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 regularized by an *atomic norm* or an *atomic gauge*. We propose to solve difficult problems in this family with a *column generation algorithm* (Larsson et al., 2015), which leads to a sequence of quadratic programs with only positivity constraints that can be solved efficiently with *active set methods for quadratic programming* (Nocedal and Wright, 2006; Goldfarb and Idnani, 1983).

1 Introduction

A number of problems in machine learning and structured optimization involve either structured constraint sets that are defined as the intersection of a number of simple sets or dually, gauges 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 norm 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 atomic 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.

Given a quadratic function f and an atomic norm used as a regularizer $\gamma_{\mathcal{A}}$, we consider in this paper optimization problems of the form

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

Our main contributions are a simple reformulation of the form taken by the Fully Corrective Frank-Wolfe (FCFW) algorithm for the regularized case and the proposal to solve the reduced problem in FCFW with a dedicated active-set algorithm for quadratic programming.

2 Gauges

Given a collection of atoms $\mathcal{A}$, an atomic gauge $\gamma_{\mathcal{A}}$ is the gauge of the set $C_{\mathcal{A}}$ defined as the convex hull of $\mathcal{A} \cup \{0\}$. It can be shown that (in a finite dimensional space) $\gamma_{\mathcal{A}}(x) = \inf\{\sum_{a\in\mathcal{A}} c_a \mid c \in \mathbb{R}^+, \sum_{a\in A} c_a a = x\}$. The polar gauge has the simpler expression $\gamma^\circ_{\mathcal{A}}(s) = \sup_{a\in\mathcal{A}}\langle s, a\rangle$.

For a number of atomic gauges, we have $\mathcal{A} = \bigcup_{j=1}^J C_j$ where C_j are convex sets. Then, $\gamma^\circ_{\mathcal{A}}(s) = \max_j \gamma^\circ_{C_j}$ and $\gamma_{\mathcal{A}} = \gamma_{C_1} \,\square\, \ldots \,\square\, \gamma_{C_J}$ where $\square$ denotes the infimal convolution[1] with $f \,\square\, g(x) = \inf_y f(x-y) + g(y)$. We thus have $\gamma_{\mathcal{A}}(x) = \inf\{\gamma_{C_1}(z_1) + \ldots + \gamma_{C_J}(z_J) \mid z_1 + \ldots + z_J = x\}$.

A natural example of a gauge defined as an infimal convolution is the Latent Group Lasso (LGL) norm (LGL) introduced in Jacob et al. (2009); Obozinski et al. (2011). Given a collection of sets $\mathcal{B}$,

[1] The infimal convolution is clearly commutative and associative.

29th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

the polar LGL norm is defined as $\Omega^\circ_{\mathrm{LGL}}(s) = \max_{B\in\mathcal{B}} \delta_B^{-1}\|s_B\|_2$ where s_B denotes the subvector of s whose entries are indexed by a set B and $\delta_B \in \mathbb{R}_+^*$. Defining $\mathcal{A}_B := \{x \mid x_{B^c} = 0,\ \delta_B\|x\|_2 = 1\}$, we have that $\Omega_{\mathrm{LGL}} = \gamma_{\mathcal{A}}$ is the infimal convolution of all the gauges $(\gamma_{C_B})_{B\subset\mathcal{B}}$, with $\mathcal{A} = \bigcup_B \mathcal{A}_B$ and C_B the convex hull of $\mathcal{A}_B$. In particular, we have $\{s \mid \Omega^\circ_{\mathrm{LGL}}(s) \le 1\} = \bigcap_B C_B^\circ$ with C_B° the cylinder $C_B^\circ = \left\{s \mid \|s_B\|_2 \le \delta_B\right\}$.

3 Column generation algorithm

For many of these norms, it is possible to compute $\operatorname{argmax}_{a\in\mathcal{A}}\langle a, s\rangle$. This has motivated a number of authors to suggest variants of the *conditional gradient algorithm*, also known as the Frank-Wolfe (FW). Main variants of *conditional gradient* are presented in Lacoste-Julien and Jaggi (2015). In Rao et al. (2015), a FCFW version with backward steps is applied to the constrained problem. Our problem can easily be reformulated as a truncated cone constrained problem as suggested by Harchaoui et al. (2015),

$$\min_{x,\tau} f(x) + \lambda\tau \quad \text{s.t.} \quad \gamma_{\mathcal{A}}(x) \le \tau, \quad \tau \le \rho, \qquad (1)$$

where one variable τ is added and a truncation level ρ that can be specified a priori as an upper bound on $\gamma_{\mathcal{A}}(x^\star)$ for $x^\star$ a solution of the problem.

Algorithm 1 Column generation

Require: f convex differentiable, ϵ
Initialisation: $x^0 = 0$, $A^0 = \varnothing$,
$k_0 = 0$, $t = 1$
repeat
  $a_t \leftarrow \arg\max_{a\in\mathcal{A}}\langle -\nabla f(x^{t-1}), a\rangle$
  $A^t \leftarrow [A^{t-1}, a_t]$
  $c^t \leftarrow \arg\min_{c\ge 0} f\left(A^t c\right) + \|c\|_1$
  $I \leftarrow \{i \mid c_i^t > 0\}$,
  $A^t \leftarrow A^t_{\cdot,I}$
  $x^t \leftarrow A^t c^t$
  $t \leftarrow t + 1$
until $\max_{a\in\mathcal{A}}\langle -\nabla f(x^{t-1}), a\rangle \le \epsilon$

The form of different Frank-Wolfe variant actually do not depend on the value ρ provided it is sufficiently large. Moreover, it is possible to show that applying FCFW on the truncated cone formulation above is equivalent to the simple column generation algorithm[2] presented in Algorithm 1. At each iteration, a new atom is added and we solve the original problem on the subset of atoms being considered. Since $\gamma_{\mathcal{A}^t}(x) = \inf\left\{\|c\|_1 \mid c \in \mathbb{R}_+^t,\ x = \sum_{i=1}^t c_i a_i\right\}$, the subproblem considered at the t-th iteration is

$$\min_{c\in\mathbb{R}_+^t} f\left(A^t c\right) + \lambda\|c\|_1, \qquad \text{with} \quad A^t := [a_1, \ldots, a_t] \in \mathbb{R}^{p\times t}. \qquad (2)$$

If f is quadratic, the problem is a Lasso problem with positivity constraints, which can efficiently be solved by a number of algorithms.

4 Active-set algorithm for quadratic programming

We propose to use *active set algorithms for convex quadratic programming* (Nocedal and Wright, 2006; Forsgren et al., 2015; Goldfarb and Idnani, 1983) . In particular, following[3] Bach (2013, Chap. 7.12), we propose to apply the active-set algorithm of Nocedal and Wright (2006, Chap. 16.5) to solve iteratively (2) . This algorithm takes the very simple[4] 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. In our active-set algorithm the iterates always remain dual feasible. Either the new iterate is primal-dual feasible — in which case we perform a *full-step* and potentially a new violated constraint is subsequently added— or not — and other constraints are added until we obtain an iterate that is primal-dual feasible for the subproblem. The solution is obtained when all constraints are satisfied.

For sparse problems, if the iterates remain in a low dimensional space, the theorem of Carathéodory guarantees that the simplex, i.e. the number of active atoms for us, remains small. This explains the

[2] In Harchaoui et al. (2015) a similar algorithm is proposed but without that a clear equivalence with FCFW on the truncated cone is stated.

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

[4] Despite the fact that, in the context of a simplicial algorithms, the polyhedral constraints sets of (2) 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.

efficiency of the algorithm, since the matrices to invert remain small. In our experiments, thanks to warm-starts, a very small number of pivots (full steps or drop steps of algorithm 2) are necessary in the active-set method, usually one or two pivots.

In order to apply the active-set algorithm, the subproblem (2) is explicitly rewritten as a quadratic problem with objective value$c^\top H^t c + b^t c$, where the Hessian H^t and b^t are updated each time a new atom is added. In the active-set, every time a constraint is added or removed, we have to invert a new Hessian to solve the subproblem. Fortunately, these operations translate into rank one updates on the Hessian and its inverse.

5 Experiments

In this section, we report experimental results to illustrate the computational efficiency of the proposed algorithm. We consider linear regression problems of the form $\min_x \frac{1}{2}\|Xw - y\|^2 + \lambda\gamma_{\mathcal{A}}(w)$ where X is a design matrix and $\gamma_{\mathcal{A}}$ the LGL. We also considered the constrained version for LGL. We compare our algorithm with the variants of Frank-Wolfe and with *forward-backward greedy* algorithm (COGEnT) of Rao et al. (2015) on simulated data. We also provide a comparison with FCFW with an interior-point solver on larger scale simulated data. Unreported comparisons[5] with the block coordinate descent algorithms of Jacob et al. (2009) on the similar data show that it is quite slow.

Algorithm 2 $[c, J]$=AS(H, b, c_0, J_0)

Solves: $P := \min_{c\geq 0} c^\top Hc + b^\top c$
Initialisation: $c = c_0$, $J = J_0$,
repeat
  $d \leftarrow {H_{J,J}}^{-1} b_J$
  if $d \geq 0$ **then**
    $c \leftarrow d$ ▷ full-step
    $g \leftarrow Hc + b$
    $k \leftarrow \arg\min_{i\in J_0\setminus J}\ g_i$
    if $g_k \geq 0$ **then**
      break
    else
      $J \leftarrow J \bigcup \{k\}$
    end if
  else
    $K \leftarrow \{i \mid c_i - d_i > 0,\ d_i < 0\}$
    $i^* \leftarrow \arg\min_{i\in K} \frac{c_i}{c_i - d_i}$
    $\tau \leftarrow \frac{c_{i^*}}{c_{i^*} - d_{i^*}}$
    $J \leftarrow J \setminus \{i\}$ ▷ drop-step
    $c \leftarrow c + \tau(d - c)$
  end if
until $g_{J_0\setminus J} \geq 0$
return c, J

k-chain Lasso We consider an LGL 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, two or three overlapping groups are typically non-zero at the solution for appropriate regularization levels. We generate $n = 300$ examples $(y_i)_{i=1,..,n}$ from $y = x^\top w + \varepsilon$ where x is a standard Gaussian vector and $\varepsilon \sim \mathcal{N}(0, \sigma I_p)$. In the left plot of Figure 1 we show a time comparison on the regularized problem. We implemented Algorithm 1 and three Frank-Wolfe versions along the lines of Harchaoui et al. (2015): simple FW, FW with line search and pairwise FW. We compare also with a regularized version of the *forward-backward greedy* algorithm of Rao et al. (2015). In the right plot of Figure 1 we show a comparison on the constrained problem. We coded constrained versions of the aforementioned methods, except for *forward-backward greedy* algorithm for which the code was available. We clearly outperform all contending methods.

Weak hierarchical sparsity In high-dimensional linear models that involve interaction terms, statisticians usually favor variable selection obeying certain logical hierarchical constraints. The Weak Hierarchical (WH) sparsity constraints (see Bien et al., 2013, and reference therein) are that if an interaction is selected, then at least one of its associated main effects should be selected. We use the latent overlapping group Lasso formulation proposed in Yan and Bien (2015) to obtain a convex formulation inducing WH sparsity.

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 consider a quadratic problem with $p = 50$ main features, which entails that we have $p \times (p-1)/2 = 1225$ potential interaction terms. We choose the parameter β to have 10% of the interaction terms β_{ij} equal to 1 and the rest equal to zero. We simulated a sample of size $n = 1000$.

We compare our algorithm with FCFW combined with an interior point solver (FCFW-ip) instead of the active-set subroutine. We also show that our method takes advantage of warm starts. See

[5] We did not add the plot of BCD because the code of Jacob et al. (2009) is written for a model with intercept term.

Figure 5 for a time comparison with FCFW-ip and our method without warm starts. Note that we use the IP solved implemented under the function `quadprog` in MATLAB, which is an optimized C++ routine, whereas the implementation of our active set algorithm is done in plain MATLAB. Clearly, an optimized C implementation of our active-set algorithm would provide an additional significant speedup. To give a better idea of the improvement brought over interior points methods, Figure 5 shows the number of matrix inversions per size of the matrix. The interior point solver requires 6-7 times more matrix inversions than the active-set algorithm we propose to use for most of the iterations of the algorithm.

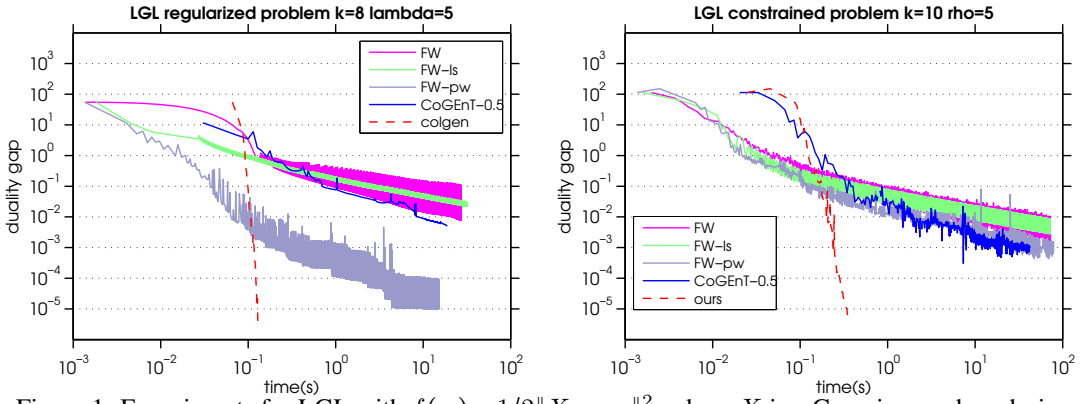

Figure 1: Experiments for LGL with $f(w) = 1/2\|Xw - y\|^2$, where X is a Gaussian random design matrix. The noise level is chosen to be $\sigma = 0.1$: *log-log* plot of the duality gap as a function of computation time. (Left) Regularized problem with $\lambda = 5$; (right) Constrained problem with $\rho = 5$. **FW**: Frank-Wolfe, **FWls**: line search FW, **FWpw**: pairwise FW, **CoGEnT**: greedy forward-backward algorithm with truncation parameter $\eta = 0.5$, **colgen**: our algorithm with warm-starts.

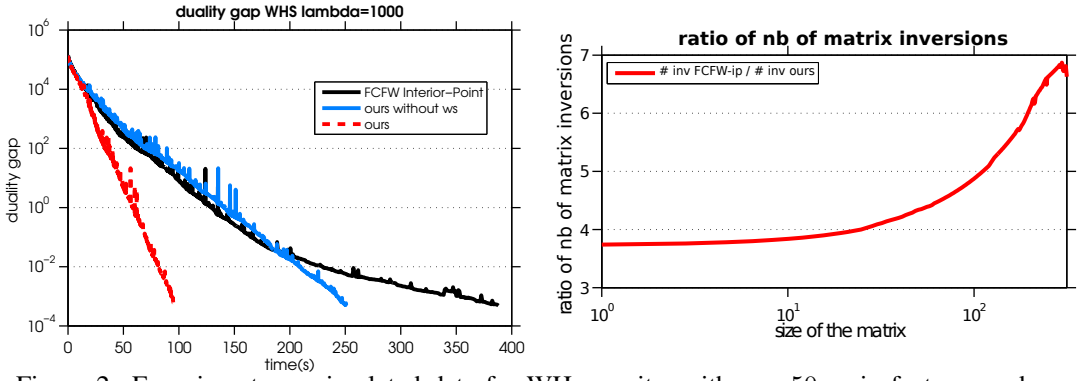

Figure 2: Experiments on simulated data for WH sparsity, with $p = 50$ main features and $p \times (p-1)/2 = 1225$ possible interactions among which 10% are true interactions. (Left) *log*-plot of the decrease of the duality gap as a function time in seconds for FCFW-ip, our algorithm with and without warm-starts (ws) in the active-set algorithm. (Right) Number of matrix inversions for FCFW-ip divided by the number of inversions in our method per size of the matrix.

6 Conclusion

In this paper, we have shown that to minimize quadratic function with an *atomic gauge* regularization or constraint, the FCFW algorithm, which corresponds exactly to a very simple column generating algorithm in the regularized case that is not well known, is particularly efficient given that sparsity make the computation of reduced Hessian relatively cheap. In particular, we showed that 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 and the algorithm of Rao et al. (2015). The performance of the algorithm could be further enhanced by low-rank updates of the inverse Hessian.

References

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

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

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.

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.

Goldfarb, D. and Idnani, A. (1983). A numerically stable dual method for solving strictly convex quadratic programs. *Mathematical Programming*, 27(1):1–33.

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. pages 496–504.

Larsson, T., Migdalas, A., and Patriksson, M. (2015). A generic column generation principle: derivation and convergence analysis. *Operational Research*, 15(2):163–198.

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.

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

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

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.

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*.