Fast column generation for atomic norm regularization

Marina Vinyes
Université Paris-Est, LIGM (UMR8049)
Ecole des Ponts
Marne-la-Vallée, France
marina.vinyes@imagine.enpc.fr

Guillaume Obozinski
Université Paris-Est, LIGM (UMR8049)
Ecole des Ponts
Marne-la-Vallée, France
guillaume.obozinski@enpc.fr

Abstract

We consider optimization problems that consist in minimizing a quadratic function under an *atomic norm*[1] regularization or constraint. In the line of work on conditional gradient algorithms, we show that the fully corrective Frank-Wolfe (FCFW) algorithm — which is most naturally reformulated as a *column generation* algorithm in the regularized case — can be made particularly efficient for difficult problems in this family by solving the *simplicial* or *conical* subproblems produced by FCFW using a special instance of a classical *active set algorithm for quadratic programming* (Nocedal and Wright, 2006) that generalizes the min-norm point algorithm (Wolfe, 1976).

Our experiments show that the algorithm takes advantages of warm-starts and of the sparsity induced by the norm, displays fast linear convergence, and clearly outperforms the state-of-the-art, for both complex and classical norms, including the standard group Lasso.

1 INTRODUCTION

A number of problems in machine learning and structured optimization involve either structured convex constraint sets that are defined as the intersection of a number of simple convex sets or dually, norms of sets that are defined as convex hull of either extreme points or of a collection of sets. A broad class of convex regularizers that can be used to encode a priori knowledge on the structure of the objects to estimate have been described as *atomic norms* and *atomic gauges* by Chandrasekaran et al. (2012). The concept of atomic norm has found several applications to design sparsity inducing norms for vectors (Jacob et al., 2009; Obozinski et al., 2011), matrices (Richard et al., 2014; Foygel et al., 2012) and tensors (Tomioka and Suzuki, 2013; Liu et al., 2013; Wimalawarne et al., 2014).

A number of these norms remain difficult to use in practice because it is in general not possible to compute the associated *proximal operator* or even the norm itself at a reasonable cost. However, the dual norm which is defined as a supremum of dot products with the atoms that define the norm can often be computed efficiently because of the structure of the set of atoms. Also a number of atomic norms are actually naturally defined as *infimal convolution* of other norms (Jacob et al., 2009; Tomioka and Suzuki, 2013; Liu et al., 2013) and this structure has been used to design either block-coordinate descent approaches or dual ADMM optimization schemes (Tomioka and Suzuki, 2013) involving latent variables associated with the elementary norms convolved.

In this paper, we propose to solve problems regularized or constrained by atomic norms using a fully corrective Frank-Wolfe algorithm—which can be reformulated as simple column generation algorithm in the regularized case—combined with a dedicated active-set algorithm for quadratic programming. Our experiments show that we achieve state-of-the-art performance. We also include a formal proof of the correspondance between the column generation algorithm and Fully Corrective Frank-Wolfe.

After a review of the concept of atomic norms, as well some illustrations, we present a number of the main algorithmic approaches that have been proposed. We then present the scheme we propose and finally some experiments on synthetic and real datasets.

[1] or more generally *atomic gauge*

Proceedings of the 20^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2017, Fort Lauderdale, Florida, USA. JMLR: W&CP volume 54. Copyright 2017 by the author(s).

1.1 Notations

$[\![p]\!]$ denotes the set $\{1, \dots, p\}$. If $x \in \mathbb{R}^p$, x_G denotes the subvector of x whose entries are indexed by a set $G \in [\![p]\!]$. Given a function ψ, ψ^* denotes its Fenchel conjugate $\psi^*(s) := \max_x \langle s, x\rangle - \psi(x)$. $\|M\|_{\text{tr}}$ denotes the trace norm of the matrix M defined as the ℓ_1-norm of its singular values.

2 ATOMIC NORMS

In many machine learning applications applications, and particularly for ill-posed problems, models are constrained structurally so they have a simple representation in terms of some fundamental elements. Examples of such elements include sparse vectors for many sparsity inducing norms, rank-one matrices for the trace norm or low-rank tensors as used in the *nuclear tensor norm* (Liu et al., 2013). We call *atoms* these elements and *atomic set* $\mathcal{A}$ their (possibly infinite) collection. Assuming $\mathcal{A}$ is bounded and centrally symmetric, and provided its convex hull $C_{\mathcal{A}}$ has non empty interior, we can define an atomic norm $\gamma_{\mathcal{A}}$ as the norm of unit ball $C_{\mathcal{A}}$. It can be shown that (in a finite dimensional space) $\gamma_{\mathcal{A}}(x) := \inf\{\sum_{a\in\mathcal{A}} c_a \mid \sum_{a\in A} c_a a = x,\ c_a \geq 0,\ a \in \mathcal{A}\}$. The polar norm or dual norm is defined as: $\gamma^\circ_{\mathcal{A}}(s) := \sup_{a\in\mathcal{A}} \langle s, a\rangle$. If $\mathcal{A}$ is not symmetric, or if $C_{\mathcal{A}}$ is empty, as long as $\mathcal{A}$ contains the origin and is closed, $\gamma_{\mathcal{A}}$ can still be defined as a *gauge* instead of a norm and the theory and algorithms presented in this paper still apply. We restrict the discussion to norms for simplicity. For a reference on gauges, see Rockafellar (1997).

We consider in this paper formulations in which an atomic norm is used as a regularizer, and which lead to an optimization problem of the form

$$\min_{x\in\mathbb{R}^p} f(x) + \gamma_{\mathcal{A}}(x), \tag{1}$$

where f is a *quadratic* function. The case where f is more generally twice differentiable is obviously of interest, but beyond the scope of this work.

2.1 Examples of atomic norms

Lasso. The Lasso is a natural example of atomic norm, whose atoms are the $(\pm e_i)_{i\in[\![p]\!]}$, where the $(e_i)_{i\in[\![p]\!]}$ is the canonical basis of $\mathbb{R}^p$. The Lasso polar norm is defined as $\Omega^\circ_{\text{Lasso}}(s) = \max_{i\in[\![p]\!]} |s_i|$.

Latent group lasso (LGL). The norms introduced in Jacob et al. (2009) are a strong motivating example. For instance Obozinski and Bach (2016) show that a broad family of tight relaxations for structured sparsity can be written in LGL form. Given a collection of sets $\mathcal{B}$ covering $[\![p]\!]$ and which can overlap, and fixed positive weights δ_B for each set $B \in \mathcal{B}$, the atoms of LGL norm are the vectors of norm δ_B^{-1} and support in $\mathcal{B}$. The polar LGL norm is defined as $\Omega^\circ_{\text{LGL}}(s) = \max_{B\in\mathcal{B}} \delta_B^{-1} \|s_B\|_2$. In the particular case where $\mathcal{B}$ form a partition of $[\![p]\!]$ we recover the group Lasso norm. Maurer and Pontil (2012) consider a generalization to a broader family of atomic norms with dual norms of the form $\sup_{M\in\mathcal{M}} \|Ms\|_2$, where $\mathcal{M}$ is a collection of operators. Matrix counterparts of the latent group Lasso norms are the *latent group trace norms* (Tomioka and Suzuki, 2013; Wimalawarne et al., 2014).

Additive decompositions. There has been interest in the literature for additive matrix decompositions (Agarwal et al., 2012), the most classical example being "sparse+low rank decompositions" which have been proposed for robust PCA and multitask learning (Candès et al., 2011; Chandrasekaran et al., 2011). This formulation leads to a problem of the form $\min_{L,S} f(L+S) + \mu\|L\|_{\text{tr}} + \lambda\|S\|_1$, which under the form $\min_M f(M) + \gamma_{\mathcal{A}}(M)$ with $\gamma_{\mathcal{A}}$ the atomic norm where $\mathcal{A} \subset \mathbb{R}^{p_1\times p_2}$ is defined as

$$\begin{aligned}
\mathcal{A} &:= \lambda\,\mathcal{A}_1 \cup \mu\,\mathcal{A}_{\text{tr}}, \qquad \text{where}\\
\mathcal{A}_1 &:= \big\{\pm e_i e_j^\top,\ (i,j) \in [\![p_1]\!] \times [\![p_2]\!]\big\},\\
\mathcal{A}_{\text{tr}} &:= \big\{uv^\top,\ \|u\|_2 = \|v\|_2 = 1\big\}.
\end{aligned}$$

As a consequence, $C^\circ_{\mathcal{A}} = \frac{1}{\lambda} C^\circ_1 \cap \frac{1}{\mu} C^\circ_{\text{tr}}$ with C°_1 a unit ℓ_∞ ball and C°_{tr} a unit spectral norm ball.

Convex sparse SVD and PCA. A third example are the norms introduced in Richard et al. (2014), including the (k,q)-trace norm for which

$$\mathcal{A} := \bigcup\{\mathcal{A}_{I,J} \mid (I,J) \subset [\![p_1]\!] \times [\![p_2]\!],\ |I| = k, |J| = q\},$$

$$\text{with} \qquad \mathcal{A}_{I,J} := \{uv^\top \in \mathcal{A}_{\text{tr}} \mid \|u\|_0 \leq k,\ \|v\|_0 \leq q\},$$

and the sparse-PCA norm[2] for which

$$\mathcal{A} := \bigcup\{\mathcal{A}_{I,\geq} \mid I \subset [\![p_1]\!],\ |I| = k\},$$

with $\mathcal{A}_{I,\geq} := \{uu^\top \mid u \in \mathcal{A}_I\}$, and $\mathcal{A}_I$ defined like $\mathcal{A}_B$ for LGL.

Beyond these examples a number of structured convex optimization problems encountered in machine learning and operations research that involve combinatorial or structured tasks such as finding permutations or alignments, convex relaxation of structured matrix factorization problems (Bach et al., 2008; Ding et al., 2010), Procrustes analysis, etc, involve difficult convex constraint sets such as elliptope, the Birkhoff polytope, the set of doubly nonnegative matrices that are naturally written (themselves or their polar) as intersections of simpler sets such as the p.s.d. cone, the positive orthant, simplices, hypercubes, etc, and which lead to optimization problems whose duals are regularized by associated atomic norms.

[2] In fact this is not a *norm* but only a *gauge*.

2.2 Existing algorithmic approaches

2.2.1 Conditional gradient algorithms

For many[3] of these norms, it is assumed that an efficient algorithm is available to compute $\operatorname{argmax}_{a\in\mathcal{A}}\langle a, s\rangle$. For the case of the constrained problem

$$\min_x f(x) \quad \text{s.t.} \quad x \in C_{\mathcal{A}}, \tag{2}$$

this has motivated a number of authors to suggest variants of the *conditional gradient algorithm*, also known as the Frank-Wolfe algorithm when the objective is quadratic, as a tool of choice to solve problems with atomic norm constraints. Indeed, the principle of conditional gradient algorithms is to build a sequence of approximations to the solution of the problem as convex combinations of extreme points of the constraint sets, which here correspond to atoms, so that the expansion take the form $x = \sum_{i=1}^t c_i a_i$ with $\sum_{i=1}^t c_i = 1$. This procedure guarantees a feasible sequence. At each iteration a new atom, also called Frank-Wolfe direction or *forward* direction, is added in the expansion. This atom is the extreme point of the constraint set defined by $a^{t+1} := \operatorname{argmax}_{a\in\mathcal{A}}\langle a, -\nabla f(x^t)\rangle$. The Frank-Wolfe (FW) algorithm writes

$$x_{\text{FW}}^{t+1} = (1-\eta^t)\, x^t + \eta^t a^{t+1},$$

where $\eta^t \in [0,1]$ is a scalar stepsize and $x^0 = 0$. It can be set to $\frac{1}{1+t}$ or found by line search.

Other variants of FW algorithms have been proposed, notably, FW with away steps (which we do not describe here), pairwise FW (PWFW) and fully corrective Frank-Wolfe (FCFW). We refer the reader to Lacoste-Julien and Jaggi (2015) for a detailed presentation and summarize hereafter the form of the different updates for PWFW and FCFW. The active set of atoms $\mathcal{A}^t$ at time t is recursively defined by $\mathcal{A}^{t+1} = \widetilde{\mathcal{A}}^t \cup \{a^{t+1}\}$ with $\widetilde{\mathcal{A}}^t$ the set of *active* atoms of $\mathcal{A}^t$ at the end of iteration t, i.e. the ones that contributed with a non-zero coefficient in the expansion of x^t.
PWFW makes use of a *backward* direction also called *away atom*, and defined as $a_B^{t+1} = \operatorname{argmax}_{a\in\widetilde{\mathcal{A}}^t}\langle a, \nabla f(x^t)\rangle$, i.e. it is the active atom of largest projection on the gradient direction. The idea in PWFW is to move by transferring weight from the away atom a_B^{t+1} to the FW atom a^{t+1}:

$$x_{\text{PWFW}}^{t+1} = x^t + \eta_p^t\,(a^{t+1} - a_B^{t+1}),$$

where $\eta^t \in [0, c_B^t]$, with $c_B^t \geq 0$ the weight attributed to atom a_B^{t+1} at iteration t, and η^t is found by line search.

[3]This is not true for the norms introduced in Richard et al. (2014) whose dual are NP-hard to compute, but for which reasonable heuristic algorithms or relaxations are available.

The optimal step sizes $\eta^t \in \mathbb{R}$ for FW and PWFW are easily obtained in closed form when f is quadratic.

In FCFW, all weights are reoptimized at each iteration:

$$x_{\text{FCFW}}^{t+1} = \operatorname{argmin}_x f(x) \quad \text{s.t.} \quad x \in \text{Convhull}(\mathcal{A}^{t+1}).$$

If $\mathcal{A}^t = \{a_1, ..., a_{k_t}\}$, where $k_t \leq t$ is the number of atoms in $\mathcal{A}^t$, the subproblem that has to be solved at each iteration t of FCFW rewrites

$$\min_{c\geq 0} f\Big(\sum_{i=1}^{k_t} c_i a_i\Big) \quad \text{s.t.} \quad \sum_{i=1}^{k_t} c_i = 1. \tag{3}$$

Lacoste-Julien and Jaggi (2015) show that PWFW and FCFW converge linearly for strongly convex objectives when $\mathcal{A}$ is finite.

Rao et al. (2015) propose a variant of FCFW to solve (2) for f smooth and specifically for atomic norm constraints, with an enhancing "backward step" which applies hard-thresholding to the coefficients c^t. To solve (3) they use a projected gradient algorithm.

Beyond constrained optimization problems, the basic conditional gradient algorithm (corresponding to plain FW when f is quadratic) has been generalized to solve problems of the form $\min_x f(x) + \psi(x)$ where the set constraint $C_{\mathcal{A}}$ is replaced by a proper convex function ψ for which the subgradient of ψ^* can be computed efficiently (Bredies et al., 2009; Yu et al., 2014). Bach (2015) shows that the obtained algorithm can be interpreted as a dual mirror descent algorithm. Yu et al. (2014); Bach (2015) and Nesterov et al. (2015) prove sublinear convergence rates for these algorithms. Corresponding generalizations of PWFW and FCFW are however not obvious. As exploited in Yu et al. (2014); Harchaoui et al. (2015), if $\psi = h \circ \gamma_{\mathcal{A}}$, with h a nondecreasing convex function and $\gamma_{\mathcal{A}}$ an atomic norm, and if an upper bound ρ can be specified a priori on $\gamma_{\mathcal{A}}(x^\star)$ for $x^\star$ a solution of the problem, it can reformulated as

$$\min_{x,\tau} f(x) + h(\tau) \quad \text{s.t.} \quad \gamma_{\mathcal{A}}(x) \leq \tau, \quad \tau \leq \rho, \tag{4}$$

and it is natural to apply the different variant of Frank-Wolfe on the variable (x, τ), because the FW direction is easy to compute (see Section 3.1).

2.2.2 Proximal block-coordinate descent

In the context where they are applicable, proximal gradient methods provide an appealing alternative to Frank-Wolfe algorithms. However, the former require to be able to compute efficiently the *proximal operator* of the norm $\gamma_{\mathcal{A}}$ appearing in the objective, which is typically more difficult to compute than the Frank-Wolfe direction.

For a number of atomic norms, we have $\mathcal{A} = \bigcup_{j=1}^{J} C_j$ where C_j are convex sets. As a consequence the polar norm takes the form $\gamma^\circ_{\mathcal{A}}(s) = \max_j \gamma^\circ_{C_j}$, with γ_{C_i} the atomic norm (or gauge) associated with the set C_i, and it is a standard result that

$$\gamma_{\mathcal{A}}(x) = \inf\{\gamma_{C_1}(z_1) + \ldots + \gamma_{C_J}(z_J) \mid z_1 + \ldots + z_J = x\}.$$

Technically, $\gamma_{\mathcal{A}}$ is called the *infimal convolution* of the norms $(\gamma_{C_i})_i$ (see Rockafellar, 1997). In fact most of the norms that we presented in section 2.1 are of this form, including LGL norms, latent group trace norms, norms arising from additive decomposition (obviously by construction), and the norms for sparse SVD and sparse PCA.

For all these norms, problem (1) can be reformulated as

$$\min_{z_1,\ldots,z_J} f(z_1 + \ldots + z_J) + \gamma_{C_1}(z_1) + \ldots + \gamma_{C_J}(z_J).$$

Since the objective is then a sum of a smooth and of a separable function, randomized proximal block-coordinate descent algorithm are typical candidates. These algorithms have attracted a lot of attention in the recent literature (see **?**, and reference therein) and have been applied successfully to a number of formulations involving convex sparsity inducing regularizers (Shalev-Shwartz and Tewari, 2011; Friedman et al., 2010; Gu et al., 2016), where they achieve state-of-the-art performance. Such BCD algorithms where the ones proposed for the norms proposed in Jacob et al. (2009) and Richard et al. (2014).

Unfortunately these algorithms are slow in general even if f is strongly convex because of the composition with the linear mapping $(z_1, \ldots, z_J) \mapsto z_1 + \ldots + z_J$. Intuitively if the atoms of the different norms are similar, then the formulation is badly conditioned. If they are different or essentially decorrelated, BCD remains one of the most efficient algorithms (Shalev-Shwartz and Tewari, 2011; Gu et al., 2016).

3 PIVOTING FRANK-WOLFE

After reviewing the form of the corrective step of FCFW and reformulating FCFW in the regularized case as a column generation algorithm, we introduce active-set algorithms to solve efficiently sequences of corrective steps.

3.1 Simplicial and conical subproblems

We focus on the sequence of subproblems that need to be solved at the corrective step of FCFW. Let $k_t := |\mathcal{A}^t|$ be the number of selected atoms at iteration t, and $A^t \in \mathbb{R}^{p \times k_t}$, the matrix whose columns are the atoms $\mathcal{A}^t$, then, for the constrained problem (2), the subproblem is the *simplicial* problem:

$$\min_c f(A^t c) \quad \text{s.t.} \quad c \in \Delta^{k_t}, \tag{5}$$

with $\Delta^k := \{c \in \mathbb{R}^k_+ \mid \sum_{i=1}^k c_i = 1\}$ the canonical simplex. The regularized problem (1) can be reformulated as the constrained optimization problem (4) on a truncated cone, provided the truncation level ρ is an upper bound of the value of $\gamma_{\mathcal{A}}$ at the optimum. Actually, if ρ is sufficiently large, several Frank-Wolfe algorithms do not depend any longer on the value of ρ and can be interpreted as algorithms in which whole extreme rays of the cone $\{(x;\tau) \mid \gamma_{\mathcal{A}}(x) \le \tau\}$ enter the active set via the linear minimization oracle, and where the original cone is locally approximated from inside by the simplicial cone obtained as their conical hull. In particular in the case of FCFW, the subproblem considered at the t-th iteration takes the form of the *conical* problem

$$\min_c f(A^t c) + \sum_i c_i \quad \text{s.t.} \quad c \ge 0, \tag{6}$$

which is simply a Lasso problem with positivity constraints when f is quadratic. The fact that problem (1) can be solved by as sequence of problems of the form (6) is shown in Harchaoui et al. (2015, Sec. 5), who argue that this leads to an algorithm no worse and possibly better. We formally show that the simple column generating scheme presented as Algorithm 1 is in fact exactly equivalent to FCFW applied to the truncated cone formulation as soon as ρ is large enough:

Proposition 1. *If f is assumed lower bounded by 0 and if $\rho > f(0)$, or more generally if the level sets of $x \mapsto f(x) + \gamma_{\mathcal{A}}(x)$ are bounded and ρ is sufficiently large, then the sequence $(\bar{x}^t)_t$ produced by the FCFW algorithm applied to the truncated cone constrained problem (4) and initialized at $(\bar{x}^0; \tau^0) = (0; 0)$ is the same as the sequence $(x^t)_t$ produced by Algorithm 1 initialized with $x^0 = 0$, with equivalent sequences of subproblems, active sets and decomposition coefficients.*

See the appendix for a proof. As discussed as well in the appendix, a variant of Algorithm 1 without pruning of the atoms with zero coefficients (at step 7) is derived very naturally as the dual of a cutting plane algorithm.

3.2 Leveraging active-set algorithms for quadratic programming

Problems (5) and (6) can efficiently be solved by a number of algorithms. In particular, an appropriate variant LARS algorithm solves both problem in a finite number of iterations and it is fast if the solution in sparse, in spite of the fact that it solves exactly a sequence of linear systems. Interior point algorithms can always be used, and are often considered to be a

Algorithm 1 Column generation

Require: f convex differentiable, tolerance ϵ
Initialization: $x^0 = 0$, $A^0 = \varnothing$, $k_0 = 0$, $t = 1$
repeat
  $a_t \leftarrow \arg\max_{a \in \mathcal{A}} \langle -\nabla f(x^{t-1}), a \rangle$
  $A^t \leftarrow [A^{t-1}, a_t]$
  $c^t \leftarrow \arg\min_{c \geq 0} f\big(A^t c\big) + \|c\|_1$
  $I \leftarrow \{i \mid c_i^t > 0\}$,
  $c^t \leftarrow c_I^t$
  $A^t \leftarrow A_{\cdot,I}^t$
  $x^t \leftarrow A^t c^t$
  $t \leftarrow t + 1$
until $\max_{a \in \mathcal{A}} \langle -\nabla f(x^{t-1}), a \rangle \leq \epsilon$

natural choice to solve this step in the literature. For larger scale problems, and if f has Lipschitz gradients (which is obviously the case for a quadratic function), the forward-backward proximal algorithm can be used as well, since the projection on the simplex for (5) and the asymmetric soft-thresholding for (6) can be computed efficiently. For the constrained case, this is the algorithm used by Rao et al. (2015).

In our case, we need to solve a sequence of problems of the form (5) or (6), that differ each from the previous one by the addition of a single atom. So being able to use *warm-starts* is key! If the simplicial problems remains of small size, and if the corresponding Hessians can be computed efficiently, using second order algorithms is likely to outperform first order methods. But the LARS and interior point methods cannot take advantage of warm-starts. Thus, when f is quadratic, we propose to use *active set algorithms for convex quadratic programming* (Nocedal and Wright, 2006; Forsgren et al., 2015). In particular, following[4] Bach (2013, Chap. 7.12), we propose to apply the active-set algorithm of Nocedal and Wright (2006, Chap. 16.5) to iteratively solve (5) and (6). This algorithm takes the very simple[5] form of Algorithm 2. In fact, as noted in Bach (2013, Chap. 9.2), this algorithm is a generalization of the famous *min-norm point algorithm* (Wolfe, 1976), the latter being recovered when the Hessian is the identity.

Algorithm 2 is illustrated in Figure 1. The obtained iterates always remain in the positive orthant (i.e. primal feasible). Each update of c in Algorithm 2 is called a *pivot*, which is either *full-step* or *drop-step*. Given a collection of active atoms indexed by a set J, the solution d of the non-constrained quadratic program restricted to this set of atoms and obtained by removing the positivity constraints is computed (line 4). If d lies in the positive orthant, we set $c = d$, and we say that we perform a *full-step*. In that case, the index of an atom that must become active (if any), based on gradients, is added to J. If $d \notin \mathbb{R}_+^{|J|}$, a *drop-step* is performed: c is updated as the intersection between segment $[c_{\text{old}}, d]$ and the positive orthant, and the index i such that $c_i = 0$ is dropped from J (line 13). The algorithm stops if after a full-step, no new index is added in J.

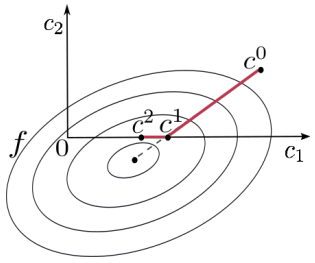


Figure 1: Illustration of Algorithm 2. Here, it converges after a *drop-step* (variable c_2 is dropped) leading to c^1 followed by a *full-step* (along c_1) leading to c^2.

Algorithm 2 $[c, J]$=Active-set(H, b, c_0, J_0)

1: **Solves:** $P := \min_c c^\top H c + b^\top c$, s.t. $c \geq 0$
2: **Initialization:** $c = c_0$, $J = J_0$,
3: **repeat**
4: $d \leftarrow {H_{J,J}}^{-1} b_J$
5: **if** $d \geq 0$,
6: $c \leftarrow d$ ▷ full-step
7: $g \leftarrow Hc + b$
8: $k \leftarrow \arg\min_{i \in J_0 \setminus J} g_i$
9: **if** $g_k \geq 0$, **then break, else** $J \leftarrow J \bigcup \{k\}$ **end**
10: **else**
11: $i^* \leftarrow \arg\min_i \frac{c_i}{c_i - d_i}$ s.t. $c_i - d_i > 0$, $d_i < 0$
12: $\tau \leftarrow \frac{c_{i^*}}{c_{i^*} - d_{i^*}}$
13: $J \leftarrow J \backslash \{i^*\}$ ▷ drop-step
14: $c \leftarrow c + \tau(d - c)$
15: **end**
16: **until** $g_{J_0 \setminus J} \geq 0$
17: **return** c, J

3.3 Convergence and computational cost

In this section, we discuss first the convergence of the algorithm and the number of pivots needed for convergence, and the the cost of each pivot.

Algorithm 2 is an instance of min-norm point (MNP) with a general quadratic instead of Euclidean distance,

[4] Bach (2013) proposed to use this active-set algorithm to optimize convex objectives involving the Lovász extension of a submodular function.

[5] Despite the fact that, in the context of a simplicial algorithms, the polyhedral constraints sets of (5) and (6) as convex hulls, the algorithm of Nocedal and Wright (2006, Chap. 16.5) actually exploits their structure as intersections of half-spaces, and thus the active constraints of the algorithm actually correspond counter-intuitively to dropped atoms.

but the algorithm is affine invariant, so the convergence is the same. MNP is known to be finitely convergent. The positive orthant in dimension k_t has at most 2^{k_t} faces which is a naive bound on the number of pivots in the active-set at iteration t of FCFW. But, Lacoste-Julien and Jaggi (2015) prove that MNP is linearly convergent. In practice, the solution is most of the time either strictly inside the orthant or in one of the $k-1$ dimensional faces in which case it is in fact found in just 1 or respectively 2 iterations! The number of pivots per call is illustrated in Figure 4 upper left.

Let $s = \max_{a\in\mathcal{A}} \|a\|_0$ be the sparsity of the atoms, k the number of active atoms at iteration t and $H^t = A^{t\top} Q A^t$ the Hessian of the quadratic problem in the active set, where Q is the Hessian of the quadratic function f.

The cost of one pivot is the cost of computing the Hessian H^t and its inverse, which is $\mathcal{O}\big(\min(k^2s^2, kps + k^2s)\big)$ for building the Hessian and an extra $\mathcal{O}\big(k^3\big)$ for the inversion. In the active-set with warm starts we only add or remove one atom at a time. We can take advantage of this to efficiently update the Hessian H^t and its inverse with rank one updates. The computational cost for updating the Hessian is $\mathcal{O}\big(\min(ks^2, ps + ks)\big)$ when an atom is added and $\mathcal{O}(k)$ when removing an atom. The additional cost to update $(H^t)^{-1}$ is then just $\mathcal{O}(k^2)$ in both cases. See the appendix for more details on the rank one updates.

4 EXPERIMENTS

In this section, we report experiments that illustrate the computational efficiency of the proposed algorithm. We consider linear regression problems of the form of (1) with $f(w) = 1/2\|Xw - y\|^2$, where X is a design matrix and $\gamma_{\mathcal{A}}$ the LGL or the sparse-PCA norms described in Section 2. We also considered the constrained version for LGL, $\min_x f(x)$ s.t. $\Omega_{\text{LGL}}(w) \le \rho$, in section 4.2.

Section 4.1 compares the performance of our proposed algorithm with state-of-the-art algorithms for the group Lasso. Section 4.2 presents comparisons with the variants of Frank-Wolfe and with COGEnT on problem involving the latent group Lasso. Section 4.3 provides a comparison with a version of FCFW relying on interior-point solver on larger scale problems. Sections 4.3 and 4.4 provide comparisons with randomized block proximal coordinate descent algorithms. Most experiments are on simulated data to control characteristics of the experiments, except in section 4.3.

4.1 Classical group Lasso

We consider an example with group Lasso regularization with groups of size 10, $\mathcal{B} = \big\{\{1,\dots,10\},\{11,\dots,20\},\dots\big\}$. We choose the support of the parameter $w_0 \in \mathbb{R}^{1000}$ of the model to be $\{1,\dots,50\}$ and all non zero coefficients are set to 2. We generate $n = 200$ examples $(y_i)_{i=1,..,n}$ from $y = x^\top w + \varepsilon$. Block Coordinate Descent (BCD) algorithms are the standard method for this problems but they suffer slow convergence when the design matrix is highly correlated. In this experiment we choose a highly correlated design matrix (with singular values in $\{1, 0.9^2, .., 0.9^{2(p-2)}, 0.9^{2(p-1)}\}$) to highlight the advantages of our algorithm for the harder instances. We compared our algorithm to our own implementation of BCD and an enhanced BCD from Qin et al. (2013) (hyb-BCD). Figure 2 shows that we outperform both methods.

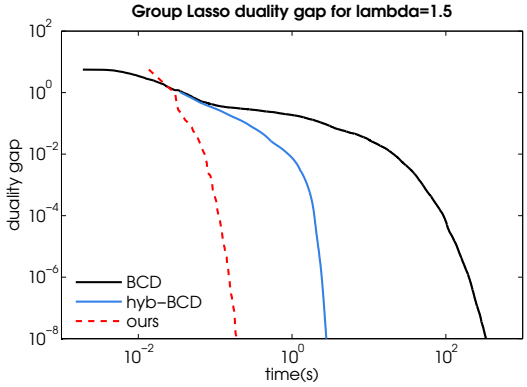


Figure 2: Experiment for classical group Lasso. *log-log* plot of progress of the duality gap.

4.2 k-chain latent group Lasso

We consider a toy example involving latent group Lasso regularization where the groups are chains of continuous indices of length $k = 8$, that is where the collection of group is $\mathcal{B} = \big\{\{1,\dots,k\},\{2,\dots,k+1\},\dots,\{p-k+1,\dots,p\}\big\}$. We choose the support of the parameter w_0 of the model to be $\{1,\dots,10\}$. Hence, three overlapping chains are needed to retrieve the support of w_0. We generate $n = 300$ examples $(y_i)_{i=1,..,n}$ from $y = x^\top w + \varepsilon$ where x is a standard Gaussian vector and $\varepsilon \sim \mathcal{N}(0, \sigma^2 I_p)$. The noise level is chosen to be $\sigma = 0.1$. In upper Figure 3 we show a time comparison of our algorithm on the regularized problem. We implemented Algorithm 1 and three Frank-Wolfe versions: simple FW, FW with line search (FW-ls) and pairwise FW (FW-pw). We compare also with a regularized version of the *forward-backward greedy* algorithm from Rao et al. (2015)(CoGEnT). In the bottom plot of Figure 3 we show a comparison on the constrained problem. All codes are in MATLAB and we used Rao et al.'s code for the *forward-backward greedy* algorithm.

Figure 4 illustrates complexity and memory usage of our algorithm for the same experiment. Top plots show

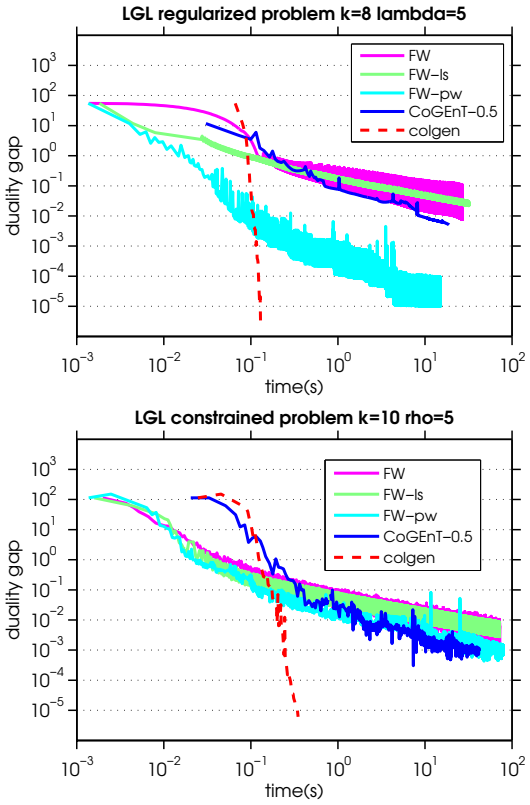


Figure 3: Experiments for k-chain group Lasso, where X is a generated random design matrix.*log-log* plot of progress of the duality gap during computation time. CoGEnT truncation parameter is set to $\eta = 0.5$.

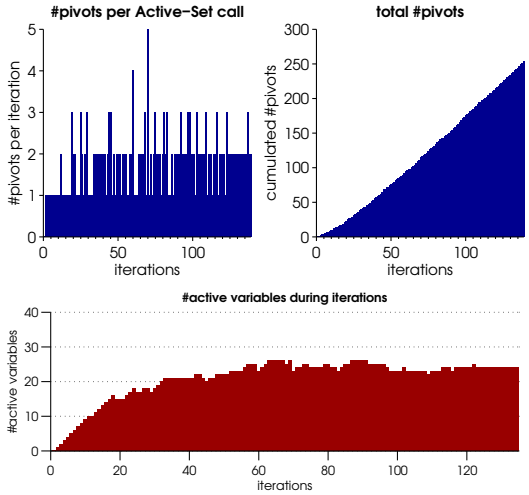


Figure 4: Experiment for the k-chain group Lasso. Top:number of pivots, i.e., drop/full step in active set. Left figure shows the number of pivots per active set call and right plot shows the total number of pivots during iterations. Bottom: evolution of the number of active atoms in our algorithm.

that each call to the active-set algorithm has low cost. Indeed less than two pivots in average, i.e. drop or full steps, are needed to converge. This is clearly due to the use of warm starts. Bottom plot shows the number of active atoms during iterations.

4.3 Hierarchical sparsity

In high-dimensional linear models that involve interaction terms, statisticians usually favor variable selection obeying certain logical hierarchical constraints. In this section we consider a quadratic model (linear + interaction terms) of the form

$$y = \sum_{i=1}^{p} \beta_i x_i + \sum_{i \neq j} \beta_{ij} x_i x_j .$$

Strong and weak hierarchical sparsity are usually distinguished (see Bien et al., 2013, and reference therein). The Weak Hierarchical (WH) sparsity constraints are that if an interaction is selected, then at least one of its associated main effects is selected, i.e., $\beta_{ij} \neq 0 \Rightarrow \beta_i \neq 0$ or $\beta_j \neq 0$. We use the latent overlapping group Lasso formulation proposed in Yan and Bien (2015) to formulate our problem. The corresponding collection of groups $\mathcal{B}$ thus contains the singletons $\{i\}$ and contains for all pairs $\{i, j\}$ the sets $\{i, \{i, j\}\}$ and $\{j, \{i, j\}\}$ (coupling respectively the selection of β_{ij} with that of β_i or that of β_i). We focussed on WH sparsity which is more challenging here because of the group overlaps, but the approach applies also to the counterpart for strong hierarchical constraints.

Simulated data We consider a quadratic problem with $p = 50$ main features, which entails that we have $p \times (p-1)/2 = 1225$ potential interaction terms and simulate $n = 1000$ samples. We choose the parameter β to have 10% of the interaction terms β_{ij} equal to 1 and the rest equal to zero. In order to respect the WH structure, the minimal number of necessary unary terms β_i possible given the WH constrains are included in the model with $\beta_i = 0.5$. We compare our algorithm with FCFW combined with an interior point solver (FCFW-ip) instead of the active-set subroutine, and with a degraded version of our algorithm not using warm starts. Figure 5 shows that FCFW-ip becomes slower than our algorithm only beyond 200 seconds. A plausible explanation is that at the begining the subproblems being solved are small and time is dominated by the search of the new direction; when the size of the problem grows, the active-set with warm start is faster, meaning that the active-set exploits the structure of positivity constraints better than IP, which has to invert bigger matrices. Full corrections of FCFW-ip call the `quadprog` function of MATLAB, which is an optimized C++ routine, whereas our implementation is done in MATLAB. An optimized C implementation of our active-set algorithm, in particular leveraging the

rank one updates on the inverse Hessian described in sections 3.3 should provide an additional significant speedup.

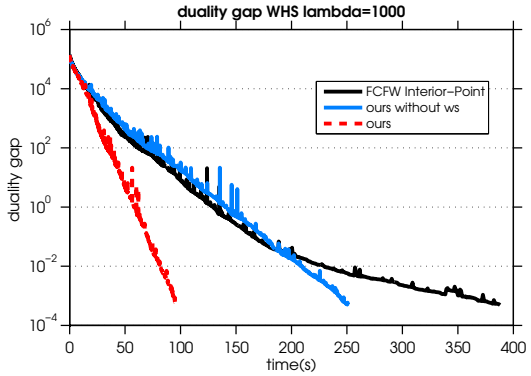


Figure 5: Experiments on simulated data for WH sparsity. *log*-plot of progress of the duality gap as a function time in seconds.

California housing data set We apply the previous hierarchical mode to the California housing data (Pace and Barry, 1997). The data contains 8 variables, so with interaction terms the intial model contains 36 variables. To make the selection problem more challenging, following She and Jiang (2014), we add 20 main nuisance variables, generated as standard Gaussian random variables corresponding to 370 additional noisy interaction terms. We compare our algorithm to the greedy Forward-Backward algorithm with a truncation parameter $\eta = 0.5$ and with Block Coordinate Descent (BCD). Table 1 shows running time for different levels of regularization λ. $\lambda = 10^{-3}$ is the value selected by 10-fold cross validation on the validation risk. Figure 6 shows the running time for the different algorithms.

Table 1: Computation time in seconds needed to reach a duality gap of 10^{-3} on California housing data set. Time is not reported when larger than 10^3 seconds.

λ	10^{-5}	10^{-4}	10^{-3}	10^{-2}	10^{-1}
BCD	-	-	585	73	5
CoGEnT	-	-	1300	14	0.2
ours	27	1.4	0.4	0.06	0.02

4.4 Sparse PCA

We compare our method to the block proximal gradient descent (BCD) described in Richard et al. (2014). We generate a sparse covariance matrix $\Sigma^\star$ of size 150×150 obtained as the sum of five overlapping rank one blocks $\mathbf{1}\mathbf{1}^\top$ of size $k \times k$ with $k = 10$. We generate a noisy covariance with a noise level $\sigma = 0.3$. We consider an ℓ_2 loss and a regularization by the gauge $\gamma_{\mathcal{A}_{k,\geq}}$ described in Section 2 with $k = 10$. The regularization parameter is λ. Figure 7 shows a time comparison with BCD.

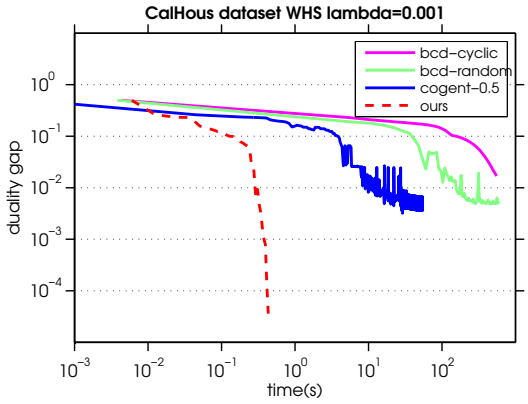


Figure 6: Experiments on California House data set. *log-log* plot of progress of the duality gap during computation time.

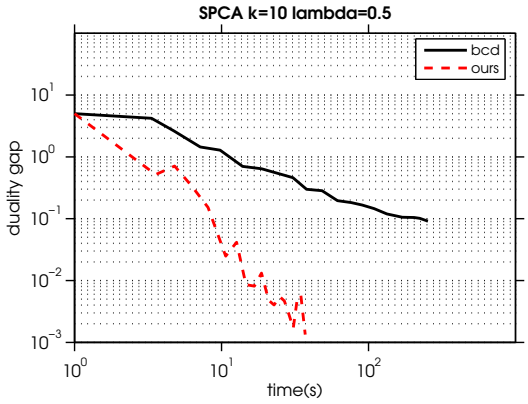


Figure 7: Experiments on sparse PCA. *log-log* plot of progress of the duality gap.

5 CONCLUSION

In this paper, we have shown that to minimize a quadratic function with an *atomic norm* regularization or constraint, the fully corrective Frank-Wolfe algorithm, which in the regularized case corresponds exactly to a very simple column generating algorithm that is not well known, is particularly efficient given that sparsity make the computation of the reduced Hessian relatively cheap. In particular, the corrective step is solved very efficiently with a simple active-set methods for quadratic programming. The proposed algorithm takes advantage of warm-starts, and empirically outperforms other Frank-Wolfe schemes, block-coordinate descent (when applicable) and the algorithm of Rao et al. (2015). Its performance could be enhanced by low-rank updates of the inverse Hessian. In future work we intend to generalize the algorithm to smooth loss functions using sequential quadratic programming.

Acknowledgements

Marina Vinyes is funded by ANR CHORUS research grant 13-MONU-0005-10.

References

Agarwal, A., Negahban, S., Wainwright, M. J., et al. (2012). Noisy matrix decomposition via convex relaxation: Optimal rates in high dimensions. *The Annals of Statistics*, 40(2):1171–1197.

Argyriou, A., Signoretto, M., and Suykens, J. A. (2014). Hybrid conditional gradient-smoothing algorithms with applications to sparse and low rank regularization. In *Regularization, Optimization, Kernels, and Support Vector Machines*, pages 53–82. Chapman and Hall/CRC.

Bach, F. (2013). Learning with submodular functions: A convex optimization perspective. *Foundations and Trends in Machine Learning*, 6(2):145–373.

Bach, F. (2015). Duality between subgradient and conditional gradient methods. *SIAM Journal on Optimization*, 25(1):115–129.

Bach, F., Mairal, J., and Ponce, J. (2008). Convex sparse matrix factorizations. *arXiv preprint arXiv:0812.1869*.

Bien, J., Taylor, J., Tibshirani, R., et al. (2013). A lasso for hierarchical interactions. *The Annals of Statistics*, 41(3):1111–1141.

Bredies, K., Lorenz, D. A., and Maass, P. (2009). A generalized conditional gradient method and its connection to an iterative shrinkage method. *Computational Optimization and Applications*, 42(2):173–193.

Candès, E. J., Li, X., Ma, Y., and Wright, J. (2011). Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11.

Chandrasekaran, V., Recht, B., Parrilo, P. A., and Willsky, A. S. (2012). The convex geometry of linear inverse problems. *Foundations of Computational mathematics*, 12(6):805–849.

Chandrasekaran, V., Sanghavi, S., Parrilo, P. A., and Willsky, A. S. (2011). Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2):572–596.

Ding, C., Li, T., and Jordan, M. I. (2010). Convex and semi-nonnegative matrix factorizations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(1):45–55.

Forsgren, A., Gill, P. E., and Wong, E. (2015). Primal and dual active-set methods for convex quadratic programming. *Mathematical Programming*, pages 1–40.

Foygel, R., Srebro, N., and Salakhutdinov, R. R. (2012). Matrix reconstruction with the local max norm. In *Advances in Neural Information Processing Systems*, pages 935–943.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1.

Gu, Q., Wang, Z., and Liu, H. (2016). Low-rank and sparse structure pursuit via alternating minimization. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pages 600–609.

Harchaoui, Z., Juditsky, A., and Nemirovski, A. (2015). Conditional gradient algorithms for norm-regularized smooth convex optimization. *Mathematical Programming*, 152(1–2):75–112.

Jacob, L., Obozinski, G., and Vert, J.-P. (2009). Group lasso with overlap and graph lasso. In *ICML*.

Lacoste-Julien, S. and Jaggi, M. (2015). On the global linear convergence of Frank-Wolfe optimization variants. *Advances in Neural Information Processing Systems 28*, pages 496–504.

Liu, J., Musialski, P., Wonka, P., and Ye, J. (2013). Tensor completion for estimating missing values in visual data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(1):208–220.

Maurer, A. and Pontil, M. (2012). Structured sparsity and generalization. *The Journal of Machine Learning Research*, 13(1):671–690.

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

Nocedal, J. and Wright, S. (2006). *Numerical optimization*. Springer Science & Business Media.

Obozinski, G. and Bach, F. (2016). A unified perspective on convex structured sparsity: Hierarchical, symmetric, submodular norms and beyond. *HAL Id : hal-01412385, version 1*.

Obozinski, G., Jacob, L., and Vert, J.-P. (2011). Group Lasso with overlaps: the Latent Group Lasso approach. *preprint HAL - inria-00628498*.

Pace, R. K. and Barry, R. (1997). Sparse spatial autoregressions. *Statistics and Probability Letters*, 33(3):291 – 297.

Qin, Z., Scheinberg, K., and Goldfarb, D. (2013). Efficient block-coordinate descent algorithms for the group lasso. *Mathematical Programming Computation*, 5(2):143–169.

Rao, N., Shah, P., and Wright, S. (2015). Forward -Backward Greedy Algorithms for Atomic Norm Regularization. *IEEE Transactions on Signal Processing*, 63(21):5798–5811.

Richard, E., Obozinski, G. R., and Vert, J.-P. (2014). Tight convex relaxations for sparse matrix factorization. In *Advances in Neural Information Processing Systems*, pages 3284–3292.

Rockafellar, R. T. (1997). *Convex Analysis. Princeton landmarks in mathematics*. Princeton University Press, Princeton, NJ.

Shalev-Shwartz, S. and Tewari, A. (2011). Stochastic methods for l1-regularized loss minimization. *Journal of Machine Learning Research*, 12(Jun):1865–1892.

She, Y. and Jiang, H. (2014). Group regularized estimation under structural hierarchy. *Journal of the American Statistical Association*.

Tomioka, R. and Suzuki, T. (2013). Convex tensor decomposition via structured Schatten norm regularization. In *Advances in Neural information Processing Systems*, pages 1331–1339.

Wimalawarne, K., Sugiyama, M., and Tomioka, R. (2014). Multitask learning meets tensor factorization: task imputation via convex optimization. In *Advances in Neural Information Processing Systems 27*, pages 2825–2833.

Wolfe, P. (1976). Finding the nearest point in a polytope. *Mathematical Programming*, 11(1):128–149.

Yan, X. and Bien, J. (2015). Hierarchical sparse modeling: A choice of two regularizers. *arXiv preprint arXiv:1512.01631*.

Yu, Y., Zhang, X., and Schuurmans, D. (2014). Generalized conditional gradient for sparse estimation. *arXiv preprint arXiv:1410.4828*.

Zhang, X., Schuurmans, D., and liang Yu, Y. (2012). Accelerated training for matrix-norm regularization: A boosting approach. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 2906–2914. Curran Associates, Inc.