Learning With Subquadratic Regularization : A Primal-Dual Approach

Raman Sankaran[1,3*], **Francis Bach**[2] and **Chiranjib Bhattacharyya**[3]
[1]LinkedIn, Bengaluru
[2]INRIA - Ecole Normale Supérieure - PSL Research University, Paris
[3]Indian Institute of Science, Bengaluru
rsankaran@linkedin.com, francis.bach@inria.fr, chiru@iisc.ac.in

Abstract

Subquadratic norms have been studied recently in the context of structured sparsity, which has been shown to be more beneficial than conventional regularizers in applications such as image denoising, compressed sensing, banded covariance estimation, etc. While existing works have been successful in learning structured sparse models such as trees, graphs, their associated optimization procedures have been inefficient because of hard-to-evaluate proximal operators of the norms. In this paper, we study the computational aspects of learning with subquadratic norms in a general setup. Our main contributions are two proximal-operator based algorithms ADMM-η and CP-η, which generically apply to these learning problems with convex loss functions, and achieve a proven rate of convergence of $O(1/T)$ after T iterations. These algorithms are derived in a primal-dual framework, which have not been examined for subquadratic norms. We illustrate the efficiency of the algorithms developed in the context of tree-structured sparsity, where they comprehensively outperform relevant baselines.

1 Introduction

Structured sparse regularizers [Kyrillidis, 2016; Bach *et al.*, 2011; Jenatton *et al.*, 2011b; Micchelli *et al.*, 2013] have emerged as efficient and versatile tools to add prior knowledge to estimation problems arising in domains such as computer vision [Mairal *et al.*, 2014], bioinformatics [Obozinski *et al.*, 2011], neural imaging [Jenatton *et al.*, 2011a] among many others. They typically lead to convex optimization problems of the form

$$\min_{w\in\mathbb{R}^d} F(Xw) + \lambda\Omega(w), \tag{1}$$

where $X \in \mathbb{R}^{n\times d}$ is the data matrix, $w \mapsto F(Xw)$ is the convex data-fitting term, the norm Ω is the regularizer, and $\lambda > 0$ is a balancing hyper-parameter. We refer to $F : \mathbb{R}^n \to \mathbb{R}$ as the loss function. Popular loss functions include the square loss used in least-squares regression [Tibshirani, 1994], the hinge loss used in support vector machines (SVM) [Vapnik, 2000], or the logistic loss used in logistic regression [Lee *et al.*, 2006]. Many algorithms exist if the proximal operator for Ω (denoted as prox_Ω) is known and efficient to compute [Bach *et al.*, 2011; Parikh and Boyd, 2014]. Depending on the loss function F, we may use forward-backward splitting methods such as FISTA [Beck and Teboulle, 2009; Nesterov, 2007] when F is smooth; ADMM [Parikh and Boyd, 2014], when one knows the proximal operator of $F \circ X : w \mapsto F(Xw)$; or Chambolle-Pock algorithm [Chambolle and Pock, 2011], when only prox_F is known.

In this paper, we focus on norms Ω which are subquadratic [Bach *et al.*, 2011], expressed as follows:

$$\Omega(w) = \frac{1}{2}\left(\inf_{\eta\in\mathbb{R}^d_+} \sum_{j=1}^{d} \frac{w_j^2}{\eta_j} + \Gamma(\eta)\right), \tag{2}$$

where $\Gamma : \mathbb{R}^d_+ \mapsto \mathbb{R}$ is convex and positively homogeneous. Note that the norm (2) includes as special cases the popular examples such as the ℓ_1-norm [Tibshirani, 1994] and grouped $\ell_{1,p}$-norms for $p \in [1, 2]$ [Jenatton *et al.*, 2011b; Kloft *et al.*, 2011]. For many such norms of the form (2), neither Ω nor prox_Ω is easily evaluated, where prox_Γ may be computable with an existing efficient algorithm. Examples include various norms derived from tree and graph structured constraints [Micchelli *et al.*, 2013; Baldassarre *et al.*, 2012], box-structured constraints [McDonald *et al.*, 2016] and norms obtained as convex relaxations of combinatorial penalties [Obozinski and Bach, 2016; Sankaran *et al.*, 2017], among many others. Thus, to solve (1) with Ω of the form (2), we may not be able to easily extend the aforementioned techniques which rely on efficient computation of prox_Ω. Hence, we use the relation (2) and reformulate (1) into the following optimization problem in variables w and η, which is the main focus of this paper:

$$\min_{w\in\mathbb{R}^d, \eta\in\mathbb{R}^d_+} \underbrace{F(Xw) + \tfrac{\lambda}{2}\textstyle\sum_{j=1}^{d} \frac{w_j^2}{\eta_j} + \frac{\lambda}{2}\Gamma(\eta)}_{\Psi(w,\eta)}. \tag{3}$$

To solve (3), when $\text{prox}_{F\circ X}$ and prox_Γ are easy to compute, one can devise an alternating optimization routine (minimizing w.r.t w and η alternatively until convergence). But this has convergence issues in general because the objective function is not smooth around vectors with zero elements, and

*The author is currently affiliated with LinkedIn, while the work was carried out when the affiliation was Indian Institute of Science.

does not offer a convergence rate. Alternatively, under the same assumptions, we can pose (3) as a smooth problem in η alone, which can be solved using FISTA, but suffers from expensive per-step computation. The only generic (i.e., with no strong assumptions on F) first-order algorithm is subgradient descent, which is very slow with a rate of $O(1/\sqrt{T})$ after T iterations. Thus, we are in need of efficient algorithms to solve problem (3). We make the following contributions:

- We investigate primal-dual algorithms for the problem (3). Though it is complicated to derive saddle point problems directly from (3), we propose a conic reformulation of the problem (3) in Section 5.1, which opens the possibilities for applying efficient primal-dual procedures. Also, the cone constraints derived are separable over the set of variables involved, and have closed form solutions for projection, which is used by the primal-dual algorithms.
- When $\text{prox}_{F\circ X}$ and prox_{Γ} are easy to compute, we propose in Section 5.2, a primal-dual algorithm ADMM-η to solve problem (3) which converges at the rate of $O(1/T)$ in T iterations (see Theorem 1). To achieve an ϵ-approximate solution with respect to the duality gap, ADMM-η takes $O\left((d + c_{F\circ X} + c_{\Gamma})/\epsilon\right)$, where $c_{F\circ X}$ and c_{Γ} denote the complexity to evaluate the proximal operator for $F \circ X$ and Γ respectively. In general, $c_{F\circ X}$ may be high because of the dependence on X. However for popular loss functions such as the squared loss and hinge loss, this computation turns out to be simple (see Table 1).
- When only prox_{F} and prox_{Γ} are easy to compute, we propose a generic first-order primal-dual algorithm CP-η in Section 5.3, which also converges at a rate of $O(1/T)$ (See Theorem 2). To guarantee an ϵ-approximate solution, CP-η takes $O\left((nd + c_F + c_{\Gamma})/\epsilon\right)$ operations, where c_F denotes the complexity to evaluate the proximal operator for F. In situations where $c_{F\circ X}$ is high, CP-η serves as a effective alternative to ADMM-η, while also being the most generic and efficient solution for (3).
- In Section 6, we consider an example of subquadratic norm (See Example 1) which encourages tree sparsity. As studied in [Obozinski and Bach, 2016], prox_{Γ} can be evaluated in $O(d \log d)$ time, whereas prox_{Ω} requires $O(d^2)$ computations. We illustrate that ADMM-η and CP-η outperform existing benchmarks on this chosen example.

Notations. 1_d (resp. 0_d) denotes a vector in $\mathbb{R}^d$ with all 1's (resp. 0's). Given $z \in \mathbb{R}^d$, we denote by $D(z)$ the diagonal matrix formed with z_i in the $(i,i)^{th}$ entry, and define $\mathrm{I}_d = D(1_d)$. Given a set A, define $I_A(a) = 0$ if $a \in A$, and $+\infty$ otherwise. Given $f : \mathbb{R}^n \to \mathbb{R}$, the Fenchel dual of f defined as $f^*(\alpha) = \sup_{x\in\mathbb{R}^n} x^\top \alpha - f(x)$. The proximal operator of a function f at z defined as $\text{prox}_f(z) = \text{argmin}_x \frac{1}{2}\|x - z\|^2 + f(x)$, and we define $\text{prox}_f^\tau(z) = \text{prox}_{\tau f}(z)$. We define $\text{prox}_{F\circ X}(z) = \text{argmin}_w \frac{1}{2}\|w - z\|^2 + F(Xw)$. Following [Nesterov, 2013], we define a function f as L-smooth if $\forall x, y \in \mathbb{R}^n$, $f(y) \le f(x) + \nabla f(x)^\top (y - x) + \frac{L}{2}\|y - x\|_2^2$. We define f as μ-strongly convex if $\forall x, y \in \mathbb{R}^n$, $f(y) \ge f(x) + \nabla f(x)^\top (y - x) + \frac{\mu}{2}\|y - x\|_2^2$.

2 Subquadratic Norms

Many existing studies have illustrated benefit of using subquadratic norms for structured sparsity in many applications: compressed sensing and image denoising [Baldassarre *et al.*, 2012], multi task learning [McDonald *et al.*, 2016], banded covariance matrix estimation [Yan and Bien, 2015]. Next, we discuss specific examples where prox_{Γ} is easy to compute.

Example 1. ***Tree-Structured $\mathcal{H}$-norms*** *[Baldassarre* et al.*, 2012; Micchelli* et al.*, 2013]. Consider a rooted tree with d nodes represented by the edge matrix $A \in \mathbb{R}^{d\times d}$, where $A_{ei} = 1, A_{ej} = -1$ if e^{th} edge links i to j. Define $\mathcal{H} = \{\eta \in \mathbb{R}_+^d | A\eta \in \mathbb{R}_+^d\}$, and let $\Gamma(\eta) = \eta^\top 1_d + 1_{\mathcal{H}}(\eta)$. With this choice of Γ, let us denote the norm* (2) *as $\Omega_{\mathcal{H}}$. By enforcing the tree structured prior on η, $\Omega_{\mathcal{H}}$ thus encourages w to be tree structured. The proximal operator for $\Omega_{\mathcal{H}}$ may be computed approximately using Picard iterates [Baldassarre* et al.*, 2012], which also lacks convergence rates. Whereas, we can compute prox_{Γ} easily since the projection onto $\mathcal{H}$ is computed easily through a pool-adjacent-violators (PAV) algorithm [Pardalos and Xue, 1999; Best and Chakravarti, 1990] in $O(d \log d)$ time.*

Example 2. ***Convex Relaxation of Combinatorial Penalties*** *[Bach, 2010; Bach, 2011b; Obozinski and Bach, 2016]. Denote $V = \{1, \ldots, d\}$. Given a tree structured ordering over V, define $\mathcal{G}_j = \{j \cup D_j\}$, where D_j denotes the descendants of j. We may arrive at a subquadratic regualarizer which encourages tree structures through the steps discussed next. Consider a set function $S(A) = \sum_{\mathcal{G}_j\in\mathcal{G}} 1_{\mathcal{G}_j \cap A \neq \emptyset}$. S is submodular [Bach, 2011a], and its corresponding Lovász extension denoted by $\Gamma(\eta)$ equals $\sum_j \|\eta_{\mathcal{G}_j}\|_\infty$, where $\eta_{\mathcal{G}_j}$ is the restriction of η to the coordinates given by $\mathcal{G}_j$. With this choice of Γ, we denote the norm* (2) *as $\Omega_2^S(w)$. prox_{Ω} is more expensive in this case: can be computed using a divide-and-conquer strategy involving a sequence of submodular function minimization (SFM), whose complexity is d(O(SFM)). Whereas, prox_{Γ} can be computed in time $O(dh)$, where h is the depth of the tree[Jenatton* et al.*, 2011b].*

Comparing Examples 1 and 2, note that $\Omega_2^S(w) = \Omega_{\mathcal{H}}(w)$ [Obozinski and Bach, 2016], illustrating that Γ need not be unique for a given subquadratic norm Ω. While these existing works on subquadratic norms illustrate the benefits in terms of applicability, they do not focus on computational efficiency, which is the focus of this paper.

3 Existing Algorithms

FISTA-η. Defining a smooth function $J(\eta)$ as follows:

$$J(\eta) = \inf_{w\in\mathbb{R}^d} F(Xw) + \frac{\lambda}{2}\sum_{j=1}^{d} w_j^2/\eta_j, \tag{4}$$

problem (3) is equivalent to $\min_{\eta\ge 0} J(\eta) + \frac{\lambda}{2}\Gamma(\eta)$ on which we can use (accelerated) proximal gradient descent [Beck and Teboulle, 2009] with J as the smooth component, and

$\Gamma(\eta)$ as the non-smooth component, whose proximal operator will be needed. The key is the computation of the gradient of J, for which we solve (4). With the proper change of variable, the minimizer of (4) is equivalent to $w = D(\eta)^{1/2}\text{prox}^{1/\lambda}_{F\circ XD(\eta)^{1/2}}(0)$. When the Lipshitz constant of the gradient of J is not known, backtracking is used to find the stepsize [Scheinberg *et al.*, 2014], which increases the number of times the gradient of J is calculated. We denote this algorithm as FISTA-η. Note that the per-iteration cost may overweigh the benefit of the fast convergence rate $O(1/T^2)$ of FISTA-η.

Alt-η. A simple algorithm is to alternatively solve for w and η at each iteration. Given η, to solve for w, we get an ℓ_2-regularized problem on w, whose solution is given as $w = D(\eta)^{1/2}\text{prox}^{1/\lambda}_{F\circ XD(\eta)^{1/2}}(0)$. Given w, solving for η is equivalent to evaluating the norm $\Omega(w)$ and returning the minimizer η. We refer to this algorithm as Alt-η. Though lacking in convergence guarantees and having issues with convergence when η has components close to 0, this algorithm is used in practice with good performances. Initializing η far from 0 is recommended or making sure that η stays away from 0 by adding a penalty $\varepsilon\sum_{j=1}^{d}\eta_j^{-1}$ [Canu and Grandvalet, 1999; Daubechies *et al.*, 2010; Argyriou *et al.*, 2008].

4 Primal-Dual Algorithms

Before discussing our proposed algorithms for solving (3), we briefly discuss the generic primal-dual setup we work on in this paper. Following [Chambolle and Pock, 2011], for finite-dimensional vector spaces $\mathcal{U}$ and $\mathcal{V}$, we consider a generic primal problem of the form

$$\min_{u\in\mathcal{U}}\left(\psi(u) \coloneqq H(Ku) + G(u)\right), \tag{5}$$

where $K : \mathcal{U} \to \mathcal{V}$ is a linear operator from $\mathcal{U}$ to $\mathcal{V}$ with the operator norm $\|K\| = \max\{\|Ku\|,\ u \in \mathcal{U}, \|u\| \leq 1\}$, $H : \mathcal{V} \to \mathbb{R}\cup\{+\infty\}$ and $G : \mathcal{U} \to \mathbb{R}\cup\{+\infty\}$ are proper convex functions, whose proximal operators are easily computable. This leads to the following saddle-point problem.

$$\min_{u\in\mathcal{U}}\max_{v\in\mathcal{V}}\left(\Upsilon(u,v) \coloneqq v^\top Ku + G(u) - H^*(v)\right). \tag{6}$$

Additionally when H is $(1/\gamma)$-smooth, H^* is γ-strongly convex. The primal-dual algorithm (CP) [Chambolle and Pock, 2011] for (6) by is given in Algorithm 1, which guarantees that there exists positive R_1, R_2 such that after T steps the following bounds hold true:

$$\min_{i=1}^{T}\psi(u^{(i)}) - \psi(u^*) \leq (R_1/T), \tag{7}$$

$$\psi(u^{(k)}) - \psi(u^*) \leq \left(R_2/\left(\gamma^2T^2\right)\right), \tag{8}$$

the second inequality being valid when F^* is γ-strongly-convex. The key features of Algorithm 1 are as follows:

1. It requires as input prox_G and prox_{H^*}, initial primal and dual step-sizes $\tau^{(0)}, \sigma^{(0)}$ satisfying $\tau^{(0)}\sigma^{(0)}\|K\|^2 \leqslant 1$.
2. It accesses K through only matrix-vector products, and hence strictly *first-order* in nature.

Algorithm 1 CP [Chambolle and Pock, 2011]

Require: $K, \text{prox}_G, \text{prox}_{H^*}, u^{(0)}, v^{(0)}, T$
Choose $\tau^{(0)}, \sigma^{(0)} > 0$ such that $\tau^{(0)}\sigma^{(0)}\|K\|^2 \leqslant 1$.
Initialize $\bar{u}^{(0)} = u^{(0)}$
for $k = 0, 1, 2, ..., T-1$ **do**
  $v^{(k+1)} = \text{prox}_{\sigma^{(k)}H^*}(v^{(k)} + \sigma K\tilde{u}^{(k)})$
  $u^{(k+1)} = \text{prox}_{\tau^{(k)}G}(u^{(k)} - \tau K^\top v^{(k+1)})$
  $\theta^{(k)} = \frac{1}{1+2\gamma\tau^{(k)}}$, $\tau^{(k)} = \theta^{(k)}\tau^{(k)}$, $\sigma_k = \frac{\tau^{(k)}}{\theta^{(k)}}$
  $\tilde{u}^{(k+1)} = u^{(k+1)} + \theta^{(k)}(u^{(k+1)} - u^{(k)})$
end for
Return $(u^{(1:T)}, v^{(1:T)})$.

3. The cost per iteration is c_u + c_v + cost(prox_G) + cost(prox^*_H), where c_u and c_v are the cost to compute Ku and $K^\top v$ respectively for vectors $u \in \mathcal{U}, v \in \mathcal{V}$.
4. coincides with ADMM, when $K = I$.

4.1 Step-Size Selection

Given $(u, v) \in \mathcal{U}\times\mathcal{V}$, referring to Theorem 1 of [Chambolle and Pock, 2011], when $\gamma = 0$, the following bound holds for all $\sigma^{(0)} = \sigma, \tau^{(0)} = \tau$ satisfying $\sigma\tau\ \|K\|^2 \leq 1$.

$$\Upsilon\left(\bar{u}^{(T)}, v\right) - \Upsilon\left(u, \bar{v}^{(T)}\right) \leq \left(\hat{R}_{\sigma,\tau}(u,v)/T\right), \text{with} \tag{9}$$

$$\hat{R}_{\sigma,\tau}(u,v) = \left(\frac{\|u-u^{(0)}\|_2^2}{2\tau} + \frac{\|v-v^{(0)}\|_2^2}{2\sigma}\right), \tag{10}$$

$$\bar{u}^{(T)} = \frac{1}{T}\sum_{t=1}^{T}u^{(t)}, \bar{v}^{(T)} = \frac{1}{T}\sum_{t=1}^{T}v^{(t)}, \tag{11}$$

Let (u^*, v^*) be an optimal solution of (6). Using the constraint $\sigma\tau\|K\|^2 \leq 1$, we optimize the upper bound in (9) for the choice $(u, v) = (u^*, v^*)$ resulting in the value $\sigma = \frac{1}{\|K\|}\frac{\|v^*-v^{(0)}\|_2}{\|u^*-u^{(0)}\|_2}$. Since it is not possible to calculate σ in practice because we do not have access to u^* or v^*, we will compute rough estimates using the information we have. In the experiments, we derive the step-sizes for the squared loss $F(z) = \frac{1}{2n}\|z-y\|_2^2$ and the ℓ_1 norm $\Gamma(\eta) = \eta^\top 1_d$. Note that the obtained step-sizes work well enough in our experiments, but that by an additional tuning, they could be improved.

5 Primal-Dual Formulation for Subquadratic Norms

We first derive a conic reformulation of (3), which gets rid of the ratio term $\frac{w_j^2}{2\eta_j}$, enabling application of Algorithm 1.

5.1 Conic-Constrained Primal Reformulation

Note that the term $\frac{w_j^2}{2\eta_j}$ may be written as $\frac{w_j^2}{2\eta_j} = \min_{t_j} t_j$ such that $t_j \geqslant 0, w_j^2 \leqslant 2t_j\eta_j$. Now, Eq. (3) is equivalent to:

$$\min_{w,\eta,t\in\mathbb{R}^d} F(Xw) + \tfrac{\lambda}{2}\Gamma(\eta) + \lambda 1_d^\top t + \textstyle\sum_{j=1}^{d} I_{\mathcal{C}}(w_j, \eta_j, t_j), \tag{12}$$

where $\mathcal{C} = \{(a,b,c) \in \mathbb{R}^3,\ a^2 \leqslant 2bc, b \geqslant 0, c \geqslant 0\}$ is the rotated second order-cone. Note that the cone constraint is

separable across the sets of variables (w_j, η_j, t_j) and thus can be handled separately. The cone $\mathcal{C}$ is self-dual and the proximal operator for $\sum_{j=1}^{d} I_{\mathcal{C}}(w_j, \eta_j, t_j)$, which will be the key to designing the primal-dual algorithms, involves computing the orthogonal projection onto $\mathcal{C}$ using a simple closed form expression. Thus, (12) makes it easy for deriving saddle-point problems of the form (6), which can be efficiently solved using Algorithm (1). In the next subsections, we consider reformulations of (12) under cases – (a) $\text{prox}_{F\circ X}$ is easy to compute, (b) only prox_F is computable easily.

5.2 ADMM-η: when $\text{prox}_{F\circ X}$ is Computable

When $\text{prox}_{F\circ X}$ is computable easily, we propose a primal-dual algorithm ADMM-η to solve (12). Algorithm 2 lists the steps, which accepts prox_{Γ} and $\text{prox}_{F\circ X}$ inputs. The following result proves that Algorithm 2 results in a rate of $O(1/T)$ for the problem (3).

Algorithm 2 ADMM-η

Require: $X, \text{prox}_{\Gamma}, \text{prox}_{F\circ X}, k$
1: Initialize $u^{(0)}, v^{(0)} \in \mathbb{R}^{3d}$, $K = -\text{I}_{3d}$. (Ref. (15)).
2: Define functions G, H^* as in (16).
3: $(u^{(k)}, v^{(k)}) = \text{CP}(K, \text{prox}_G, \text{prox}_{H^*}, u^{(0)}, v^{(0)}, k)$
4: $w^{(k)} = u^{(k)}(1:d), \eta^{(k)} = u^{(k)}(d+1:2d)$.
5: **Return** $w^{(k)}, \eta^{(k)}$.

Theorem 1. *Let $F : \mathbb{R}^n \to \mathbb{R}$ and $\Gamma : \mathbb{R}^n_+ \to \mathbb{R}$ be convex, and let $\Psi(w, \eta)$ be as defined in* (3) *with $(w^*, \eta^*) = \text{argmin}_{w,\eta} \Psi(w, \eta)$, and $\left(u^{(k)} = \left[w^{(k)}\ \eta^{(k)}\ t^{(k)}\right]^\top, v^{(k)} = \left[\delta^{(k)}\beta^{(k)}\gamma^{(k)}\right]^\top\right)_{k\geq 0}$ be defined as in Algorithm 2. Then, there exists a constant $R > 0$ such that after T iterations*

$$\min_{i=1}^{T} \Psi(w^{(i)}, \eta^{(i)}) - \Psi(w^*, \eta^*) \leq R/T \tag{13}$$

Proof. (Sketch) First, we can rewrite the cone constraint $I_{\mathcal{C}}$ in (12) in terms of its Fenchel dual, which again is a similar cone constraint on the dual variables, arriving at the following saddle-point problem equivalent to (12):

$$\min_{w,\eta,t\in\mathbb{R}^d} \max_{\delta,\beta,\nu\in\mathbb{R}^d} \left(F(Xw) + \frac{\lambda}{2}\Gamma(\eta) + \lambda 1_d^\top t\right) - \sum_{j=1}^{d} I_{\mathcal{C}}(\delta_j, \beta_j, \nu_j) + \sum_{j=1}^{d} \langle(\delta_j, \beta_j, \nu_j), (w_j, \eta_j, t_j)\rangle . \tag{14}$$

We can equate the above problem to (6) through the mapping of the variables u, v and functions G, H^* as given below.

$$u = \left[w^\top \eta^\top t^\top\right]^\top, u = \left[\delta^\top \beta^\top \nu^\top\right]^\top, K = -I_{3d}, \tag{15}$$

$$G(u) = F(Xw) + \frac{\lambda}{2}\Gamma(\eta) + \lambda 1_d^\top t,$$

$$H^*(v) = -\sum_{j=1}^{d} I_{\mathcal{C}}(\delta_j, \beta_j, \nu_j). \tag{16}$$

Since the function G is separable in terms of the primal variables w, η, t, prox_G simply reduces to computing the proximal operators independently on each group of variables. prox_G and prox^*_H may be computed straightforward using $\text{prox}_{F\circ X}$ and prox_{Γ}. To evaluate prox_G, the cost is usually dominated by the complexity to solve $\text{prox}_{F\circ X}$. Now result (7) applies, and since at optimality problem (12) and (3) have the same objective, we get the result. □

Discussion. We can make the following remarks:

1. The per-step cost of ADMM-η is dominated by $\text{prox}_{F\circ X}$. For the square loss $F(z) = \frac{1}{2n}\|z - y\|_2^2$, it requires solving a linear system which is constant for all iterations and hence can be solved efficiently in $O(d^2)$ time, after a single $O(d^3)$ operation at the start of the algorithm. Whereas for the hinge loss $F(z) = \frac{1}{n}\sum_{i=1}^{n} \max(0, 1 - z_i y_i)$, [W. Kienzle, 2006] proposed an "SMO-like" algorithm with $O(dn^2)$ operations.
2. The assumption make by ADMM-η (regarding $\text{prox}_{F\circ X}$) can be compared with those for Alt-η and FISTA-η, both of which requiring $\text{prox}_{F\circ \hat{X}}$, with $\hat{X} = XD^{\frac{1}{2}}(\eta)$. Since the matrix $\hat{X}$ changes at every iteration, computing $\text{prox}_{F\circ \hat{X}}$ is more difficult than $\text{prox}_{F\circ X}$. For instance, for the square loss, one needs to solve a different linear system at each iteration, which amounts to $O(d^3)$.
3. The step-size selection scheme we described in Section 5.3 leads to the following choices $\sigma = \|X\|^2/(\sqrt{3}n)$, $\tau = \sqrt{3}n/\|X\|^2$.

The next section discusses a generic first-order algorithm for (12) when $\text{prox}_{F\circ X}$ is difficult to compute.

5.3 CP-η: A generic first-order algorithm For (3)

While $\text{prox}_{F\circ X}$ may be difficult to obtain, prox_F may be easier to compute (this is the case for all common loss functions). We derive a first-order procedure denoted CP-η (Algorithm 3) having a convergence guarantee of $O(1/T)$ for all losses and norms, where prox_F and prox_{Γ} are easy to compute.

Algorithm 3 CP-η

Require: $X, \text{prox}_{\Gamma}, \text{prox}_{F^*}, k$
1: Initialize $u^{(0)} \in \mathbb{R}^{3d}, v^{(0)} \in \mathbb{R}^{n+3d}$ (Ref. (20)).
2: Initialize $r = \|X\|$.
3: Define functions G, H^* as in (21).
4: $K = \begin{bmatrix} X\ 0_{n\times 2d} \\ -r\text{I}_{3d} \end{bmatrix}$
5: $(u^{(k)}, v^{(k)}) = \text{CP}(K, \text{prox}_G, \text{prox}_{H^*}, u^{(0)}, v^{(0)}, k)$
6: $w^{(k)} = u^{(k)}(1:d), \eta^{(k)} = u^{(k)}(d+1:2d)$.
7: **Return** $w^{(k)}, \eta^{(k)}$

Theorem 2. *Let $\Psi(w, \eta)$ be as defined in* (3) *with $(w^*, \eta^*) = \text{argmin}_{w,\eta} \Psi(w, \eta)$, and $\left(u^{(k)} = \left[w^{(k)}\ \eta^{(k)}\ t^{(k)}\right]^\top, v^{(k)} = \left[\alpha^{(k)}\ \delta^{(k)}\ \beta^{(k)}\ \nu^{(k)}\right]^\top\right)$, K be defined as in Algorithm 3. Then $\forall\lambda > 0$, there exists a constant $R > 0$ such that after T iterations*

$$\min_{i=1}^{k} \Psi(w^{(i)}, \eta^{(i)}) - \Psi(w^*, \eta^*) \leq R/T. \tag{17}$$

Proof. (Sketch) Similar to arriving at (6) from (5) through the Fenchel dual of the loss function F, we apply the same trick in (12) to get the following equivalent problem.

$$\min_{w,\eta,t\in\mathbb{R}^d} \max_{\alpha\in\mathbb{R}^n} \alpha^\top Xw - F^*(\alpha) + \tfrac{\lambda}{2}\Gamma(\eta) + \lambda 1_d^\top t + \sum_{j=1}^{d} I_{\mathcal{C}}(w_j, \eta_j, t_j). \quad (18)$$

For the problem (18), we now use Fenchel duality on the constraint $I_{\mathcal{C}}$ and arrive at the following problem.

$$\min_{w,\eta,t\in\mathbb{R}^d} \max_{\alpha\in\mathbb{R}^n,\delta,\beta,\nu\in\mathbb{R}^d} \alpha^\top Xw - F^*(\alpha) + \frac{\lambda}{2}\Gamma(\eta) + \lambda 1_d^\top t + r\sum_{j=1}^{d} \left(\langle(\delta_j,\beta_j,\nu_j),(w_j,\eta_j,t_j)\rangle - I_{\mathcal{C}}(\delta_j,\beta_j,\nu_j)\right). \quad (19)$$

Now, the mapping of variables in this formulation to Eq. (6) is given by

$$u = \left[w^\top, \eta^\top, t^\top\right]^\top, v = \left[\alpha^\top, \delta^\top, \beta^\top, \nu^\top\right]^\top,$$
$$K = \begin{bmatrix} X\ 0_{n\times 2d} \\ -r\mathrm{I}_{3d} \end{bmatrix}, G(u) = \frac{\lambda}{2}\Gamma(\eta) + \lambda 1_d^\top t, \text{ and} \quad (20)$$
$$H^*(v) = F^*(\alpha) + r\sum_{j=1}^{d} I_{\mathcal{C}}(\delta_j,\beta_j,\nu_j). \quad (21)$$

Here, we have introduced the constant r to balance the scale of α against κ, with the choice $r = \|X\|$ in experiments, since $\|K\|^2 = \|X^\top X + r^2 I\|^2 \leqslant \|X\|^2 + r^2$. Algorithm 1 applies and we get Algorithm 3 to solve the problem (12). We see that G is separable function in terms of its variables w, η and t. And so is H^* separable in terms of α and the triplets $(\delta_j, \beta_j, \nu_j)$ for all j. This makes it easy to compute the proximal operators needed for G and H^*. Now the result (7) applies, and we get the result. □

Discussion. We can make the following remarks:

1. Algorithm 3 is the first ever $O(1/T)$ efficient *first-order* algorithm for (3), since it accesses X only through matrix-vector multiplications.
2. One may also equate problems (18) and (6) with the following mappings of u, v, G, H^*, for which CP achieves $O(1/T^2)$ convergence rate when F is smooth (see (8)).

$$u = \left[w^\top\ \eta^\top\ t^\top\right]^\top, v = \alpha, K = [X\ 0_{n\times 2d}],$$
$$H^*(v) = F^*(\alpha),$$
$$G(u) = \frac{\lambda}{2}\Gamma(\eta) + \lambda 1_d^\top t + \sum_{j=1}^{d} I_{\mathcal{C}}(w_j,\eta_j,t_j).$$

   But, prox_G is not easy to compute except for simple norms like the ℓ_1 or grouped-ℓ_1 norm, making the algorithm impractical for general norms.
3. We derive the following step size choices for CP-η: $\sigma = 1/n$ and $\tau = n/(2\|X\|^2)$.

Algorithm	$\mathrm{F}_{sq}(z)$	$\mathrm{F}_H(z)$
FISTA-η	$(d^3 + nd)/\sqrt{\epsilon}$	$(dn^2 + d\log d)/\sqrt{\epsilon}$
ADDM-η	d^2/ϵ	$(dn^2 + d\log d)/\epsilon$
CP-η	$(nd + d\log d)/\epsilon$	$(nd + d\log d)/\epsilon$

Table 1: Number of operations (in Big-O notation) needed to guarantee duality gap of (3) $\leq \epsilon$, for $\Omega_{\mathcal{H}}$ as in Example 1. $\mathrm{F}_{sq}(z) = \frac{1}{2}\|z-y\|_2^2$ is the square loss, $\mathrm{F}_H(z) = \sum_{i=1}^n \max(1-y_i z_i, 0)$ is the hinge loss.

5.4 General Discussion

We can differentiate the various algorithms for the problem (3) as follows:

- Between Alt-η and FISTA-η, the former does not have any known convergence rates, whereas the latter has a convergence rate of $O(1/T^2)$. But the per-step cost of FISTA-η is much higher because of backtracking. As seen in experiments the backtracking cost is quite high which leads to FISTA-η being impractical to use for general losses and norms.
- As discussed before, ADMM-η is similar to FISTA-η and Alt-η, since all of these assume easy computability of $\mathrm{prox}_{F\circ\tilde{X}}$, with $\tilde{X} = X$ for ADMM-η and $XD^{\frac{1}{2}}(\eta)$ for the other two algorithms. This subtle difference does make an impact, especially for losses like the squared loss, for which the proximal operator for ADMM-η is easier to obtain than those for FISTA-η and Alt-η.
- When $\mathrm{prox}_{F\circ X}$ is not very efficient to compute, the only choice we have is CP-η, which works with all norms and losses with easy to compute prox_F and prox_Γ.
- When $\mathrm{prox}_{F\circ X}$ is easy to compute, the choice of the algorithm depends on the cost of prox_Γ. Cheaper the cost of prox_Γ computation, better is CP-η in performance compared to ADMM-η. It is because of the experimental observation that ADMM-η takes far fewer iterations than CP-η, and costly prox_Γ does make CP-η run longer.
- ADMM-η is observed to be less sensitive to the conditioning of X. This may be because the inner CP routine does not depend on X explicitly, which is taken care of by $\mathrm{prox}_{F\circ X}$ as a black box. The dependence on X within the CP routine comes only through the step sizes τ and σ. This makes ADMM-η preferred in ill-conditioned cases.

6 Experiments

To illustrate the efficiency of CP-η and ADMM-η over existing algorithms, we choose the aforementioned tree-sparsity inducing norm $\Omega_{\mathcal{H}}$ (Example 1), which is popular in wavelet coefficients estimation. As discussed earlier, $\Omega_S^2(w) = \Omega_{\mathcal{H}}(w)$, but the primal-dual procedures are different because of the difference in the function Γ. Since time complexity to compute prox_Γ is identical in both cases ($O(d\log d)$), we choose only $\Omega_{\mathcal{H}}$ for the purpose of simulations. We include the following solvers: (1) FISTA-η, (2) Alt-η, (3) FISTA[1]

[1]The non-accelerated variant ISTA was very slow in experiments and hence excluded from the results.

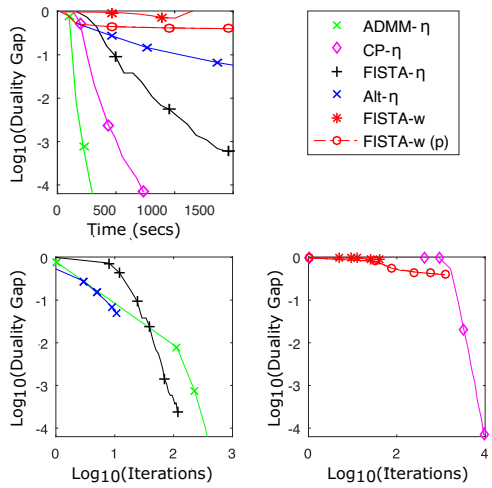

Figure 1: Duality gap convergence for squared loss with tree-structured norm (Example 1). Top: all algorithms. Bottom left: algorithms using prox_F, Bottom right: algorithms using $\text{prox}_{F \circ X}$.

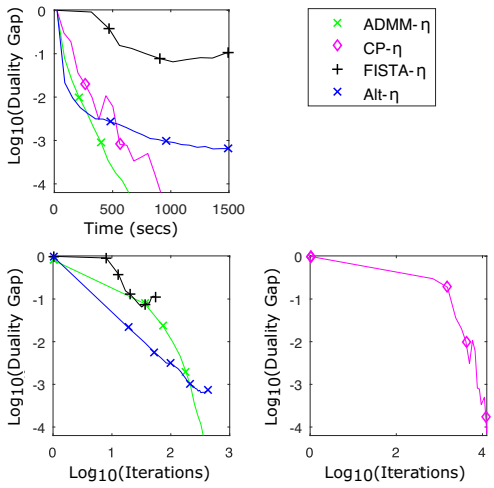

Figure 2: Duality gap convergence for hinge loss with tree-structured norm (Example 1). Top: all algorithms. Bottom left: algorithms using prox_F, Bottom right: algorithms using $\text{prox}_{F \circ X}$.

on primal (1) with prox_Ω computed using (a) decomposition algorithm ($O(d^2)$) given in [Obozinski and Bach, 2016], denoted as FISTA-w, (b) Picard iterations [Baldassarre *et al.*, 2012], denoted as FISTA-$w(p)$.

Setup. We perform numerical simulations[2] by generating synthetic data. Following [Bach *et al.*, 2011], we generate $X \in \mathbb{R}^{n \times d}$ as $X_{ij} \sim \mathcal{N}(0,1)$. For the ground truth model w^*, we assumed a tree structure with uniform branching factor of 4 and a depth k. We generated w^* uniformly at random from $[0,1]^d$ and set $s = 0.75d$ randomly chosen indices to 0 satisfying the tree structure. The labels y were generated as $y = Xw + \xi$ for the squared loss examples, and $y = \text{sign}(Xw + \xi)$ for the hinge loss examples, where ξ is a standard Gaussian noise. We fixed $n = 1000$, $d = 15000$, $\lambda = 0.01$, and the convergence criteria was the relative duality gap (with threshold $\epsilon = 10^{-4}$).

6.1 Squared Loss

In this case, $\text{prox}_{F \circ X}$ can be efficiently computed using matrix-vector multiplications in each iteration, qualifying ADDM-η as a first-order algorithm. We make the following inferences from the simulation plots given in Figure 1.

- All the solutions for the problem (3) do better than that of (1). Both FISTA-w and FISTA-w-p do not converge within a limit of 30 minutes, the latter though being faster than the former suffers from approximate solutions at each step.
- Both CP-η and ADMM-η are better than all the compared algorithms in running time, justifying the claims made in the paper.
- In general ADMM-η is faster than CP-η because of efficient computation of $\text{prox}_{F \circ X}$ and overall lesser number of iterations.

[2]Conducted on a Ubuntu PC with Core i7 processor, 8G RAM.

	ADMM-η	CP-η
Sq. loss	458 / 341s / 0.53s	9260 / 734s / 0.08s
Hinge.loss	360 / 646s / 1.97s	12810 / 923s / 0.07s

Table 2: No. of Iterations/Total time/Time-per-iteration

6.2 Hinge Loss

$\text{prox}_{F \circ X}$ is computed in this case through solving an SVM [W. Kienzle, 2006], which is based on an SMO algorithm ($O(dn^2)$). Hence CP-η is the only first-order algorithm in this case. Similar to the squared loss case, both the proposed algorithms perform better than the rest as shown in Figure 2.

6.3 Comparison Of CP-η And ADMM-η

Although Table 2 may suggest that ADMM-η is better than CP-η for both loss functions, this is not always true, as it depends on d and n (see Table 1). When d increases, ADMM-η becomes expensive for square loss due to the $O(d^2)$ complexity. For hinge loss, both ADMM-η and CP-η have linear dependency on d, and hence theoretically, CP-η has no advantage over ADMM-η in very high dimensional scalings, for a fixed n. However, for ADMM-η, due to quadratic dependency on n (see Table 1), the per-step cost goes high on large sample settings. This dependency on n for ADMM-η is justified in Table 2, where we see that the per-step cost for ADMM-η goes much higher for hinge loss compared to square loss.

7 Conclusions and Future Directions

We studied efficient primal-dual algorithms to learn with Subquadratic norms. One may also investigate for potential extensions of these algorithms towards general reweighted least-squares formulations [Bach *et al.*, 2011] for norms. Study of inexact proximal opeartors for subquadratic norms also provides alternate directions.

References

[Argyriou *et al.*, 2008] A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.

[Bach *et al.*, 2011] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 4(1):1–106, 2011.

[Bach, 2010] F. Bach. Structured sparsity-inducing norms through submodular functions. In *Advances in Neural Information Processing Systems (NIPS)*, 2010.

[Bach, 2011a] F. Bach. Learning with submodular functions: A convex optimization perspective. *Foundations and Trends in Machine Learning*, 6-2-3:145–373, 2011.

[Bach, 2011b] F. Bach. Shaping level sets with submodular functions. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

[Baldassarre *et al.*, 2012] Luca Baldassarre, Jean Morales, Andreas Argyriou, and Massimiliano Pontil. A general framework for structured sparsity via proximal optimization. In *Artificial Intelligence and Statistics*, pages 82–90, 2012.

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

[Best and Chakravarti, 1990] M. J. Best and N. Chakravarti. Active set algorithms for isotonic regression: a unifying framework. *Mathematical Programming*, 47(1):425–439, 1990.

[Canu and Grandvalet, 1999] S. Canu and Y. Grandvalet. Outcomes of the equivalence of adaptive ridge with least absolute shrinkage. *Adv. NIPS*, 1999.

[Chambolle and Pock, 2011] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vis.*, 40(1):120–145, May 2011.

[Daubechies *et al.*, 2010] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk. Iteratively reweighted least squares minimization for sparse recovery. *Communications on Pure and Applied Mathematics*, 63(1):1–38, 2010.

[Jenatton *et al.*, 2011a] R. Jenatton, A. Gramfort, V. Michel, G. Obozinski, F. Bach, and B. Thirion. Multi-scale mining of fmri data with hierarchical structured sparsity. *Pattern Recognition in NeuroImaging (PRNI)*, 2011.

[Jenatton *et al.*, 2011b] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach. Proximal methods for hierarchical sparse coding. *J. Mach. Learn. Res.*, 12:2297–2334, July 2011.

[Kloft *et al.*, 2011] M. Kloft, U. Brefeld, S. Sonnenburg, and A. Zien. l_p-norm multiple kernel learning. *Journal of Machine Learning Research*, 12:953–997, Mar 2011.

[Kyrillidis, 2016] Kyrillidis. Structured sparsity: discrete and convex approaches. *Foundations and Trends in Machine Learning*, 2016.

[Lee *et al.*, 2006] Su-In Lee, Honglak Lee, Pieter Abbeel, and Andrew Y. Ng. Efficient l1 regularized logistic regression. In *Neural Information Processing Systems*, 2006.

[Mairal *et al.*, 2014] J. Mairal, F. Bach, and J. Ponce. Sparse modeling for image and vision processing. *Foundations and Trends in Computer Graphics and Vision*, 8(2–3):85–283, 2014.

[McDonald *et al.*, 2016] Andrew M McDonald, Massimiliano Pontil, and Dimitris Stamos. New perspectives on k-support and cluster norms. *Journal of Machine Learning Research*, 17(155):1–38, 2016.

[Micchelli *et al.*, 2013] C. Micchelli, J. Morales, and M. Pontil. Regularizers for structured sparsity. *Advances in Computational Mathematics*, 38(3):455–489, 2013.

[Nesterov, 2007] Y. Nesterov. Gradient methods for minimizing composite objective function. *CORE Discussion Paper*, 2007.

[Nesterov, 2013] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2013.

[Obozinski and Bach, 2016] Guillaume Obozinski and Francis Bach. A unified perspective on convex structured sparsity: Hierarchical, symmetric, submodular norms and beyond. Technical report, HAL-01412385, December 2016.

[Obozinski *et al.*, 2011] G. Obozinski, L. Jacob, and J.P. Vert. Group lasso with overlaps: The latent group lasso approach. *ArXiv preprint:1110.0413v1*, 2011.

[Pardalos and Xue, 1999] Panos M Pardalos and Guoliang Xue. Algorithms for a class of isotonic regression problems. *Algorithmica*, 23(3):211–222, 1999.

[Parikh and Boyd, 2014] N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):123–231, 2014.

[Sankaran *et al.*, 2017] Raman Sankaran, Francis Bach, and Chiranjib Bhattacharya. Identifying groups of strongly correlated variables through smoothed ordered weighted l_1-norms. In *Artificial Intelligence and Statistics*, pages 1123–1131, 2017.

[Scheinberg *et al.*, 2014] Katya Scheinberg, Donald Goldfarb, and Xi Bai. Fast first-order methods for composite convex optimization with backtracking. *Foundations of Computational Mathematics*, 14(3):389–417, 2014.

[Tibshirani, 1994] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58:267–288, 1994.

[Vapnik, 2000] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 2000.

[W. Kienzle, 2006] K. Chellapilla W. Kienzle. Personalized handwriting recognition via biased regularization. In *Proceedings of the International Conference on Machine Learning(ICML)*, 2006.

[Yan and Bien, 2015] Xiaohan Yan and Jacob Bien. Hierarchical sparse modeling: A choice of two group lasso formulations. *arXiv preprint arXiv:1512.01631*, 2015.