{x}^\star$ s.t. $\|\mathbf{x}^\star\|_{\mathcal{A}} = \lambda$ and $\|\mathbf{x}^\star - \mathbf{x}^0\|_{\mathcal{A}} \leq t^\star - \delta_0$. Then by Lemma 3, we can now solve instead: $\min_{\mathbf{c}^x \geq 0} \{\tilde{\mathbf{z}}^T \mathbf{c}^x : \mathbb{1}^T \mathbf{c}^x = \lambda, \|\mathbf{c}^x - \mathbf{c}^{x^0}\|_1 \leq t\}$ where $\tilde{\mathbf{z}}_i = \mathbf{z}^T \mathbf{a}_i$. This has been considered by [12], to obtain a local linear oracle. The rest of our algorithm, i.e., after entering the for loop on line 10, reduces to theirs. We refer the reader to their proof of correctness [12, Lemma 5.2]. The correctness of the search for t^* follows from the correctness of this greedy approach. Finally, it is clear that the most expensive step in Algorithm 1 is the sorting operation on line 5, and hence its time complexity is $O(p\mathcal{T} + p \log p)$.

Remark 1. *If* $g(\mathbf{x}) = \lambda \|\mathbf{x}\|_{\mathcal{A}}$ *where* $\mathcal{A} := \{\mathbf{a}_1, \cdots, \mathbf{a}_{2m}\}$, $(\mathbf{a}_i)_1^m$*'s are linearly independent. Its Fenchel conjugate is given by* $g^*(\mathbf{x}) = \iota_{\{\|\cdot\|_{\mathcal{A}^*} \leq \lambda\}}(\mathbf{x})$, *where* $\|\cdot\|_{\mathcal{A}^*}$ *is the dual norm of* $\|\cdot\|_{\mathcal{A}}$, *then* $\mathrm{prox}_g^{\mathcal{A}}$ *can be obtained by computing* $\mathrm{prox}_{g^*}^{\mathcal{A}^*}$ *via Algorithm 1 and applying Proposition 1.*

Note that Algorithm 1 only adds *one* atom to the set of active atoms of $\mathbf{u}$ and possibly remove others, hence the corresponding iterates in GPM retain "sparsity."

Algorithm 1 Prox of linearly independent atomic norms: $\text{prox}_g^{\mathcal{A}}(\mathbf{u}, \mathbf{z}, L)$

Input: $\mathbf{c}^u = \text{MR}(\mathbf{u})$.
Initialize: $\mathbf{x}^0 = \mathbf{u}, \mathbf{c}^x = \mathbf{c}^u, t_u^0 = 0$.
Let $\mathbf{a}_{i_{\min}} := \text{LMO}_{\mathcal{A}}(\mathbf{z})$
Guess $t_l^0 = \max\{-\mathbf{z}^T\mathbf{a}_{i_{\min}}/L, 0\}$.
Sort $\mathbf{z}^T\mathbf{a}_i$ for active atoms: $\mathbf{z}^T\mathbf{a}_{j_1} \geq \mathbf{z}^T\mathbf{a}_{j_2} \geq \cdots, \forall c_j^u > 0$.
Let $\delta_0 = \max_{\delta>0}\{\delta\colon \mathbf{u} + \delta\mathbf{a}_{j_{\min}} \in \lambda P_{\mathcal{A}} \cap (t_l^0 P_{\mathcal{A}} + \mathbf{u})\} = \min\{t_l^0, \lambda - \|\mathbf{u}\|_{\mathcal{A}} + 2c^u_{i_{\min}+m}\}$
Update $\mathbf{x}^0 = \mathbf{u} + \delta_0\mathbf{a}_{i_{\min}}$.
Update weights: $c^x_{i_{\min}} = \max\{\delta_0 + c^u_{i_{\min}} - c^u_{i_{\min}+m}, 0\}, c^x_{i_{\min}+m} = -\min\{\delta_0 + c^u_{i_{\min}} - c^u_{i_{\min}+m}, 0\}$
Update $t_u^0 = \delta_0, t_l^r = t_l^0 - t_u^0$.
while $k = 1, \cdots, p$ and $t_l^k \geq 0$ **do**
  Update guess $t_l^k = \max\{-0.5\mathbf{z}^T(\mathbf{a}_{i_{\min}} - \mathbf{a}_{j_k})/L - t_u^k, 0\}$.
  Let $\delta_k = \max_{\delta>0}\{\delta\colon \mathbf{x}^{k-1} + \delta(\mathbf{a}_{i_{\min}} - \mathbf{a}_{j_k}) \in \lambda P_{\mathcal{A}} \cap (t_l^k P_{\mathcal{A}} + \mathbf{u})\} = \min\{c^x_{j_k}, t_l^k/2\}$
  Update $\mathbf{x}^k = \mathbf{x}^{k-1} + \delta_k(\mathbf{a}_{i_{\min}} - \mathbf{a}_{j_k})$
  Update $t_l^{k+1} = t_l^k - 2\delta_k$
end while
Return: $\mathbf{x}^k$

4 Accelerated generalized proximal gradient method

In this section, we present an accelerated variant of GPM in Algorithm 2 and show that it has the same convergence rate as fast proximal gradient methods, such as FISTA [6].

The literature is vast on how to accelerate first order methods in non-Euclidean norms [20, 29, 18, 2, 3, 33]. However, unlike accGPM, these schemes require the computation of a proximity/projection operation w.r.t a strongly convex function in each iteration, which imposes the same computational bottleneck of computing $\text{prox}_g^{\ell_2}$. Similar to the classical fast methods, the accGPM introduces a momentum term. However, a novel term $\mathbf{p}^k$ in line 10 of accGPM is essential in our analysis.

Algorithm 2 Accelerated proximal gradient method

Input: $L > 0$, $\mu > 0$, $\mathbf{x}^0 \in \mathbb{R}^p$, $\beta_0 > 0$.
Initialization: $\mathbf{w}^0 = \mathbf{x}^0$, $\mathbf{y}^0 = \mathbf{x}^0$.
for $k = 0, 1, \ldots$ **do**
  $\gamma_k \in (0, 1/L]$, $\alpha_k = \frac{1}{2}\big(\sqrt{\beta_k^2\gamma_k^2 + 4\beta_k\gamma_k} - \beta_k\gamma_k\big)$, $\beta_{k+1} = (1-\alpha_k)\beta_k + \alpha_k\tau_k\mu$
  $\mathbf{y}^{k+1} = (1-\alpha_k)\mathbf{x}^k + \alpha_k\mathbf{w}^k$
  $\mathbf{x}^{k+1} \in \text{prox}(\mathbf{y}^{k+1}, \nabla f(\mathbf{y}^{k+1}), 1/\gamma_k)$
  if $\mathbf{x}^{k+1} = \mathbf{y}^{k+1}$ **then**
    stop
  end if
  $\mathbf{p}^k \in -\partial\big(\frac{1}{2\gamma_k}\|\cdot\|^2\big)(\mathbf{x}^{k+1} - \mathbf{y}^{k+1}) \cap \big(\nabla f(\mathbf{y}^{k+1}) + \partial g(\mathbf{x}^{k+1})\big)$
  $\mathbf{w}^{k+1} = \arg\min_{\mathbf{x}\in\mathbb{R}^p} e_{k+1}(\mathbf{x})$
end for
Return: $\mathbf{x}^{k+1}$

Computation of $\mathbf{p}^k$: When $g = 0$, this term reduces to the gradient of f; $\mathbf{p}^k = \nabla f(\mathbf{y}^{k+1})$. When $\|\cdot\|^2$ or $g(\mathbf{x})$ is differentiable, $\mathbf{p}^k$ is unique. In general, since the subdifferential of any norm can be described by $\partial\|\mathbf{x}\| = \{\mathbf{z} : \mathbf{z}^T\mathbf{x} = \|\mathbf{x}\|, \|\mathbf{z}\|_* \leq 1\}$, then if g and $\|\cdot\|$ are atomic norms, $\mathbf{p}^k$ can be

computed via a linear feasibility problem. In Section 5, we show specifically how to compute $\mathbf{p}^k$ for ℓ_1-norm and ℓ_∞-latent group Lasso norm examples.

In [20], the authors proposed a non-Euclidean projected gradient algorithm to solve the special case of Problem (1) when $g = \iota_{\mathcal{X}}$ for a convex set $\mathcal{X}$. The analysis of this scheme is based on the concept of estimate sequences (c.f., [5, 22]). Algorithm 2 solves Problem 1 in the general setting and for non-Euclidean norms, by constructing a novel estimate sequence e_k defined as follows.

Definition 1. *Let $(\alpha_k)_{k\in\mathbb{N}}$, $(\tau_k)_{k\in\mathbb{N}}$ and $(\gamma_k)_{k\in\mathbb{N}}$ be sequences in $(0,+\infty)$ and let $(\mathbf{x}^k)_{k\in\mathbb{N}}$, $(\mathbf{y}^k)_{k\in\mathbb{N}}$ and $(\mathbf{p}^k)_{k\in\mathbb{N}}$ be sequences in $\mathbb{R}^p$. We define the estimate sequence e_k recursively $e_0 := \dfrac{\beta_0}{\sigma}d + F(\mathbf{x}^0)$ and $e_{k+1} := (1-\alpha_k)e_k + \alpha_k\big((1-\tau_k)\psi_k + \tau_k\phi_k\big)$, where $\psi_k := F(\mathbf{x}^{k+1}) + \langle \cdot - \mathbf{x}^{k+1}, \mathbf{p}^k\rangle - \frac{1}{2\gamma_k}\|\mathbf{x}^{k+1} - \mathbf{y}^{k+1}\|^2$ and $\phi_k := f(\mathbf{y}^{k+1}) + \big\langle \cdot - \mathbf{y}^{k+1}, \nabla f(\mathbf{y}^{k+1})\big\rangle + g$. and where the prox-function d is σ-strongly convex with respect to $\|\cdot\|$ and $\mathbf{x}^0 = \arg\min_{\mathbf{x}\in\mathbb{R}^p} d(\mathbf{x})$, assuming without loss of generality that $d(\mathbf{x}^0) = 0$.*

Note that the parameter τ_k allows us to choose between ψ_k and ϕ_k, depending on which is more suitable to the problem at hand. The estimate sequence resulting from ϕ_k ($\tau_k = 1$) is a direct extension of the one considered in [20]. If g is strongly convex, this type of estimate sequence is preferable as it can exploit strong convexity, leading to a linear rate (c.f., Theorem 1). However, this approach requires the minimization of a proximal-type subproblem involving the strongly-convex function d (c.f., line 11 in Algo 2), which we will avoid in this paper. In fact, if $d = \frac{1}{2}\|\cdot\|_2^2$, this subproblem reduces to the Euclidean prox of g. On the other hand, choosing instead the novel estimate sequence resulting from ψ_k ($\tau_k = 0$) avoids such expensive subroutine. If the prox-function is chosen to be $d = \frac{1}{2}\|\cdot\|_q^2, 1 < q \leq 2$, $\mathbf{w}^{k+1}$ can be computed in closed-form solution.

Theorem 2. *Consider Problem 1 where g is μ-strongly convex w.r.t. $\|\cdot\|$. If accGPM terminates at iteration k, i.e., $\mathbf{x}^{k+1} = \mathbf{y}^{k+1}$, then $\mathbf{x}^{k+1}$ is a solution to (1). Otherwise, let $x^\star \in \mathcal{X}^\star$, the iterates of accGPM satisfy the following.*

1. *If $\mu = 0$. Then $\forall k \in \mathbb{N}$, we have $F(x^{k+1}) - F^\star \leq \dfrac{4\big(\sigma(F(x^0) - F^\star) + \beta_0 d(x^\star)\big)}{\sigma\{2 + \sqrt{\beta_0}\sum_{i=0}^k \sqrt{\gamma_i}\}^2}$.*

   Consequently, if $\forall k \in \mathbb{N}, \gamma_k = 1/L$, then $F(x^{k+1}) - F^\star \leq \dfrac{4L\big(\sigma(F(x^0) - F^\star) + \beta_0 d(x^\star)\big)}{\sigma\{2\sqrt{L} + \sqrt{\beta_0}(k+1)\}^2}$.

2. *If $\mu > 0$. Set $\tau = \inf_{k\in\mathbb{N}} \tau_k$, and $\rho = \frac{\tau\mu}{2L}\left\{\sqrt{1 + \frac{4L}{\beta_0+\mu}} - 1\right\}$. If $\beta_0 \geq \tau\mu$ and $\forall k \in \mathbb{N}, \gamma_k = 1/L$, then we have $F(x^{k+1}) - F^\star \leq (1-\rho)^{k+1}\{F(x^0) - F^\star + \frac{\beta_0}{\sigma}d(x^\star)\}$.*

Note that the choice of the norm in accGPM affects the Lipschitz constant L as in GPM, but also affects implicity the term $d(\mathbf{x}^\star)/\sigma$.

5 Numerical Illustration

The purpose of this experimental section is to demonstrate how choosing a non-Euclidean norm in GPM leads in some cases to better estimation quality and in others to easier-to-solve proximity operators. To that end, we consider in Section 5.1, the classical Lasso problem [28] and illustrate how ℓ_1-GPM improves the learning quality. Then, in Section 5.2, we consider the latent group Lasso problem [25] and illustrate how our results yield an *efficient* proximity operator of the ℓ_∞-LGL norm.

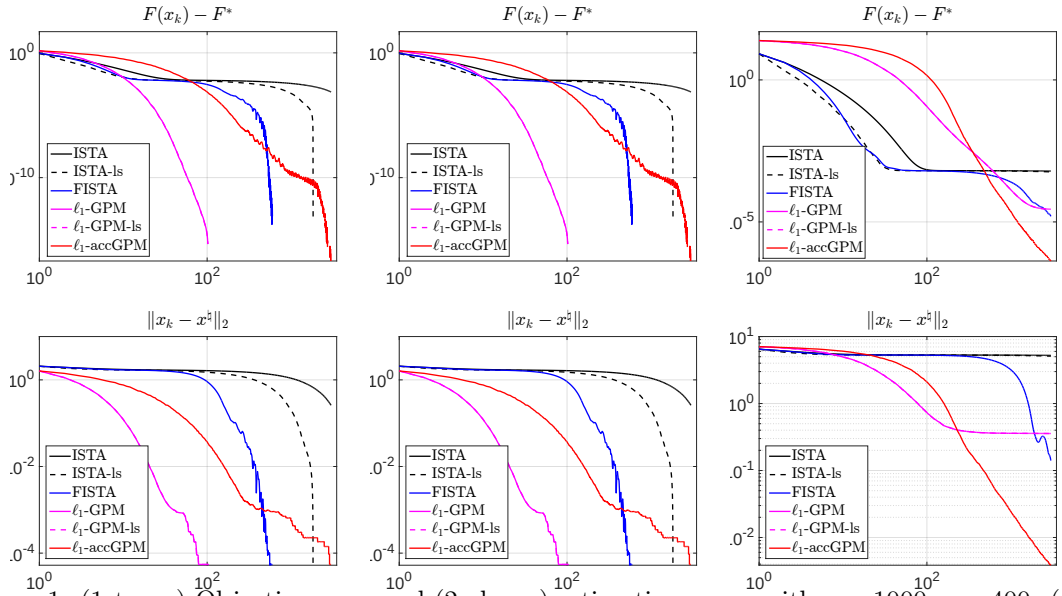

Figure 1: (1st row) Objective error and (2nd row) estimation error, with $p = 1000, n = 400$: (*Left*) $s = 10$, (*Middle*) $s = 50$, (*Right*) $s = 100$.

5.1 Sparse Linear Regression

In this section, we consider the classical Lasso problem [28]: $\min_{\mathbf{x}\in\mathbb{R}^p} \frac{1}{2n}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda\|\mathbf{x}\|_1$. We propose to solve it with ℓ_1-GPM, i.e., with $\text{prox}_{\ell_1}^{\ell_1}$. The motivation of this choice is two folds. The resulting iterates from $\text{prox}_{\ell_1}^{\ell_1}$ are *sparse* (c.f., Sect. 3.2) which is naturally preferred in this set-up. Also, the *Restricted Strong Convexity* parameter, which governs the learning quality of Lasso problems is known to be better w.r.t. the ℓ_1-norm vs the ℓ_2-norm [31], implying stronger estimation guarantees (c.f., Appendix for more details). Our experiment verifies this theoretical claim.

The standard $\text{prox}_{\ell_1}^{\ell_2}$ can be computed in $O(p)$ using the so-called soft thresholding operator [6]. By Remark 1, $\text{prox}_{\ell_1}^{\ell_1}$ can be solved in $O(p \log p)$ time by the greedy algorithm 1 and the decomposition in Prop. 1. We choose instead to solve it directly via another greedy algorithm, of the same "flavor" as Algorithm 1, presented in the Appendix. The momentum $\mathbf{p}^k$ for accGPM has a closed form solution in this case, given in the Appendix.

We synthetically set up a linear model $\mathbf{b} = \mathbf{A}\mathbf{x}^\natural + \mathbf{w}$, where $\mathbf{x}^\natural$ is an s-sparse vector with normalized ℓ_1-norm. $\mathbf{A} \in \mathbb{R}^{n\times p}$ is an i.i.d Gaussian matrix and $\mathbf{w}$ an i.i.d. Gaussian noise vector of variance σ^2 where $\sigma = 10^{-4}$. We fix $p = 1000$, $n = 400$, and vary the sparsity level s from 10 to 100. The number of samples is chosen to exceed the sample complexity [19], while approaching to the statistical phase transition as sparsity increases. The regularization parameter is set to $\lambda = \sigma\sqrt{\frac{\log p}{n}}$ according to the theory of [19].

We compare ISTA and FISTA to ℓ_1-GPM and ℓ_1-accGPM (with $\tau = 0$). Figure 5.1 plots (in logscale) the objective error and estimation error, in the different sparsity setups. We use an accuracy based stopping condition with `tol`$= 10^{-9}$ where the optimal objective value is obtained by `cvx`. We also use a 3000 iteration limit. We equip both ISTA and ℓ_1-GPM with line-search. We use $d(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_{1+\epsilon}^2$ as the prox-function, in Definition 1, for the ℓ_1-accGPM. Such function is strongly convex in the ℓ_1-norm, with $\sigma = \epsilon/p^{\frac{2\epsilon}{1+\epsilon}}$. We set $\epsilon = 0.03$ to maximize σ. Unfortunately, the dimension-dependence of σ leads to a slower convergence of ℓ_1-accGPM, as observed in 5.1. Otherwise, a clear advantage of sparse updates in the sparse regime can be inferred from the leftmost pair, where ℓ_1-GPM significantly outperforms the classical ISTA/FISTA. As the sparsity level increases, the benefits of sparse updates

vanish (mid pair), and around the phase transition classical gradient methods perform better.

5.2 Latent group Lasso

In this section, we consider the latent group Lasso (LGL) problem: $\min_{\mathbf{x}\in\mathbb{R}^p} \frac{1}{2}\|\mathbf{A}\mathbf{x}-\mathbf{b}\|_2^2 + \lambda\|\mathbf{x}\|_{\mathcal{G}}$, where $\|\mathbf{x}\|_{\mathcal{G}}$ is the LGL-norm, proposed by [25] to induce supports that corresponds to union of groups. Given a collection of groups $\mathcal{G} : \{G_1, \cdots, G_M\}$, the ℓ_q-LGL norm is given by $\|\mathbf{x}\|_{\mathcal{G}} = \min_{\mathbf{v}}\{\sum_{i=1}^M \|\mathbf{v}_{G_i}\|_q : |x| = \sum_{i=1}^M v_{G_i}, \mathrm{supp}(\mathbf{v}_{G_i}) \subseteq G_i\}$. It is known that ℓ_q-LGL is an atomic norm, with atoms $\mathcal{A} = \{\mathbf{v} \in \mathbb{R}^p : \mathrm{supp}(\mathbf{v}_{G_i}) \subseteq G_i, \|\mathbf{v}_{G_i}\|_q \leq 1\}$ [25]. We focus on the case with finitely many atoms where $q = \infty$. The ℓ_∞-LGL is of particular interest as it corresponds to the convex envelope of the set cover function over the unit ℓ_∞-ball [11, 24].

To the best of our knowledge, the only available approaches to compute the standard prox of ℓ_∞-LGL, i.e., $\mathrm{prox}_{\mathcal{G}}^{\ell_2}$, is either via duplicating the variables in the overlapping groups, which is very inefficient for groups with substantial overlap, or via the cyclic projections approach proposed in [32], which is guaranteed to converge but with no convergence rate guarantees. Our approach to circumvent the difficulty of $\mathrm{prox}_{\mathcal{G}}^{\ell_2}$ is to solve instead LGL with ℓ_∞-GPM, i.e., with $\mathrm{prox}_{\mathcal{G}}^{\ell_\infty}$. Note that ℓ_∞-LGL satisfies the assumptions in Section 3.1 and hence $\mathrm{prox}_{\mathcal{G}}^{\ell_\infty}$ can be solved via Algorithm 3 and its $\mathbf{p}_k$ can be computed by a feasibility LP, as detailed in Table 1. We use `Gurobi` to solve the resulting LPs. We choose $d(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|_2^2$ as the prox-function.

Table 1: Running time (in sec) of $\mathrm{prox}_{\mathcal{G}}^{\ell_2}$ (LHS) and $\mathrm{prox}_{\mathcal{G}}^{\ell_\infty} + \mathbf{p}_k$ (RHS), averaged over 10 runs.

	p = 64		p = 128		p = 256		p = 512	
`tol`= 10^{-2}	**0.055**	0.055 + 0.003	**0.103**	0.137+ 0.005	**0.192**	0.247 + 0.009	**0.461**	0.714 + 0.016
`tol`= 10^{-3}	0.502	**0.101 + 0.003**	0.944	**0.149 + 0.004**	2.038	**0.360 + 0.008**	4.213	**1.276 + 0.013**
`tol`= 10^{-4}	5.234	**0.252 + 0.006**	9.422	**0.203 + 0.004**	18.92	**0.460 + 0.006**	41.21	**1.857 + 0.016**
`tol`= 10^{-5}	42.62	**0.214 + 0.005**	98.13	**0.428 + 0.009**	170.6	**0.614 + 0.006**	377.5	**1.487 + 0.015**

We first assess the time complexity of the proximity operator $\mathrm{prox}_{\mathcal{G}}^{\ell_\infty}$ vs $\mathrm{prox}_{\mathcal{G}}^{\ell_2}$. We fix the size of the groups to $|G_i| = 10$ and generate $M = 2.5p/10$ (to ensure substantial overlap) groups with randomly selected elements. The input $\mathbf{u} \in \mathbb{R}^p$ is generated as a random Gaussian vector. For fairness, we set $\lambda = 0.8 \min_i \|\mathbf{u}_{G_i}\|_1$ to ensure all groups are active. We report, in a table in the Appendix, the CPU time (in sec) of $\mathrm{prox}_{\mathcal{G}}^{\ell_\infty}$ and $\mathrm{prox}_{\mathcal{G}}^{\ell_2}$, as we vary the dimension p from 64 to 512 and the accuracy `tol` from 10^{-2} to 10^{-5}, where a true solution is obtained via `cvx`. $\mathrm{prox}_{\mathcal{G}}^{\ell_\infty}$ provides up to 300× speed up.

To assess if the slow performance of $\mathrm{prox}_{\mathcal{G}}^{\ell_2}$ is compensated by a better convergence rate, we compare the performance of FISTA to ℓ_∞-accGPM on a synthetic learning problem, where the true vector $x^\natural$ is given by the union of $s = 2$ randomly selected groups. We follow otherwise the same setup as in Section 5.1, with $p = 100, n = 50$ and the groups generated as before. We stop both prox algorithms after 10^5 iterations, or when the distance between iterates reaches a precision, initialized to 10^{-5} and decreased linearly with iterations. For the outer algorithms, we use an accuracy based stopping condition with `tol`= 10^{-9} where the optimal objective value is obtained by `cvx`. We also use a 5000 iteration limit. We choose the regularization parameter λ that yields the best performance on the `cvx` solution. Figure 5.2 plots (in logscale) the objective error and optimization error. FISTA indeed has a better convergence rate in this case, but this is undermined by the slow performance of $\mathrm{prox}_{\mathcal{G}}^{\ell_2}$. Indeed, with the set iteration limit, $\mathrm{prox}_{\mathcal{G}}^{\ell_2}$ is not able to reach the requested precision, and thus FISTA doesn't converge to the true solution.

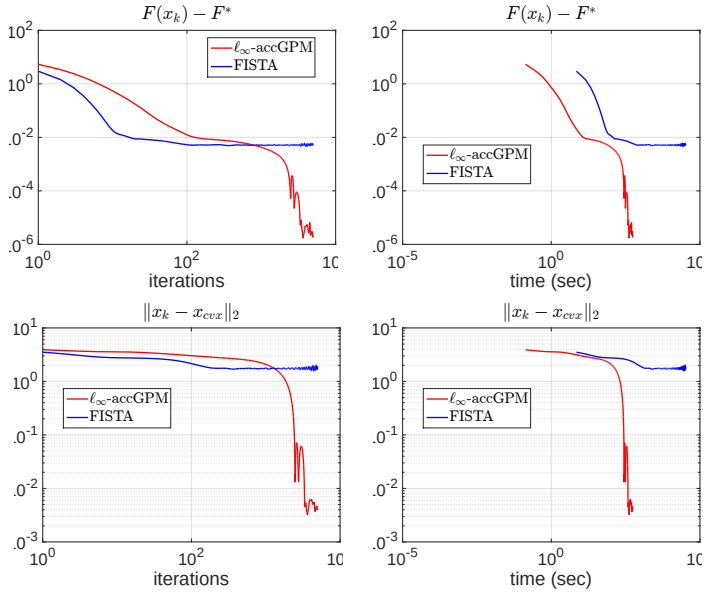

Figure 2: Objective error (top) and optimization error (bottom), for $p = 100, n = 50$ and $s = 2$.

Acknowledgments

We would like to thank Yu-Chun Kao for useful discussions. This work was supported in part by the European Commission under ERC Future Proof, SNF 200021-146750, SNF CRSII2-147633, NCCR Marvel.

References

[1] Alekh Agarwal, Sahand Negahban, Martin J Wainwright, et al. Fast global convergence of gradient methods for high-dimensional statistical recovery. *The Annals of Statistics*, 40(5):2452–2482, 2012.

[2] Masoud Ahookhosh. Accelerated first-order methods for large-scale convex minimization. *arXiv preprint arXiv:1604.08846*, 2016.

[3] Zeyuan Allen-Zhu and Lorenzo Orecchia. Linear coupling: An ultimate unification of gradient and mirror descent. *arXiv preprint arXiv:1407.1537*, 2014.

[4] F. Bach. Structured sparsity-inducing norms through submodular functions. In *NIPS*, pages 118–126, 2010.

[5] Michel Baes. Estimate sequence methods: extensions and approximations. *Institute for Operations Research, ETH, Zürich, Switzerland*, 2009.

[6] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):183–202, 2009.

[7] Claire Boyer, Pierre Weiss, and Jérémie Bigot. An algorithm for variable density sampling with block-constrained acquisition. *SIAM Journal on Imaging Sciences*, 7(2):1080–1107, 2014.

[8] Venkat Chandrasekaran, Benjamin Recht, Pablo A Parrilo, and Alan S Willsky. The convex geometry of linear inverse problems. *Foundations of Computational mathematics*, 12(6):805–849, 2012.

[9] Ioana Cioranescu. *Geometry of Banach spaces, duality mappings and nonlinear problems*, volume 62 of *Mathematics and its Applications*. Kluwer Academic Publishers Group, Dordrecht, 1990.

[10] Alexandre d'Aspremont, Cristóbal Guzmán, and Martin Jaggi. An optimal affine invariant smooth minimization algorithm. *arXiv preprint arXiv:1301.0465*, 2013.

[11] M. El Halabi and V. Cevher. A totally unimodular view of structured sparsity. *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, pp. 223–231*, 2015.

[12] Dan Garber and Elad Hazan. A linearly convergent variant of the conditional gradient algorithm under strong convexity, with applications to online and stochastic optimization. *SIAM Journal on Optimization*, 26(3):1493–1528, 2016.

[13] Osman Güler. New proximal point algorithms for convex minimization. *SIAM J. Optim.*, 2(4):649–664, 1992.

[14] Martin Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 427–435, 2013.

[15] Anatoli Juditsky and Arkadi Nemirovski. Solving variational inequalities with monotone operators on domains given by linear minimization oracles. *Mathematical Programming*, 156(1-2):221–256, 2016.

[16] Jonathan A Kelner, Yin Tat Lee, Lorenzo Orecchia, and Aaron Sidford. An almost-linear-time algorithm for approximate max flow in undirected graphs, and its multicommodity generalizations. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 217–226. SIAM, 2014.

[17] Simon Lacoste-Julien and Martin Jaggi. On the global linear convergence of frank-wolfe optimization variants. In *Advances in Neural Information Processing Systems*, pages 496–504, 2015.

[18] Guanghui Lan. An optimal method for stochastic composite optimization. *Mathematical Programming*, 133(1-2):365–397, 2012.

[19] Sahand N Negahban, Pradeep Ravikumar, Martin J Wainwright, Bin Yu, et al. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. *Statistical Science*, 27(4):538–557, 2012.

[20] Yu Nesterov. Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1):127–152, 2005.

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

[22] Yurii Nesterov. *Introductory lectures on convex optimization*, volume 87 of *Applied Optimization*. Kluwer Academic Publishers, Boston, MA, 2004. A basic course.

[23] Yurii Nesterov et al. Complexity bounds for primal-dual methods minimizing the model of objective function. *Center for Operations Research and Econometrics, CORE Discussion Paper*, 2015.

[24] G. Obozinski and F. Bach. Convex relaxation for combinatorial penalties. *arXiv preprint arXiv:1205.1240*, 2012.

[25] Guillaume Obozinski, Laurent Jacob, and Jean-Philippe Vert. Group lasso with overlaps: the latent group lasso approach. *arXiv preprint arXiv:1110.0413*, 2011.

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

[27] Shai Shalev-Shwartz and Yoram Singer. Online learning: Theory, algorithms, and applications. Technical report, Hebrew University, 2007.

[28] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[29] Paul Tseng. On accelerated proximal gradient methods for convex-concave optimization. submitted to siam j. *J. Optim*, 2008.

[30] Sara van de Geer and Alan Muro. On higher order isotropy conditions and lower bounds for sparse quadratic forms. *Electronic Journal of Statistics*, 8(2):3031–3061, 2014.

[31] Sara A Van De Geer and Peter Bühlmann. On the conditions used to prove oracle results for the lasso. *Electronic Journal of Statistics*, 3:1360–1392, 2009.

[32] Silvia Villa, Lorenzo Rosasco, Sofia Mosci, and Alessandro Verri. Proximal methods for the latent group lasso penalty. *Computational Optimization and Applications*, 58(2):381–407, 2014.

[33] Andre Wibisono, Ashia C Wilson, and Michael I Jordan. A variational perspective on accelerated methods in optimization. *arXiv preprint arXiv:1603.04245*, 2016.

[34] Fu Chun Yang, Zhou Wei, and Dong Wang. Subdifferential representation of homogeneous functions and extension of smoothness in banach spaces. *Acta Mathematica Sinica, English Series*, 26(8):1535–1544, 2010.

[35] Ian En-Hsu Yen, Cho-Jui Hsieh, Pradeep K Ravikumar, and Inderjit S Dhillon. Constant nullspace strong convexity and fast convergence of proximal methods under high-dimensional settings. In *Advances in Neural Information Processing Systems*, pages 1008–1016, 2014.

[36] Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[37] Constantin Zalinescu. *Convex analysis in general vector spaces*. World scientific, 2002.

6 Appendix

6.1 Proof of Theorem 1

Theorem 1. *The iterates $\mathbf{x}^k$ of GPM satisfy $\forall k \in \mathbb{N}$:*

$$F(\mathbf{x}^k) - F^\star \leq \frac{2\max\{\mathcal{R}(\mathbf{x}^0), F(\mathbf{x}^0) - F^\star\}}{k}$$

where $\mathcal{R}(\mathbf{x}^0) = \max_{\{\mathbf{x}: F(\mathbf{x}) \leq F(\mathbf{x}^0)\}} \max_{\mathbf{x}^\star \in \mathcal{X}^*} L\|\mathbf{x} - \mathbf{x}^\star\|^2$. If, in addition, $f(\mathbf{x})$ is μ-strongly convex w.r.t. norm $\|\cdot\|$, then GPM satisfies $F(\mathbf{x}^k) - F^\star \leq (1 - \frac{\mu}{L})^k \big(F(\mathbf{x}^0) - F^\star\big)$.

Proof. Without loss of generality, we assume that f is μ-strongly convex with $\mu \in [0, +\infty)$ (the case when $\mu = 0$ corresponds to the fact that f is convex). Fix $\mathbf{x}^\star$ a minimizer of F and $k \in \mathbb{N}$. If $\mathbf{x}^k$ is a minimizer of F then the claims are trivial. Otherwise, let us define

$$(\forall \mathbf{x} \in \mathbb{R}^p) \quad Q(\mathbf{x}, \mathbf{x}^k) = f(\mathbf{x}^k) + \big\langle \mathbf{x} - \mathbf{x}^k, \nabla f(\mathbf{x}^k) \big\rangle + g(\mathbf{x}) + \frac{L}{2}\|\mathbf{x} - \mathbf{x}^k\|^2. \tag{9}$$

Then

$$\mathbf{x}^{k+1} \in \operatorname{prox}_g(\nabla f(\mathbf{x}^k), \mathbf{x}^k, L) = \operatorname*{arg\,min}_{\mathbf{x} \in \mathbb{R}^p} Q(\mathbf{x}, \mathbf{x}^k). \tag{10}$$

Since the gradient f is L-Lipschitz continuous,

$$f(\mathbf{x}^{k+1}) \leq f(\mathbf{x}^k) + \big\langle \mathbf{x}^{k+1} - \mathbf{x}^k, \nabla f(\mathbf{x}^k) \big\rangle + \frac{L_1}{2}\|\mathbf{x}^{k+1} - \mathbf{x}^k\|^2 \tag{11}$$

and hence (10) yields

$$F(\mathbf{x}^k) = Q(\mathbf{x}^k, \mathbf{x}^k) \geq Q(\mathbf{x}^{k+1}, \mathbf{x}^k) \geq F(\mathbf{x}^{k+1}). \tag{12}$$

By strongly convexity of f we have,

$$(\forall \mathbf{x} \in \mathbb{R}^p) \quad f(\mathbf{x}^k) + \big\langle \mathbf{x} - \mathbf{x}^k, \nabla f(\mathbf{x}^k) \big\rangle \leq f(\mathbf{x}) - \frac{\mu}{2}\|\mathbf{x} - \mathbf{x}^k\|^2. \tag{13}$$

Also by strongly convexity of F and by lemma 13 in [27] we have

$$(\forall \alpha \in [0,1]) \quad F(\alpha\mathbf{x}^\star + (1-\alpha)\mathbf{x}^k) \leq \alpha F(\mathbf{x}^\star) + (1-\alpha)F(\mathbf{x}^k) - \frac{\alpha(1-\alpha)\mu_1}{2}\|\mathbf{x}^\star - \mathbf{x}^k\|^2. \tag{14}$$

It hence follows from (10), (13), and (14) that

$$\begin{aligned}
Q(\mathbf{x}^{k+1}, \mathbf{x}^k) &= \min_{\mathbf{x} \in \mathbb{R}^p} \; f(\mathbf{x}^k) + \big\langle \mathbf{x} - \mathbf{x}^k, \nabla f(\mathbf{x}^k) \big\rangle + g(\mathbf{x}) + \frac{L}{2}\|\mathbf{x} - \mathbf{x}^k\|^2 \\
&\leq \min_{\mathbf{x} \in \mathbb{R}^p} \; F(\mathbf{x}) + \frac{L-\mu}{2}\|\mathbf{x} - \mathbf{x}^k\|^2 \\
&\leq \min_{\alpha \in [0,1]} \; F(\alpha\mathbf{x}^\star + (1-\alpha)\mathbf{x}^k) + \frac{(L-\mu)\alpha^2}{2}\|\mathbf{x}^\star - \mathbf{x}^k\|^2 \\
&\leq \min_{\alpha \in [0,1]} \; \alpha F(\mathbf{x}^\star) + (1-\alpha)F(\mathbf{x}^k) - \frac{\alpha(1-\alpha)\mu_1}{2}\|\mathbf{x}^\star - \mathbf{x}^k\|^2 + \frac{(L_1-\mu_1)\alpha^2}{2}\|\mathbf{x}^k - \mathbf{x}^\star\|^2. \\
&\leq \min_{\alpha \in [0,1]} \; F(\mathbf{x}^k) + \alpha\big(F(\mathbf{x}^\star) - F(\mathbf{x}^k)\big) - \frac{\alpha(1-\alpha)\mu - (L-\mu)\alpha^2}{2}\|\mathbf{x}^\star - \mathbf{x}^k\|^2.
\end{aligned} \tag{15}$$

For $\mu = 0$, the function in (15) admits a minimizer at

$$\alpha_k^* = \min\{\frac{F(\mathbf{x}^k) - F^\star}{L\|\mathbf{x}^k - \mathbf{x}^\star\|^2}, 1\} \in [0, 1], \tag{16}$$

we deduce from (15) that

$$Q(\mathbf{x}^{k+1}, \mathbf{x}^k) - F^\star \leq \max\left\{1 - \frac{F(\mathbf{x}^k) - F^\star}{2L\|\mathbf{x}^k - \mathbf{x}^\star\|^2}, \frac{1}{2}\right\}\big(F(\mathbf{x}^k) - F^\star\big). \tag{17}$$

Consequently, (12) yields

$$F(\mathbf{x}^{k+1}) - F^\star \leq Q(\mathbf{x}^{k+1}, \mathbf{x}^k) - F^\star \leq \left(1 - \frac{F(\mathbf{x}^k) - F^\star}{\rho}\right)\big(F(\mathbf{x}^k) - F^\star\big). \tag{18}$$

Let $a_k = F(\mathbf{x}^k) - F^\star$. Since $a_k - a_{k+1} \geq \frac{a_k^2}{\rho}$, we obtain

$$\frac{1}{a_{k+1}} - \frac{1}{a_k} = \frac{a_k - a_{k+1}}{a_k a_{k+1}} \geq \frac{a_k^2}{\rho a_k^2} = \frac{1}{\rho}. \tag{19}$$

Consequently, $a_k \leq \frac{\rho}{k}$, which proves the first claim. For the second claim, we note that

$$(\forall \mathbf{x} \in \mathbb{R}^p) \quad \frac{\mu}{2}\|\mathbf{x} - \mathbf{x}^\star\|^2 \leq f(\mathbf{x}) - f^\star \leq \frac{L}{2}\|\mathbf{x} - x^\star\|^2 \tag{20}$$

and hence $\alpha_k^* = \dfrac{\mu_1}{L_1} \in (0, 1]$. It then follows from (15) that $Q(\mathbf{x}^{k+1}, \mathbf{x}^k) \leq F(\mathbf{x}^k) - \alpha_k^*\big(F(\mathbf{x}^k) - F^\star\big)$, and hence,

$$F(\mathbf{x}^{k+1}) - F^\star \leq (1 - \alpha_k^*)\big(F(\mathbf{x}^k) - F^\star\big) = \left(1 - \frac{\mu}{L}\right)\big(F(\mathbf{x}^k) - F^\star\big). \tag{21}$$

□

6.2 Proof of Proposition 1

Proposition 1. *Generalized Moreau's decomposition Given $g \in \Gamma_0$ and its Fenchel g^*, we have*

$$\mathbf{p} - \mathbf{z} \in \mathrm{prox}_{\mathrm{g}^*}^*(-\mathbf{z}, -\mathbf{u}, 1/L) \text{ and } \mathbf{x}^\star - \mathbf{u} \in -\partial\big(\frac{L}{2}\|\cdot\|_*^2\big)(\mathbf{p}) \cap \big(-\mathbf{u} + \partial g(\mathbf{p} - \mathbf{z})\big) \tag{6}$$

where $\mathbf{p} \in -\partial\big(\frac{L}{2}\|\cdot\|^2\big)(\mathbf{x}^\star - \mathbf{u}) \cap \big(\mathbf{z} + \partial g(\mathbf{x}^\star)\big)$ and $\mathbf{x}^\star \in \mathrm{prox}_{\mathrm{g}}(\mathbf{u}, \mathbf{z}, L)$.

Proof. Recall that $\mathbf{y} \in \partial f(\mathbf{x}) \Leftrightarrow \mathbf{x} \in \partial f^*(\mathbf{y})$ for any $f \in \Gamma_0$ and its fenchel conjugate f^*. Then since the fenchel conjugate of $-\frac{L}{2}\|\cdot\|^2$ is given by $-\frac{1}{2L}\|\cdot\|_*^2$, we have

$$\begin{aligned}
&\mathbf{x}^\star - \mathbf{u} \in \partial\big(-\frac{L}{2}\|\cdot\|_*^2\big)(\mathbf{p}) \\
&\quad \mathbf{x}^\star \in \partial g(\mathbf{p} - \mathbf{z}) \\
\Leftrightarrow\ &\mathbf{x}^\star - \mathbf{u} \in \partial\big(-\frac{L}{2}\|\cdot\|_*^2\big)(\mathbf{p} - \mathbf{z} + \mathbf{z}) \cap \big(-\mathbf{u} + \partial g(\mathbf{p} - \mathbf{z})\big) \\
\Leftrightarrow\ &\mathbf{p} - \mathbf{z} \in \mathrm{prox}_{\mathrm{g}^*}^*(-\mathbf{z}, -\mathbf{u}, 1/L)
\end{aligned}$$

□

6.3 Proof of Lemma 1

Lemma 1. *Let* $h(t) = \min_{\|\mathbf{x}-\mathbf{u}\| \leq t} \mathbf{z}^T\mathbf{x} + g(\mathbf{x})$ *and* $t^\star \in \frac{\partial h(t^\star)}{L}$ *then*

$$\mathbf{x}^\star \in \mathrm{prox}_g(\mathbf{u}, \mathbf{z}, L) \Leftrightarrow \mathbf{x}^\star \in \operatorname*{arg\,min}_{\|\mathbf{x}-\mathbf{u}\| \leq t^\star} \mathbf{z}^T\mathbf{x} + g(\mathbf{x}).$$

Proof. The two problems are related in the following way:

$$\begin{aligned}
&\min_{\mathbf{x}\in\mathbb{R}^p} \mathbf{z}^T\mathbf{x} + \frac{L}{2}\|\mathbf{x}-\mathbf{u}\|^2 + g(\mathbf{x}) \\
&= \min_{t\geq 0} \frac{L}{2}t^2 + \min_{\|\mathbf{x}-\mathbf{u}\| \leq t} \mathbf{z}^T\mathbf{x} + g(\mathbf{x}) \\
&= \min_{t\geq 0} \frac{L}{2}t^2 + h(t)
\end{aligned}$$

The lemma follows by optimality conditions. □

6.4 Atomic proximity operator of polyhedral functions

Algorithm 3 Atomic prox of polyhedral functions

Input: $t_{\min} > 0, t_{\max} > 0, \delta > 0, \epsilon > 0$
while $|t_{\max} - t_{\min}| > \delta$ **do**
  $t = (t_{\min} + t_{\max})/2$;
  $\mathrm{slope}_1 = \frac{h(t)-h(t+\epsilon)}{\epsilon}$
  $\mathrm{slope}_2 = \frac{h(t-\epsilon)-h(t)}{\epsilon}$
  if $\mathrm{slope}_1 \leq Lt \leq \mathrm{slope}_2$ **then**
    break
  else if $t - \mathrm{slope}_1/L > 0$ **then**
    $t_{\max} = t$
  else
    $t_{\min} = t$
  end if
end while
Return: $\mathbf{x}^{k+1} \in \arg\min_{\|\mathbf{x}-\mathbf{u}\|_{\mathcal{A}} \leq t} \mathbf{z}^T\mathbf{x} + g(\mathbf{x})$

6.5 Proof of Lemma 2

Lemma 2. *Given* $\mathbf{x} = \sum_{i=1}^{2m} c_i^x \mathbf{a}_i, \mathbf{c}^x \geq 0$, *then* $\sum_{i=1}^{2m} c_i^x = \|\mathbf{x}\|_{\mathcal{A}} \Leftrightarrow \forall i, c_i^x = 0$ *or* $c_{i+m}^x = 0$.

Proof. Assume towards contradiction that $\exists i'$, such that $c_{i'}^x \neq 0, c_{i'+m}^x \neq 0$, then let $\tilde{c}_{i'}^x = c_{i'}^x - \min\{c_{i'}^x, c_{i'+m}^x\}, \tilde{c}_{i'+m}^x = c_{i'+m}^x - \min\{c_{i'}^x, c_{i'+m}^x\}$, which makes one of them zero and keep all other coefficients unchanged. Note then that $\mathbf{x} = \sum_{i=1}^{2m} \tilde{c}_i^x \mathbf{a}_i, \tilde{\mathbf{c}}^x \geq 0$ and $\mathbb{1}^T\tilde{\mathbf{c}}^x < \mathbb{1}^T\mathbf{c}^x = \|\mathbf{x}\|_{\mathcal{A}}$ leading to a contradiction. The uniqueness follows from the linear independence of the atoms. The other direction follows from the uniqueness observation. □

6.6 Proof of Lemma 3

Lemma 3. *Given* $\mathbf{x}, \mathbf{y} \in \mathbb{R}^p$*, s.t* $\mathbf{c}^x = \mathrm{MR}(\mathbf{x}), \mathbf{c}^y = \mathrm{MR}(\mathbf{y})$*, we have* $\|\mathbf{x}-\mathbf{y}\|_{\mathcal{A}} = \|\mathbf{c}^x - \mathbf{c}^y\|_1$.

Proof. We can write $\mathbf{x}-\mathbf{y} = \sum_{i=1}^{2m} c_i^{x-y}\mathbf{a}_i$ where $\mathbf{c}^{x-y} = \mathrm{MR}(\mathbf{x}-\mathbf{y})$. By linear independence, we have $(c_i^x - c_{i+m}^x) - (c_i^y - c_{i+m}^y) = (c_i^{x-y} - c_{i+m}^{x-y})$. By lemma 2, we know that $\forall i$ either c_i^{x-y} or c_{i+m}^{x-y} is zero. It follows then that the other will be equal to $|(c_i^x - c_{i+m}^x) - (c_i^y - c_{i+m}^y)|$. Hence $\|(\mathbf{c}^x(1:m) - \mathbf{c}^y(1:m)) - (\mathbf{c}^x(m+1:2m) - \mathbf{c}^y(m+1:2m))\|_1 = \mathbb{1}^T\mathbf{c}^{x-y} = \|x-y\|_{\mathcal{A}}$.

By lemma 2, we only need to consider these cases:

c_i^x	c_{i+m}^x	c_i^y	c_{i+m}^y	$\lvert(c_i^x - c_{i+m}^x) - (c_i^y - c_{i+m}^y)\rvert$	$\lvert c_i^x - c_i^y\rvert + \lvert c_{i+m}^x - c_{i+m}^y\rvert$
>0	0	>0	0	$\lvert c_i^x - c_i^y\rvert$	$\lvert c_i^x - c_i^y\rvert$
>0	0	0	>0	$c_i^x + c_{i+m}^y$	$c_i^x + c_{i+m}^y$
0	>0	>0	0	$c_{i+m}^x + c_i^y$	$c_{i+m}^x + c_i^y$
0	>0	0	>0	$\lvert c_{i+m}^x - c_{i+m}^y\rvert$	$\lvert c_{i+m}^x - c_{i+m}^y\rvert$

Hence, $\|\mathbf{x}-\mathbf{y}\|_{\mathcal{A}} = \|(\mathbf{c}^x(1:m) - \mathbf{c}^y(1:m)) - (\mathbf{c}^x(m+1:2m) - \mathbf{c}^y(m+1:2m))\|_1 = \|\mathbf{c}^x(1:m) - \mathbf{c}^y(1:m)\|_1 - \|\mathbf{c}^x(m+1:2m) - \mathbf{c}^y(m+1:2m)\|_1 = \|\mathbf{c}^x - \mathbf{c}^y\|_1$. □

6.7 Proof of Proposition 2

Algorithm 4 presents the general version of Algorithm 1 which can handle the case where $\|\mathbf{u}\|_{\mathcal{A}} > \lambda$. In the case where $\|\mathbf{u}\|_{\mathcal{A}} \leq \lambda$ and t is given, Algorithm 4 reduces to algorithm 5.

Proposition 2. *Algorithm 1 returns* $\mathbf{x} \in \mathrm{prox}_g^{\mathcal{A}}(\mathbf{u}, \mathbf{z}, L)$ *in* $O(p\mathcal{T} + p\log p)$ *time, where* $\mathcal{T}$ *is the time to compute* $\mathbf{z}^T\mathbf{a}$ *for any atom* $\mathbf{a} \in \mathcal{A}$.

Proof. We know from lemma 1 that solving $\mathrm{prox}_g^{\mathcal{A}}(\mathbf{u}, \mathbf{z}, L)$ reduces to solving $\mathrm{lconj}_g^{\mathcal{A}}(\mathbf{u}, \mathbf{z}, t)$ with $t = t^\star$. We show first that given any $t \geq 0$ algorithm 5 indeed returns $x^k \in \mathrm{lconj}_g^{\mathcal{A}}(\mathbf{u}, \mathbf{z}, t)$. Making use of lemma 2 and 3, we make the following observations:

- $\delta_0 = \max_{\delta>0}\{\delta : \mathbf{u} + \delta\mathbf{a}_{j_{\min}} \in \lambda\mathrm{conv}(\mathcal{A}) \cap (t\mathrm{conv}(\mathcal{A}) + \mathbf{u})\}$.
  To see this note that for any $\delta > 0$ s.t. $\mathbf{x} = \mathbf{u} + \delta\mathbf{a}_{j_{\min}}$ is feasible, we need to have $\|\mathbf{x}-\mathbf{u}\|_{\mathcal{A}} = \delta \leq t$ and $\|\mathbf{x}\|_{\mathcal{A}} \leq \lambda$, i.e., $\sum_{i \neq i_{\min}, i_{\min}+m} c_i^u + |\delta + c_{i_{\min}}^u - c_{i_{\min}+m}^u| \leq \lambda$ (by lemma 2). Since $\mathbb{1}^T\mathbf{c}^u = \|\mathbf{u}\|_{\mathcal{A}} \leq c$, we deduce the following constraint (note that we don't need to consider cases where $\delta + c_{i_{\min}}^u - c_{i_{\min}+m}^u \leq 0$ since in that case $\|x\|_{\mathcal{A}} \leq \lambda$ is trivially satisfied for any $\delta \geq 0$), $\delta \leq \lambda - \|\mathbf{u}\|_{\mathcal{A}} + 2c_{i_{\min}+m}^u$. Hence, δ_0 is indeed the maximal feasible step in this direction.
- $\delta_0 = t$ then $\mathbf{x}^0$ is optimal.
  Given any $\mathbf{x} \in \mathbb{R}^p$ s.t., $\|\mathbf{x}-\mathbf{u}\|_{\mathcal{A}} \leq t$, i.e., $\mathbf{x}-\mathbf{u} \in t\mathrm{conv}(\mathcal{A})$, we can write it as $\mathbf{x}-\mathbf{u} = \sum_{i=1}^{2m} c_i^{x-u}\mathbf{a}_i$ with $\mathbf{c}^{x-u} \geq 0$ and $\mathbb{1}^T\mathbf{c}^{x-u} = t$ (not necessarily a minimal representation). If $t = \delta_0$ then $\mathbf{z}^T(\mathbf{x}-\mathbf{u}) = \sum_{i=1}^{2m} c_i^{x-u}\mathbf{z}^T\mathbf{a}_i \geq t\mathbf{z}^T\mathbf{a}_{i_{\min}} = \mathbf{z}^T(\mathbf{x}^0 - \mathbf{u})$, so $\mathbf{x}^0$ is optimal.
- If $\delta_0 \neq t$, we have $\|x^0\|_{\mathcal{A}} = \lambda$.
  We prove this by contradiction. Assume $\|\mathbf{x}^0\|_{\mathcal{A}} < \lambda$ and let $\delta = \min\{\lambda - \|\mathbf{x}^0\|_{\mathcal{A}}, t - \delta_0\} > 0$, and let $\mathbf{x}' = \mathbf{u} + (\delta + \delta_0)\mathbf{a}_{i_{\min}} \neq \mathbf{x}^0$. $\mathbf{x}'$ is feasible since $\|\mathbf{x}' - \mathbf{u}\|_{\mathcal{A}} = \|(\delta + \delta_0)\mathbf{a}_{i_{\min}}\|_{\mathcal{A}} = \delta + \delta_0 \leq t$ and $\|\mathbf{x}'\|_{\mathcal{A}} \leq \|\mathbf{x}^0\|_{\mathcal{A}} + \|\delta\mathbf{a}_{i_{\min}}\|_{\mathcal{A}} \leq \lambda$ (by triangle inequality). This contradicts the above observation about δ^0.

Algorithm 4 Prox of linearly independent atomic norms: $\text{prox}_g^{\mathcal{A}}(\mathbf{u}, \mathbf{z}, L)$

Input: $\mathbf{c}^u = \text{MR}(\mathbf{u})$.
Initialize: $\mathbf{x}^0 = \mathbf{u}, \mathbf{c}^x = \mathbf{c}^u, t_u^0 = 0, r = 1$.
$t_{\min} = \max\{\|\mathbf{u}\|_{\mathcal{A}} - \lambda, 0\}$
$\mathbf{a}_{i_{\min}} := \text{LMO}_{\mathcal{A}}(\mathbf{z})$
$t_l^0 = \max\{-\mathbf{z}^T\mathbf{a}_{i_{\min}}/L, t_{\min}\}$.
Sort $\mathbf{z}^T\mathbf{a}_i$ for active atoms: $\mathbf{z}^T\mathbf{a}_{j_1} \geq \mathbf{z}^T\mathbf{a}_{j_2} \geq \cdots, \forall c_j^u > 0$.
if $t_{\min} > 0$ **then**
  Let r be the smallest integer s.t. $\sum_{k=1}^{r} c_{j_k}^u \geq t_{\min}$.
  for $k = 1, \cdots, r-1$ **do**
    $\mathbf{x}^0 = \mathbf{x}^0 - c_{j_k}^u \mathbf{a}_{j_k}, c_{j_k}^x = 0$.
  end for
  $\mathbf{x}^0 = \mathbf{x}^0 - (t_{\min} - \sum_{i=1}^{k} c_{j_k}^u)\mathbf{a}_{j_k}$
  $c_{j_k}^x = c_{j_k}^u - (t_{\min} - \sum_{i=1}^{k} c_{j_k}^u)$.
  $\delta_0 = t_{\min}$.
else
  $\delta_0 = \min\{t_l^0, \lambda - \|\mathbf{u}\|_{\mathcal{A}} + 2c_{i_{\min}+m}^u\}$
  $\mathbf{x}^0 = \mathbf{u} + \delta_0 \mathbf{a}_{i_{\min}}$.
  $c_{i_{\min}}^x = \max\{\delta_0 + c_{i_{\min}}^u - c_{i_{\min}+m}^u, 0\}, c_{i_{\min}+m}^x = -\min\{\delta_0 + c_{i_{\min}}^u - c_{i_{\min}+m}^u, 0\}$
end if
$t_u^0 = \delta_0, t_l^r = t_l^0 - t_u^0$.
while $k = r, \cdots, p$ and $t_l^k \geq 0$ **do**
  $t_l^k = \max\{-0.5\mathbf{z}^T(\mathbf{a}_{i_{\min}} - \mathbf{a}_{j_k})/L - t_u^k, 0\}$.
  $\delta_k = \min\{c_{j_k}^x, t_l^k/2\}$
  $\mathbf{x}^k = \mathbf{x}^{k-1} + \delta_k(\mathbf{a}_{i_{\min}} - \mathbf{a}_{j_k})$
  $t_l^{k+1} = t_l^k - 2\delta_k$
end while
Return: $\mathbf{x}^k$

Algorithm 5 Local Conjugate of linearly independent atomic norms: $\text{lconj}_g^{\mathcal{A}}(\mathbf{u}, \mathbf{z}, t)$

Input: $\mathbf{c}^u = \text{MR}(\mathbf{u}), t \geq 0$
Initialize: $\mathbf{x}^0 = \mathbf{u}, \mathbf{c}^x = \mathbf{c}^u, t_l^0 = t$
$\mathbf{a}_{i_{\min}} \in \arg\min_{\mathbf{a} \in \mathcal{A}} \mathbf{z}^T\mathbf{a}$
Sort: $\mathbf{z}^T\mathbf{a}_{j_1} \geq \mathbf{z}^T\mathbf{a}_{j_2} \geq \cdots, \forall c_j^u > 0$.
$\delta_0 = \min\{t_l^0, \lambda - \|\mathbf{u}\|_{\mathcal{A}} + 2c_{i_{\min}+m}^u\}$
$\mathbf{x}^0 = \mathbf{u} + \delta_0 \mathbf{a}_{i_{\min}}$.
$c_{i_{\min}}^x = \max\{\delta_0 + c_{i_{\min}}^u - c_{i_{\min}+m}^u, 0\}, c_{j_{\min}+m}^x = -\min\{\delta_0 + c_{i_{\min}}^u - c_{i_{\min}+m}^u, 0\}$
$t_l^1 = t_l^0 - \delta_0$.
while $k = 1, \cdots, m$ and $t_l^k \geq 0$ **do**
  $\delta_k = \min\{c_{j_k}^x, t_l^k/2\}$
  $\mathbf{x}^k = \mathbf{x}^{k-1} + \delta_k(\mathbf{a}_{i_{\min}} - \mathbf{a}_{j_k})$
  $t_l^{k+1} = t_l^k - 2\delta_k$
end while
Return: $\mathbf{x}^k$

- If $\delta_0 \neq t$, then there exists an optimal solution $\mathbf{x}^\star$ s.t. $\|\mathbf{x}^\star\|_{\mathcal{A}} = \lambda$.
  To see this let $\delta = \min\{(\lambda - \|\mathbf{x}^\star\|_{\mathcal{A}})/2, c_j^{x^\star - u}\} > 0$, where $\mathbf{c}^{x^\star - u} = \mathrm{MR}(\mathbf{x}^\star - \mathbf{u})$, and j any index that satisfies $j \neq i_{\min}, c_j^{x^\star - u} > 0$. Such index exists unless $\mathbf{x}^\star = \mathbf{u} + c_{i_{\min}}^{x^\star - u}\mathbf{a}_{i_{\min}}$, in which case $\mathbf{x}^0$ is optimal. Let $\mathbf{x}' = \mathbf{x}^\star + \delta(\mathbf{a}_{i_{\min}} - \mathbf{a}_j) \neq \mathbf{x}^\star$, $\|\mathbf{x}' - \mathbf{u}\|_{\mathcal{A}} = \|(c_{i_{\min}}^{x^\star - u} + \delta - c_{i_{\min}+m}^{x^\star - u})\mathbf{a}_{i_{\min}} + (c_j^{x^\star - u} - \delta))\mathbf{a}_j + \sum_{i \neq j, i_{\min}, i_{\min}+m} c_i^{x^\star - u}\mathbf{a}_i\|_{\mathcal{A}} \leq \mathbb{1}^T \mathbf{c}^{x^\star - u} \leq t$, $\|\mathbf{x}'\|_{\mathcal{A}} \leq \|\mathbf{x}^\star\|_{\mathcal{A}} + \|\delta \mathbf{a}_{i_{\min}}\|_{\mathcal{A}} + \|\delta(-\mathbf{a}_{i_j})\|_{\mathcal{A}} = \|\mathbf{x}^\star\|_{\mathcal{A}} + 2\delta \leq \lambda$. So $\mathbf{x}'$ is feasible and has a better objective than $\mathbf{x}^\star$ leading to a contradiction.
- There exists an optimal solution s.t. $\|\mathbf{x}^\star - \mathbf{x}^0\|_{\mathcal{A}} \leq t - \delta_0$.
  By the above observation, this is trivial if $t = \delta_0$. It also holds trivially if $\delta_0 = 0$. Otherwise, it is enough to show that $c_{i_{\min}}^{x^\star - u} \geq \delta_0$, where $\mathbf{c}^{x^\star - u} = \mathrm{MR}(\mathbf{x}^\star - \mathbf{u})$. Since $\|\mathbf{x}^\star - \mathbf{x}^0\|_{\mathcal{A}} = \|(c_{i_{\min}}^{x^\star - u} - c_{i_{\min}+m}^{x^\star - u - \delta_0})\mathbf{a}_{i_{\min}} + \sum_{i \neq i_{\min}, i_{\min}+m} c_i^{x^\star - u}\mathbf{a}_i\|_{\mathcal{A}}$, if $c_{i_{\min}}^{x^\star - u} \geq \delta_0 > 0$, then by lemma 2, $c_{i_{\min}+m}^{x^\star - u} = 0$ and $\|\mathbf{x}^\star - \mathbf{x}^0\|_{\mathcal{A}} = \sum_{i \neq i_{\min}, i_{\min}+m} c_i^{x^\star - u} + (c_{i_{\min}}^{x^\star - u} - \delta_0) = \mathbb{1}^T \mathbf{c}^{x^\star - u} - \delta_0 \leq t - \delta_0$.
  To show that $c_{i_{\min}}^{x^\star - u} \geq \delta_0$, assume towards contradiction that $c_{i_{\min}}^{x^\star - u} < \delta_0$, and let j be an index where $c_j^{x^\star - u} > 0$ and $c_j^{x^\star - u} - c_{j+m}^u > 0$ and $j \neq i_{\min}$. Such index must exists, since otherwise $\forall i \neq i_{\min}$ where $c_i^{x^\star - u} > 0$, we'll have $0 < c_i^{x^\star - u} \leq c_{i+m}^u$, hence by lemma 2 $c_i^u = 0$. Then we can write $\mathbf{x}^\star = \mathbf{u} + \sum_i c_i^{x^\star - u}\mathbf{a}_i = \sum_{c_i^{x^\star - u} > 0, i \neq i_{\min}} (-c_{i+m}^u + c_i^{x^\star - u})\mathbf{a}_i + |c_{i_{\min}}^{x^\star - u} - c_{i_{\min}+m}^u|\mathbf{a}_{i_{\min}}$. We assume that $\exists i \neq i_{\min}, c_i^{x^\star - u} > 0$, otherwise $\mathbf{x}^\star = \mathbf{x}^0$. Hence, we'll have the following 2 cases:

$$
\begin{aligned}
\lambda = \|\mathbf{x}^\star\|_{\mathcal{A}} &= \begin{cases} \sum_{c_i^{x^\star - u} > 0} c_{i+m}^u - \sum_{c_i^{x^\star - u} > 0} c_i^{x^\star - u} & \text{if } c_{i_{\min}}^{x^\star - u} - c_{i_{\min}+m}^u \leq 0 \\ \sum_{c_i^{x^\star - u} > 0} c_{i+m}^u - \sum_{c_i^{x^\star - u} > 0, i \neq i_{\min}} c_i^{x^\star - u} + c_{i_{\min}}^{x^\star - u} - 2c_{i_{\min}+m}^u & \text{otherwise} \end{cases} \\
&< \begin{cases} \|\mathbf{u}\|_{\mathcal{A}} & \text{if } c_{i_{\min}}^{x^\star - u} - c_{i_{\min}+m}^u \leq 0 \\ \|\mathbf{u}\|_{\mathcal{A}} + \delta_0 - 2c_{i_{\min}+m}^u & \text{otherwise} \end{cases} \qquad (\text{since } c_{i_{\min}}^{x^\star - u} < \delta_0) \\
&\leq \lambda
\end{aligned}
$$

  which leads to a contradiction. Hence, such index must exists. Then let $\delta = c_j^{x^\star - u} - c_{j+m}^u > 0$ and $\mathbf{x}' = \mathbf{x}^\star + \delta(\mathbf{a}_{i_{\min}} - \mathbf{a}_j) \neq \mathbf{x}^\star$. We show that $\mathbf{x}'$ is feasible. First note that $-\mathbf{u} = \sum_i c_i^u(-\mathbf{a}_i) = \sum_i c_{i+m}^u \mathbf{a}_i$ and hence by lemma 3, $\|\mathbf{x}^\star\|_{\mathcal{A}} = \|\mathbf{x}^\star - \mathbf{u} - (-\mathbf{u})\|_{\mathcal{A}} = \|\mathbf{c}^{x^\star - u} - \tilde{\mathbf{c}}^u\|_1 = \lambda$ where $\tilde{\mathbf{c}}_i^u = \mathbf{c}_{i+m}^u$. Then, we have $\|\mathbf{x}' - \mathbf{u}\|_{\mathcal{A}} = \|(c_{i_{\min}}^{x^\star - u} + \delta - c_{i_{\min}+m}^{x^\star - u})\mathbf{a}_{i_{\min}} + (c_j^{x^\star - u} - \delta))\mathbf{a}_j + \sum_{i \neq j, i_{\min}, i_{\min}+m} c_i^{x^\star - u}\mathbf{a}_i\|_{\mathcal{A}} \leq \mathbb{1}^T \mathbf{c}^{x^\star - u} \leq t$ and $\|\mathbf{x}'\|_{\mathcal{A}} = \|\sum_{i \neq j, j+m}(c_i^{x^\star - u} + c_i^u)\mathbf{a}_i + (c_j^{x^\star - u} + c_j^u - c_{j+m}^u - \delta)\mathbf{a}_j + \delta \mathbf{a}_{i_{\min}}\|_{\mathcal{A}} \leq \|\sum_{i \neq j, j+m} c_i^{x^\star - u}\mathbf{a}_i + (c_j^{x^\star - u} - c_{j+m}^u - \delta)\mathbf{a}_j - (-\mathbf{u})\|_{\mathcal{A}} + \|\delta \mathbf{a}_{i_{\min}}\|_{\mathcal{A}} = \|\mathbf{c}^{x^\star - u} - \tilde{\mathbf{c}}^u\|_1 - \delta + \delta = \lambda$, by lemma 13. Finally note that $\mathbf{z}^T \mathbf{x}' \leq \mathbf{z}^T \mathbf{x}^\star$ leading to a contradiciton.

Note that in the algorithm 5 we enter the loop only if $t \neq \delta_0$. So if we stop before that then we have found an optimal solution $\mathbf{x}^0$. Otherwise, there exists an optimal solution s.t. $\|\mathbf{x}^\star\|_{\mathcal{A}} = \lambda$ and $\|\mathbf{x}^\star - \mathbf{x}^0\|_{\mathcal{A}} \leq t - \delta_0$, so we can now solve this problem instead:

$$
\min_{\substack{\|\mathbf{x}\|_{\mathcal{A}} = \lambda \\ \|\mathbf{x} - \mathbf{x}^0\|_{\mathcal{A}} \leq t}} \mathbf{z}^T \mathbf{x} \tag{22}
$$

We know though by lemma 3 that $\|\mathbf{x} - \mathbf{x}^0\|_{\mathcal{A}} = ||\mathbf{c}^x - \mathbf{c}^{x^0}||_1$, for $\mathbf{c}^x = \mathrm{MR}(\mathbf{x}), \mathbf{c}^{x^0} = \mathrm{MR}(\mathbf{x}^0)$. Hence we can further reformulate problem 22 as:

$$
\min_{\substack{\mathbb{1}^T \mathbf{c}^x = \lambda, \mathbf{c}^x \geq 0 \\ \|\mathbf{c}^x - \mathbf{c}^{x^0}\|_1 \leq t}} \tilde{\mathbf{z}}^T \mathbf{c}^x \tag{23}
$$

where $\tilde{\mathbf{z}}_i = \mathbf{z}^T \mathbf{a}_i$. This problem has been considered by [12], to obtain a local linear oracle. The rest of our algorithm, i.e., after entering the loop, reduces to their algorithm. So we refer the reader

to their proof of correctness [12, Lemma 5.2]. This concludes the proof that algorithm 5 returns $x^k \in \text{lconj}_g^{\mathcal{A}}(\mathbf{u}, \mathbf{z}, t)$.

Now we argue that algorithm 1 returns $x^k \in \text{prox}_g^{\mathcal{A}}(\mathbf{u}, \mathbf{z}, L)$. Recall from section 3.1 that $h(t)$ is a non-increasing piecewise linear function. But unlike the general case where we're computing $h(t)$ using a black box optimizer, we actually can compute the slopes of the different pieces of $h(t)$ explicitly. In fact, $h'(t^\star)$ belongs to one of these intervals: $[\mathbf{z}^T\mathbf{a}_{i_{\min}}, \infty], [0.5(\mathbf{z}^T\mathbf{a}_{i_{\min}} - \mathbf{z}^T\mathbf{a}_{j_2}), 0.5(\mathbf{z}^T\mathbf{a}_{i_{\min}} - \mathbf{z}^T\mathbf{a}_{j_1})], \cdots$. Note that algorithm 5 is actually minimizing the objective along the path of possible values of $t' = \|\mathbf{x} - \mathbf{u}\|_{\mathcal{A}}$ from $t' = t_{\min}$ to $t' = t$. In fact, $\mathbf{x}^k \in \text{lconj}_{\text{g}}(\mathbf{u}, \mathbf{z}, t_u^k), \forall k$ in algorithm 5. Hence, it's easy to incorporate the search for $t^\star$ without increasing the time complexity.
Finally, it is clear that the most expensive step in algorithm 1 is the sorting operation on line 5, and hence it's time complexity is $O(p\mathcal{T} + p \log p)$. Handling the case where $\|\mathbf{u}\|_{\mathcal{A}} > \lambda$ (c.f., lines 7 -14) follows using similar arguments. □

6.8 Proof of Theorem 2

First, we present a technical lemma, which can be found in [9, Example 2.9]. We provide a proof of it here for completeness. The term $\mathbf{p}^k$ satisfies the following property, which is usefull to handle general norms.

Lemma 4 (*cf.*, [9, Example 2.9])**.** *$\|\cdot\|^2$ is differentiable at zero with $\partial(\frac{1}{2}\|\cdot\|^2)(\mathbf{0}) = 0$ and $\forall \mathbf{x} \in \mathbb{R}^p$ and $\mathbf{p} \in \partial(\frac{1}{2}\|\cdot\|^2)(\mathbf{x})$, we have*

$$\langle \mathbf{x}, \mathbf{p} \rangle = \|\mathbf{x}\|^2 = \|\mathbf{p}\|_*^2. \tag{24}$$

Proof. Note that $\forall \mathbf{x} \in \mathbb{R}^p$

$$\lim_{t \to 0} \frac{\|\mathbf{0} + t\mathbf{z}\|^2 - \|\mathbf{0}\|^2}{t} = \lim_{t \to 0} t\|\mathbf{x}\|^2 = 0, \tag{25}$$

which implies that $\|\cdot\|^2$ is differentiable at $\mathbf{0}$. Hence if $\mathbf{x} = 0$ then $\mathbf{p} = 0$ and (24) trivially holds. Otherwise if $\mathbf{x} \neq 0$, note that since $\|\cdot\|^2$ is positively homogeneous of degree 2 and is locally Lipschitz, then by [34] Euler's identity holds

$$\|\mathbf{x}\|^2 = \langle \mathbf{x}, \mathbf{p} \rangle \leq \|\mathbf{x}\|\|\mathbf{p}\|_*, \tag{26}$$

which implies that $\|\mathbf{x}\| \leq \|\mathbf{p}\|_*$. The subdifferential of $\|\cdot\|$ exists every point (c.f., [37]) and $\mathbf{p}/\|\mathbf{x}\| \in \partial\|\mathbf{x}\|$. Then since $\|\cdot\| \in \Gamma_0$, it follows by Fenchel-Young equality,

$$\|\mathbf{p}/\|\mathbf{x}\|\|^* + \|\mathbf{x}\| = \langle \mathbf{x}, \mathbf{p}/\|\mathbf{x}\| \rangle = \|\mathbf{x}\|, \tag{27}$$

where $\|\cdot\|^*$ is the Fenchel conjugate of $\|\cdot\|$. This implies that $\|\mathbf{p}/\|\mathbf{x}\|\|^* = \iota_{\|\cdot\|_* \leq 1}(\mathbf{p}/\|\mathbf{x}\|) = 0$ and hence $\|\mathbf{p}\|_* \leq \|\mathbf{x}\|$, and thus (24) holds.

□

Theorem 2. *Consider Problem 1 where g is μ-strongly convex w.r.t. $\|\cdot\|$. If accGPM terminates at iteration k, i.e., $\mathbf{x}^{k+1} = \mathbf{y}^{k+1}$, then $\mathbf{x}^{k+1}$ is a solution to* (1)*. Otherwise, let $x^\star \in \mathscr{X}^\star$, the iterates of accGPM satisfy the following.*

1. *If $\mu = 0$. Then $\forall k \in \mathbb{N}$, we have $F(x^{k+1}) - F^\star \leq \dfrac{4\big(\sigma(F(x^0) - F^\star) + \beta_0 d(x^\star)\big)}{\sigma\{2 + \sqrt{\beta_0}\sum_{i=0}^k \sqrt{\gamma_i}\}^2}$.*

   Consequently, if $\forall k \in \mathbb{N}, \gamma_k = 1/L$, then $F(x^{k+1}) - F^\star \leq \dfrac{4L\big(\sigma(F(x^0) - F^\star) + \beta_0 d(x^\star)\big)}{\sigma\{2\sqrt{L} + \sqrt{\beta_0}(k+1)\}^2}$.

2. *If $\mu > 0$. Set $\tau = \inf_{k\in\mathbb{N}} \tau_k$, and* $\rho = \frac{\tau\mu}{2L}\left\{\sqrt{1+\frac{4L}{\beta_0+\mu}}-1\right\}$. *If $\beta_0 \geq \tau\mu$ and $\forall k \in \mathbb{N}, \gamma_k = 1/L$, then we have $F(x^{k+1}) - F^\star \leq (1-\rho)^{k+1}\{F(x^0) - F^\star + \frac{\beta_0}{\sigma} d(x^\star)\}$.*

Proof. If there exists $k \in \mathbb{N}$ such that $x^{k+1} = y^{k+1}$ then it follows from Step 6 of Algorithm 2 and Fermat's rule that

$$0 \in \partial g(\mathbf{x}^{k+1}) + \nabla f(\mathbf{x}^{k+1}) + \partial(\frac{1}{2\gamma_k}\|\cdot\|^2)(\mathbf{0}) \tag{28}$$

By lemma 4, (28) yields $0 \in \partial g(\mathbf{x}^{k+1}) + \nabla f(\mathbf{x}^{k+1})$ and thus $\mathbf{x}^k$ is a minimizer of F. We now suppose that $\forall k \in \mathbb{N}, x^{k+1} \neq y^{k+1}$. Step 10 of Algorithm 2 yields

$$(\forall k \in \mathbb{N}) \quad \mathbf{p}^k - \nabla f(\mathbf{y}^{k+1}) \in \partial g(\mathbf{x}^{k+1}) \tag{29}$$

It follows then that

$$g(\mathbf{x}) \geq g(\mathbf{x}^{k+1}) + \left\langle \mathbf{x} - \mathbf{x}^{k+1}, \mathbf{p}^k - \nabla f(\mathbf{y}^{k+1})\right\rangle. \tag{30}$$

Since ∇f is L-Lipschitz and since $\forall k \in \mathbb{N}, \gamma_k \in (0, 1/L]$, it follows from that

$$f(\mathbf{x}^{k+1}) \leq f(\mathbf{y}^{k+1}) + \left\langle \mathbf{x}^{k+1} - \mathbf{y}^{k+1}, \nabla f(\mathbf{y}^{k+1})\right\rangle + \frac{1}{2\gamma_k}\|\mathbf{x}^{k+1} - \mathbf{y}^{k+1}\|^2. \tag{31}$$

In turn the convexity of f implies that

$$\begin{aligned} f(\mathbf{x}) &\geq f(\mathbf{y}^{k+1}) + \left\langle \mathbf{x} - \mathbf{y}^{k+1}, \nabla f(\mathbf{y}^{k+1})\right\rangle \\ &\geq f(\mathbf{x}^{k+1}) + \left\langle \mathbf{y}^{k+1} - \mathbf{x}^{k+1}, \nabla f(\mathbf{y}^{k+1})\right\rangle - \frac{1}{2\gamma_k}\|\mathbf{x}^{k+1} - \mathbf{y}^{k+1}\|^2 + \left\langle \mathbf{x} - \mathbf{y}^{k+1}, \nabla f(y^{k+1})\right\rangle \\ &= f(\mathbf{x}^{k+1}) + \left\langle \mathbf{x} - \mathbf{x}^{k+1}, \nabla f(\mathbf{y}^{k+1})\right\rangle - \frac{1}{2\gamma_k}\|\mathbf{x}^{k+1} - \mathbf{y}^{k+1}\|^2. \end{aligned} \tag{32}$$

Adding (30) and (32) we get

$$F(\mathbf{x}) \geq F(\mathbf{x}^{k+1}) + \left\langle \mathbf{x} - \mathbf{x}^{k+1}, \mathbf{p}^k\right\rangle - \frac{1}{2\gamma_k}\|\mathbf{x}^{k+1} - \mathbf{y}^{k+1}\|^2. \tag{33}$$

Hence, for every $\mathbf{x} \in \mathbb{R}^p$,

$$\begin{aligned} e_{k+1}(\mathbf{x}) - F(\mathbf{x}) &= (1-\alpha_k)(e_k(\mathbf{x}) - F(\mathbf{x})) + \alpha_k\big((1-\tau_k)(\psi_k(\mathbf{x}) - F(\mathbf{x})) + \tau_k(\phi_k(\mathbf{x}) - F(\mathbf{x}))\big) \\ &\leq (1-\alpha_k)(e_k(\mathbf{x}) - F(\mathbf{x})). \end{aligned} \tag{34}$$

Since d is σ-strongly convex and g is μ-strongly convex, it follows by induction that e_k is β_k-strongly convex. Next, let us show that

$$(\forall k \in \mathbb{N}) \quad e_k(\mathbf{w}^k) \geq F(\mathbf{x}^k). \tag{35}$$

Note that $e_0(\mathbf{w}^0) \geq F(\mathbf{x}^0)$. Suppose that $e_k(\mathbf{w}^k) \geq F(\mathbf{x}^k)$ for some $k \in \mathbb{N}$. Then it follows from (33) that

$$e_k(\mathbf{w}^k) \geq F(\mathbf{x}^k) \geq \psi_k(\mathbf{x}^{k+1})$$

Hence, since e_k is β_k-strongly convex, we have

$$\begin{aligned} e_k(\mathbf{w}^{k+1}) &\geq e_k(\mathbf{w}^k) + \frac{\beta_k}{2}\|\mathbf{w}^{k+1} - \mathbf{w}^k\|^2 \\ &\geq \psi_k(\mathbf{x}^{k+1}) + \frac{\beta_k}{2}\|\mathbf{w}^{k+1} - \mathbf{w}^k\|^2. \end{aligned} \tag{36}$$

However, since $\mathbf{p}^k - \nabla f(\mathbf{y}^{k+1}) \in \partial g(\mathbf{x}^{k+1})$,

$$g(\mathbf{x}) + \left\langle \mathbf{x} - \mathbf{x}^{k+1}, \nabla f(\mathbf{y}^{k+1}) - \mathbf{p}^k \right\rangle \geq g(\mathbf{x}^{k+1}), \tag{37}$$

and hence we deduce from (31) that

$$\phi_k(\mathbf{x}) \geq \psi_k(\mathbf{x}). \tag{38}$$

In turn, we deduce from (36) that

$$\begin{aligned} e_{k+1}(\mathbf{w}^{k+1}) &\geq (1-\alpha_k) e_k(\mathbf{w}^{k+1}) + \alpha_k \psi_k(\mathbf{w}^{k+1}) \\ &= F(\mathbf{x}^{k+1}) + \frac{(1-\alpha_k)\beta_k}{2}\|\mathbf{w}^{k+1} - \mathbf{w}^k\|^2 + \left\langle \mathbf{y}^{k+1} - \mathbf{x}^{k+1}, \mathbf{p}^k \right\rangle \\ &\quad + \left\langle \alpha_k(\mathbf{w}^{k+1} - \mathbf{w}^k), \mathbf{p}^k \right\rangle - \frac{1}{2\gamma_k}\|\mathbf{x}^{k+1} - \mathbf{y}^{k+1}\|^2. \end{aligned} \tag{39}$$

It follows from definition of $\mathbf{p}^k$ and Lemma 4 that

$$\left\langle \mathbf{y}^{k+1} - \mathbf{x}^{k+1}, p^k \right\rangle = \frac{1}{\gamma_k}\|\mathbf{x}^{k+1} - \mathbf{y}^{k+1}\|^2 = \gamma_k \| - \mathbf{p}^k\|_*^2. \tag{40}$$

On the other hand, the Cauchy-Schwarz inequality yields

$$\alpha_k \left\langle \mathbf{w}^{k+1} - \mathbf{w}^k, \mathbf{p}^k \right\rangle \geq -\frac{(1-\alpha_k)\beta_k}{2}\|\mathbf{w}^{k+1} - \mathbf{w}^k\|^2 - \frac{\alpha_k^2}{2(1-\alpha_k)\beta_k}\|\mathbf{p}^k\|_*^2. \tag{41}$$

Consequently, we deduce from (39) and (40) that

$$\begin{aligned} e_{k+1}(\mathbf{w}^{k+1}) &\geq F(\mathbf{x}^{k+1}) + \frac{1}{2\gamma_k}\|\mathbf{x}^{k+1} - \mathbf{y}^{k+1}\|^2 - \frac{\alpha_k^2}{2(1-\alpha_k)\beta_k\gamma_k}\|\mathbf{x}^{k+1} - \mathbf{y}^{k+1}\|^2 \\ &\geq F(\mathbf{x}^{k+1}) + \frac{1}{2\gamma_k}\left\{1 - \frac{\alpha_k^2}{(1-\alpha_k)\beta_k\gamma_k}\right\}\|\mathbf{x}^{k+1} - \mathbf{y}^{k+1}\|^2 \\ &= F(\mathbf{x}^{k+1}) \end{aligned} \tag{42}$$

which proves (35). Finally, we derive from the definition of $\mathbf{w}_{k+1}$ and (34) that

$$(\forall k \in \mathbb{N}) \quad F(\mathbf{x}^{k+1}) - F^\star \leq e_{k+1}(\mathbf{w}^{k+1}) - F^\star \leq e_{k+1}(\mathbf{x}^\star) - F^\star \leq (1-\alpha_k)\big(e_k(\mathbf{x}^\star) - F^\star\big), \tag{43}$$

where $x^\star$ is a minimizer of F. Hence, by induction,

$$F(\mathbf{x}^{k+1}) - F^\star \leq \prod_{i=0}^{k}(1-\alpha_i)\big(e_0(\mathbf{x}^\star) - F^\star\big). \tag{44}$$

(1): Note that $\forall k \in \mathbb{N}$

$$\alpha_k^2 = (1-\alpha_k)\beta_k\gamma_k \quad \text{and} \quad \beta_{k+1} = (1-\alpha_k)\beta_k$$

Hence, it follows from Lemma 2.2 in [13] that

$$\prod_{i=0}^{k}(1-\alpha_i) \leq \frac{1}{\left(1 + \sqrt{\beta_0}/2\sum_{i=0}^{k}\sqrt{\gamma_i}\right)^2}. \tag{45}$$

Consequently, the assertion follows from (44).

(2): First we note that by induction,

$$(\forall k \in \mathbb{N}) \quad \tau\mu \leq \beta_k \leq \beta_0 + \mu. \tag{46}$$

Therefore,

$$\alpha_{k+1} = \frac{\beta_k}{2L}\left\{\sqrt{1+\frac{4L}{\beta_k}} - 1\right\} \geq \frac{\tau\mu}{2L}\left\{\sqrt{1+\frac{4L}{\beta_0+\mu}} - 1\right\} \tag{47}$$

and hence the assertion follows from (43). □

7 Solving $\text{prox}_{\ell_1}^{\ell_1}$ and $\mathbf{p}^k$ for Section 5.1

7.1 Computing ℓ_1-proximity operator

Computing the standard $\text{prox}_{\ell_1}^{\ell_2}$ of ℓ_1-norm can be computed efficiently in $O(p)$ using the so-called soft thresholding operator $\text{SoftThreshold}(\mathbf{z}, \lambda) = \text{sign}(\mathbf{z}) \circ \max\{|\mathbf{z}| - \lambda, 0\}$, [6].

As explained in Remark 1, $\text{prox}_{\ell_1}^{\ell_1}$ can be solved by computing the prox over the ℓ_∞ ball by the greedy algorithm 4 and applying the decomposition in 1. By proposition 2, $\text{prox}_{\ell_1}^{\ell_1}$ can then be computed in $O(p \log p)$ time. However, $\text{prox}_{\ell_1}^{\ell_1}$ is simple enough that we opt for a direct way to solve it using again an intuitive greedy algorithm, of the same "flavor" as Algorithm 1.

Algorithm 6 ℓ_1-prox of ℓ_1-norm

Input: $\mathbf{u} \in \mathbb{R}^p, L_1 > 0$
Initialization: $\mathbf{x}^0 = \mathbf{u}, t_u^0 = 0, \mathbf{s}^0 = \text{sign}(\mathbf{x}^0), k = 0$
$s_i^0 = -\text{sign}(\text{SoftThreshold}(z_i, \lambda)), \forall i$ s.t. $x_i^0 = 0$.
$\mathbf{w} = [\mathbf{s}^0 \circ (\mathbf{z} + \lambda \mathbf{s}^0), \mathbf{s}^0 \circ (\mathbf{z} - \lambda \mathbf{s}^0)]$
Sort: $|w_{i_1}| \geq |w_{i_2}| \geq \cdots |w_{i_{2p}}|$
while $k = 1, \cdots, p+1$ and $t_l^k \geq 0$ **do**
  if $w_{i_k} = s_{i_k}^k \circ (z_{i_k} + \lambda s_{i_k}^k)$ **then**
    $t_l^{k+1} \leftarrow \max\{|w_{i_k}^k|/L_1 - t_u^k, 0\}$
    if $\text{sign}(w_{i_k}^k) > 0$ **then**
      $x_{i_k}^{k+1} = s_{i_k}^k \max\{|x_{i_k}^k| - t_l^k, 0\}$
      $t_u^{k+1} = t_u^k - |x_{i_k}^{k+1}| + |x_{i_k}^k|$
      if $x_{i_k}^{k+1} = 0$ **then**
        $s_{i_k}^{k+1} = -\text{sign}(\text{SoftThreshold}(z_i, \lambda))$
      end if
    else
      $x_{i_k}^{k+1} = s_{i_k}^k (|x_{i_k}^k| + t_l^k)$
      $t_u^{k+1} = t_u^k + |x_{i_k}^{k+1}| - |x_{i_k}^k|$
    end if
  end if
end while
Return: $\mathbf{x}^{k+1}$

Proposition 3. *Algorithm 6 returns* $\mathbf{x} \in \text{prox}_{\ell_1}^{\ell_1}(\mathbf{u}, \mathbf{z}, L)$ *in* $O(p \log p)$ *time.*

The high level idea of Algorithm 10 is the following. By lemma 3, we know that computing $\text{prox}_{\ell_1}^{\ell_1}(\mathbf{u}, \mathbf{z}, L)$ is equivalent to computing $\text{lconj}_{\ell_1}^{\ell_1}(\mathbf{u}, \mathbf{z}, L, t^\star)$, where recall from Section 3.2, $h(t)$ is a non-increasing piecewise linear function, whose slopes can be computed explicitly. Hence, the

search for $t^\star$ is done in Algorithm 10 the same way as in Algorithm 1. Algorithm 10 then solves $\text{lconj}_{\ell_1}^{\ell_1}(\mathbf{u}, \mathbf{z}, L, t)$ along the path of possible t values. Note that given t and the signs of $\mathbf{x}$, the objective in $\text{lconj}_{\ell_1}^{\ell_1}(\mathbf{u}, \mathbf{z}, L, t)$ reduces to minimizing a linear function $\text{sign}(\mathbf{x}) \circ (\mathbf{z} + \lambda \, \text{sign}(\mathbf{x}))$ over the ℓ_1-ball $\|\mathbf{x} - \mathbf{u}\|_1 \leq t^\star$ and the signs constraint, whose solution is simple (c.f., lines 10 and 16). To guess the optimal signs, we start by the feasible ones, $\text{sign}(\mathbf{u})$, and modify them gradually along the greedy solution path. It's clear that the time complexity of Algorithm 10 is dominated by the sorting operation on line 5, leading to a worst case complexity of $O(p \log p)$. However, in practice, we notice that we rarely do more than one iteration. In fact, when algorithm 10 is executed within GPM and accGPM, it's not hard to see that doing more than one itertation requires $\|\nabla f(\mathbf{x}^k)\|_\infty \leq \lambda$, and since λ is usually small, this condition implies that we're already near convergence, which is exactly what we observe in our experiments (c.f., section 5.1). Hence, in our implementation we choose instead to compute the maximum value of $\mathbf{w}$ at each iteration instead of sorting, leading to an expected complexity of $O(p)$. This observation is interesting, since it implies that running FW with carefully chosen step-size, approximate runing a proximal gradient method.

Finally note that the updates generated by $\mathbf{x} \in \text{prox}_{\ell_1}^{\ell_1}(\mathbf{u}, \mathbf{z}, L)$ are always sparse, i.e., given an s-sparse vector $\mathbf{u}$, $\mathbf{x}$ is at most $s+1$-sparse. The proof of proposition 3 follows by similar arguments as in proposition 2.

7.2 Computing the momentum term $\mathbf{p}^k$

Recall that accGPM required the computation of a momentum term $\mathbf{p}^k$ at each iteration k (c.f., line 10). Below, we show that the computation of $\mathbf{p}^k$, in this setting, has a closed form solution.

Proposition 4. *Given $\mathbf{x} \in \text{prox}_{\ell_1}^{\ell_1}(\mathbf{u}, \mathbf{z}, L)$ generated by algorithm 6, to have $\mathbf{p} \in \partial\left(-\frac{L}{2}\|\mathbf{x} - \mathbf{u}\|_1^2\right) \cap (\mathbf{z} + \lambda \partial \|\mathbf{x}\|_1)$, we can choose*

$$p_i = \begin{cases} -L\|\mathbf{x} - \mathbf{u}\|_1 \, \text{sign}(\mathbf{x} - \mathbf{u}) & \text{if } 0 = x_i \neq u_i \\ (s_i)^2(z_i + \lambda s_i) & \text{otherwise} \end{cases}$$

where $s_i = \text{sign}(x_i)$ if $x_i \neq 0$ and $s_i = -\,\text{sign}(\text{SoftThreshold}(z_i, \lambda))$ otherwise.

Proof. We note first than, by the optimality of $\mathbf{x}$, whenever one of the two sets consists of a unique element, then choosing this element must be a feasible choice. If $\|\mathbf{x} - \mathbf{u}\|_1 = 0$, then from algorithm 6, we know that $\mathbf{s} = 0$ and hence the above choice will correspond to $\mathbf{p} = 0$, which is the unique choice here. If $0 = x_i \neq u_i$ or $x_i \neq 0$ the choice of p_i above is again unique. Otherwise, if $0 = x_i = u_i$, then the choice of p_i is a feasible one, since $(s_i)^2(z_i + \lambda s_i) \in [-L\|\mathbf{x} - \mathbf{u}\|_1, L\|\mathbf{x} - \mathbf{u}\|_1] \cap z_i + [-\lambda, \lambda]$. □

7.3 Statistical Benefits of ℓ_1-GPM

This section makes a simple but powerful observation about the statistical performance of the ℓ_1-GPM in high-dimensional learning problems of the form,

$$\min_{\mathbf{x} \in \mathbb{R}^p} f(\mathbf{x}) + \lambda \|\mathbf{x}\|_1, \tag{48}$$

For this purpose, we consider separable objective functions $f(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^n f_i(\mathbf{x})$ that arises naturally in the high-dimensional learning problems, where one uses ℓ_1-regularized empirical risk minimization (ERM) to promote sparse solutions.

In the high-dimensional setting where $n \ll p$, the Hessian of $f(\mathbf{x}) \coloneqq \frac{1}{n}\sum_{i=1}^n f_i(\mathbf{x})$ is typically singular, and hence, the strong convexity assumption cannot hold. For example, if each $f_i(\mathbf{x})$ corresponds to

the negative log-likelihood of a distribution in canonical generalized linear models, then the $\nabla^2 f(\mathbf{x})$ is a sum of n rank-1 matrices, which is rank deficient when $n < p$.

A standard way to overcome such difficulty is to consider a suitable restriction set $\mathcal{M}$, and assume strong convexity only on $\mathcal{M}$. This leads to the notion of restricted strong convexity (RSC) [19]. In this paper, we consider a generalized version of it:

Definition 2 (Restricted Strong Convexity in General Norms)**.** *Let $\mathcal{M} \subseteq \mathbb{R}^p$. The function $f(\mathbf{x})$ is said to satisfy the ℓ_q-RSC with parameter μ_q if it holds that $(\forall \mathbf{x}, \mathbf{y} \in \mathcal{M}), \langle \nabla^2 f(\mathbf{x})\mathbf{y}, \mathbf{y}\rangle \geq \mu_q \|\mathbf{y}\|_q^2$.*

The ℓ_q-RSC implies consistency results in statistics: *cf.,* [31] for a comprehensive account. However, to date, only the ℓ_2-RSC is investigated in computational perspective [1, 35]. The result is that proximal gradient methods converge linearly when ℓ_2-RSC holds. Since the ℓ_1-GPM operates on the ℓ_1-norm, the ℓ_1-RSC, also known as the compatibility condition, can be used for similar faster convergence results. Importantly, the compatibility condition is strictly weaker than the ℓ_2-RSC, and sometimes the difference can be drastic [30].

Remark 2. *When the strong convexity in Theorem 1 is replaced by the compatibility condition, and if all the iterates $\mathbf{x}_k$ lie in the restriction set $\mathcal{M}$, then the conclusion of Theorem 1 holds with μ_1 being the parameter of the compatibility condition.*

Hence, as compared to euclidean proximal gradient methods, the ℓ_1-GPM can retain linear rate for a wider class of learning problems. We remark that, in general, identifying the restriction set $\mathcal{M}$ can be very difficult, and one often has to modify the standard algorithm in order to show that all the iterates lie in the set $\mathcal{M}$; cf., [1]. The rigorous proof of this condition is beyond the scope of the current paper. However, numerical results in Section 5.1 support faster convergence for the ℓ_1-GPM methods in general.

8 Solving $\mathbf{p}^k$ for $\mathrm{prox}_{\mathcal{G}}^{\ell_\infty}$ in Section 5.2

Proposition 5. *Given $\mathbf{x} \in \mathrm{prox}_{\mathcal{G}}^{\ell_\infty}(\mathbf{u}, \mathbf{z}, L)$ generated by algorithm 3, to have $\mathbf{p} \in \partial\left(-\frac{L}{2}\|\mathbf{x}-\mathbf{u}\|_\infty^2\right) \cap (\mathbf{z} + \lambda\partial\|\mathbf{x}\|_{\mathcal{G}})$, where $\|\mathbf{x}\|_{\mathcal{G}}$ is the ℓ_∞-LGL norm, we need to solve the following linear feasibility program.*

$$
\begin{aligned}
\mathbf{p} \in \underset{\mathbf{p}\in\mathbb{R}^p}{\arg\min} \quad & 0 \\
\textit{subject to} \quad & \mathbf{p}^T\left(\frac{\mathbf{x}-\mathbf{u}}{-Lt}\right) = t \\
& \left(\operatorname{sign}(\mathbf{x}-\mathbf{u}) \circ \frac{\mathbf{p}}{-Lt}\right)^T \mathbb{1} \leq 1 \\
& \mathbf{x}^T(\mathbf{p}-\mathbf{z}) = \lambda\|\mathbf{x}\|_{\mathcal{G}} \\
& \mathbf{B}^T(\operatorname{sign}(\mathbf{x}) \circ (\mathbf{p}-\mathbf{z})) \leq \lambda \\
& \operatorname{sign}(\mathbf{x}-\mathbf{u}) = \operatorname{sign}(\mathbf{p}) \\
& \operatorname{sign}(\mathbf{x}) = \operatorname{sign}(\mathbf{p}-\mathbf{z})
\end{aligned}
$$

where $t = \|\mathbf{x}-\mathbf{u}\|_\infty$ and $\mathbf{B}$ is the matrix whose columns are the indicator vectors of the groups, i.e., $\mathbf{B}_i = \mathbb{1}_{G_i}$.

Proof. By definition of dual norms, the subdifferential of the $\partial\left(-\frac{L}{2}\|\mathbf{x}-\mathbf{u}\|_\infty^2\right) = \{-Lt\kappa : \kappa^T(\mathbf{x}-\mathbf{u}) = \|\mathbf{x}-\mathbf{u}\|_\infty, \|\kappa\|_1 \leq 1\}$. The dual of ℓ_∞-LGL norm is given by $\max_{i\in[1,\cdots,M]}\|\kappa_{\mathcal{G}_i}\|_1$, hence $\lambda\partial\|\mathbf{x}\|_{\mathcal{G}} = \{\kappa : \kappa^T\mathbf{x} = \lambda\|\mathbf{x}\|_{\mathcal{G}}, \|\kappa_{\mathcal{G}_i}\|_1 \leq \lambda, \forall i \in [1,\cdots,M]\}$. The proposition then follows directly. □