Méthodes de descente par coordonnée pour les problèmes d'optimisation convexes non-dérivables

Coordinate descent methods for non-differentiable convex optimisation problems

Mémoire d'Habilitation à Diriger des Recherches

présenté par

Olivier Fercoq

Contents

Chapter 1

Introduction

1.1 Historical motivation for coordinate descent

The idea of coordinate descent is to decompose a large optimisation problem into a sequence of one-dimensional optimisation problems. The algorithm was first described for the minimization of quadratic functions by Gauss and Seidel in [Sei74]. Coordinate descent methods have become unavoidable in machine learning because they are very efficient for key problems, namely Lasso, logistic regression and support vector machines. Moreover, the decomposition into small subproblems means that only a small part of the data is processed at each iteration and this makes coordinate descent easily scalable to high dimensions.

We first decompose the space of optimisation variables X into blocks $X_1 \times \ldots \times X_n = X$. A classical choice when $X = \mathbb{R}^n$ is to choose $X_1 = \ldots = X_n = \mathbb{R}$. We will denote U_i the canonical injection from X_i to X, that is U_i is such that for all $h \in X_i$,

$$U_i h = (\underbrace{0, \ldots, 0}_{i-1 \text{ zeros}}, h^\top, \underbrace{0, \ldots, 0}_{n-i \text{ zeros}})^\top \in X.$$

For a function $f : X_1 \times \ldots \times X_n \to \mathbb{R}$, we define the following algorithm.

Algorithm 1 Exact coordinate descent

Start at $x_0 \in X$.
At iteration k, choose $l = (k \mod n) + 1$ (cyclic rule) and define $x_{k+1} \in X$ by

$$\begin{cases} x_{k+1}^{(i)} = \arg\min_{z \in X_l} f(x_k^{(1)}, \ldots, x_k^{(l-1)}, z, x_k^{(l+1)}, \ldots, x_k^{(n)}) & \text{if } i = l \\ x_{k+1}^{(i)} = x_k^{(i)} & \text{if } i \neq l \end{cases}$$

Proposition 1 ([War63]). *If f is continuously differentiable and strictly convex and there exists $x_* = \arg\min_{x \in X} f(x)$, then the exact coordinate descent method (Alg. 1) converges to x_*.*

Example 1 (least squares). $f(x) = \frac{1}{2}\|Ax - b\|_2^2 = \frac{1}{2}\sum_{j=1}^m (a_j^\top x - b_j)^2$

At each iteration, we need to solve in z the 1D equation

$$\frac{\partial f}{\partial x^{(l)}}(x_k^{(1)}, \ldots, x_k^{(l-1)}, z, x_k^{(l+1)}, \ldots, x_k^{(n)}) = 0$$

For all $x \in \mathbb{R}^n$,

$$\frac{\partial f}{\partial x^{(l)}}(x) = a_l^\top (Ax - b) = a_l^\top a_l x^{(l)} + a_l^\top (\sum_{j \neq l} a_j x^{(j)}) - a_l^\top b$$

so we get

$$z^* = x_{k+1}^{(l)} = \frac{1}{\|a_l\|_2^2}\Big(- a_l^\top (\sum_{j \neq l} a_j x_k^{(j)}) + a_l^\top b\Big) = x_k^{(l)} - \frac{1}{\|a_l\|_2^2}\Big(a_l^\top (\sum_{j=1}^n a_j x_k^{(j)}) - a_l^\top b\Big)$$

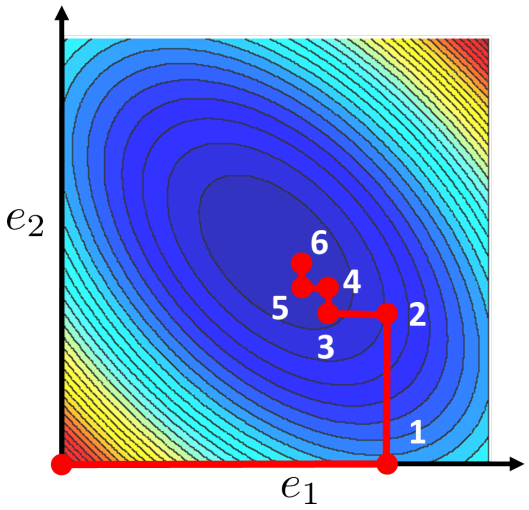

Figure 1.1: The successive iterates of the coordinate descent method on a 2D example. The function we are minimising is represented by its level sets: the bluer is the circle, the lower is the function values.

Example 2 (Adaboost)**.** *The Adaboost algorithm [CSS02] was designed to minimise the exponential loss given by*

$$f(x) = \sum_{j=1}^{m} \exp(-y_j h_j^\top x).$$

At each iteration, we select the variable l such that $l = \arg\max_i As\nabla_i f(x)$ and we perform an exact coordinate descent step along this coordinate.

This variable selection rule is called the greedy or Gauss-Southwell rule. Like the cyclic rule, it leads to a converging algorithm but requires to compute the full gradient at each iteration. Greedy coordinate descent is interesting in the case of the exponential loss because the gradient of the function has a few very large coefficients and many negligible coefficients.

1.2 Why is coordinate descent useful?

Solving a one-dimensional optimisation problems is generally easy and the solution can be approximated very well by algorithms like the bisection method. However, for the exact coordinate descent method, one needs to solve a huge number of one-dimensional problems and the expense quickly becomes prohibitive. Moreover, why should we solve to high accuracy the 1-dimensional problem and destroy this solution at the next iteration?

The idea of coordinate gradient descent is to perform one iteration of gradient descent in the 1-dimensional problem $\min_{z\in X_l} f(x_k^{(1)}, \dots, x_k^{(l-1)}, z, x_k^{(l+1)}, \dots, x_k^{(n)})$ instead of solving it completely. In general, this reduces drastically the cost of each iteration while keeping the same convergence behaviour.

Algorithm 2 Coordinate gradient descent

Start at x_0.
At iteration k, choose $i_{k+1} \in \{1, \dots, n\}$ and define x_{k+1} by

$$\begin{cases} x_{k+1}^{(i)} = x_k^{(i)} - \gamma_i \nabla_i f(x_k) & \text{if } i = i_{k+1} \\ x_{k+1}^{(i)} = x_k^{(i)} & \text{if } i \neq i_{k+1} \end{cases}$$

When choosing the cyclic or greedy rule, the algorithm does converge for any convex function f that has a Lipschitz-continuous gradient and such that $\arg\min_x f(x) \neq \emptyset$.

In fact we will assume that we actually know the *coordinate-wise* Lipschitz constants of the gradient of f, namely the Lipschitz constants of the functions

$$\begin{aligned} g_{i,x} :\ & X_i \to \mathbb{R} \\ & h \mapsto f(x + U_i h) = f(x^{(1)}, \dots, x^{(i-1)}, x^{(i)} + h, x^{(i+1)}, \dots, x^{(n)}) \end{aligned} \tag{1.1}$$

We will denote $L_i = \sup_x L(\nabla g_{i,x})$ this Lipschitz constant. Written in terms of f, this means that

$$\forall x \in X, \forall i \in \{1, \dots, n\}, \forall h \in X_i, \quad \|\nabla f(x + U_i h) - \nabla f(x)\|_2 \leq L_i \|U_i h\|_2.$$

Lemma 1. *If f has a coordinate-wise Lipschitz gradient with constants $L_1, \dots, L_n$, then $\forall x \in X$, $\forall i \in \{1, \dots, n\}, \forall h \in X_i$,*

$$f(x + U_i h) \leq f(x) + \langle \nabla_i f(x), h \rangle + \frac{L_i}{2} \|h\|^2$$

Proposition 2 ([BT13])**.** *Assume that f is convex, ∇f is Lipschitz continuous and $\arg\min_{x \in X} f(x) \neq \emptyset$. If i_{k+1} is chosen with the cyclic rule $i_{k+1} = (k \mod n) + 1$ and $\forall i$, $\gamma_i = \frac{1}{L_i}$, then the coordinate gradient descent method (Alg. 2) satisfies*

$$f(x_{k+1}) - f(x_*) \leq 4 L_{\max} (1 + n^3 L_{\max}^2 / L_{\min}^2) \frac{R^2(x_0)}{k + 8/n}$$

where $R^2(x_0) = \max_{x,y \in X} \{\|x - y\| : f(y) \leq f(x) \leq f(x_0)\}$, $L_{\max} = \max_i L_i$ and $L_{\min} = \min_i L_i$.

The proof of this result is quite technical and in fact the bound is much more pessimistic than what is observed in practice (n^3 is very large if n is large). This is due to the fact that the cyclic rule behaves particularly bad on some extreme examples. To avoid such traps, it has been suggested to randomise the coordinate selection process.

Proposition 3 ([Nes12a])**.** *Assume that f is convex, ∇f is Lipschitz continuous and $\arg\min_{x \in X} f(x) \neq \emptyset$. If i_{k+1} is randomly generated, independently of $i_1, \dots, i_k$ and $\forall i \in \{1, \dots, n\}$, $\mathbb{P}(i_{k+1} = i) = \frac{1}{n}$ and $\gamma_i = \frac{1}{L_i}$, then the coordinate gradient descent method (Alg. 2) satisfies for all $x_* \in \arg\min_x f(x)$*

$$\mathbb{E}[f(x_{k+1}) - f(x_*)] \leq \frac{n}{k+n} \Big((1 - \frac{1}{n})(f(x_0) - f(x_*)) + \frac{1}{2} \|x_* - x_0\|_L^2 \Big)$$

where $\|x\|_L^2 = \sum_{i=1}^n L_i \|x^{(i)}\|_2^2$.

Comparison with gradient descent The iteration complexity of the gradient descent method is

$$f(x_{k+1}) - f(x_*) \leq \frac{L(\nabla f)}{2(k+1)} \|x_* - x_0\|_2^2$$

This means that to get an ϵ-solution (i.e. such that $f(x_k) - f(x_*) \leq \epsilon$), we need at most $\frac{L(\nabla f)}{2\epsilon} \|x_* - x_0\|_2^2$ iterations. What is most expensive in gradient descent is the evaluation of the gradient $\nabla f(x)$ with a cost C, so the total cost of the method is

$$C_{\text{grad}} = C \frac{L(\nabla f)}{2\epsilon} \|x_* - x_0\|_2^2$$

Neglecting the effect of randomisation, we usually have an ϵ-solution with coordinate descent in $\frac{n}{\epsilon} \big((1 - \frac{1}{n})(f(x_0) - f(x_*)) + \frac{1}{2} \|x_* - x_0\|_L^2 \big)$ iterations. The cost of one iteration of coordinate descent is of the order of the cost of evaluation one partial derivative $\nabla_i f(x)$, with a cost c, so the total cost of the method is

$$C_{\text{cd}} = c \frac{n}{\epsilon} \big((1 - \frac{1}{n})(f(x_0) - f(x_*)) + \frac{1}{2} \|x_* - x_0\|_L^2 \big)$$

How do these two quantities compare?
Let us consider the case where $f(x) = \frac{1}{2} \|Ax - b\|_2^2$.

- Computing $\nabla f(x) = A^\top (Ax - b)$ amounts to updating the residuals $r = Ax - b$ (one matrix vector product and a sum) and computing one matrix vector product. We thus have $C = O(\text{nnz}(A))$.

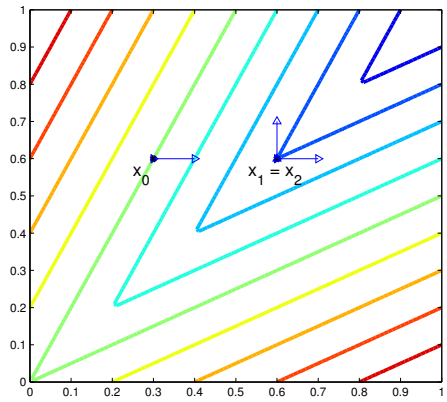

Figure 1.2: The function in Example 4

- Computing $\nabla_i f(x) = e_i^\top A^\top (Ax - b)$ amounts to
  1. updating the residuals $r = Ax - b$: one scalar-vector product and a sum since we have $r_{k+1} = r_k + (x_{k+1}^{(i_{k+1})} - x_k^{(i_{k+1})}) A e_{i_{k+1}}$,
  2. computing one vector-vector product (the i^{th} column of A versus the residuals).

  Thus $c = O(\text{nnz}(Ae_{i_{k+1}})) = O(\text{nnz}(A)/n) = C/n$ if the columns of A are equally sparse.
- $f(x_0) - f(x_*) \leq \frac{L(\nabla f)}{2} \|x_0 - x_*\|_2^2$ and it may happen that $f(x_0) - f(x_*) \ll \frac{L(\nabla f)}{2} \|x_0 - x_*\|_2^2$
- $L(\nabla f) = \lambda_{\max}(A^\top A)$ and $L_i = a_i^\top a_i$ with $a_i = Ae_i$. We always have $L_i \leq L(\nabla f)$ and it may happen that $L_i = O(L(\nabla f)/n)$.

To conclude, in the quadratic case, $C_{\text{cd}} \leq C_{\text{grad}}$ and we may have $C_{\text{cd}} = O(C_{\text{grad}}/n)$.

1.3 Two counter-examples

Example 3 (non-convex differentiable function)**.**
$f(x^{(1)}, x^{(2)}, x^{(3)}) = -(x^{(1)}x^{(2)} + x^{(2)}x^{(3)} + x^{(3)}x^{(1)}) + \sum_{i=1}^3 \max(0, |x^{(i)}| - 1)^2$

As shown by [Pow73], exact coordinate descent on this function started at the initial point $x^{(0)} = (-1-\epsilon, 1+\epsilon/2, -1-\epsilon/4)$ has a limit cycle around the 6 corners of the cube that are not minimisers and avoids the 2 corners that are minimisers.

This example shows that some care should be taken when applying coordinate descent to a non-convex function. Even with this restriction, block coordinate descent (also called alternating minimization) is often used when the objective function satisfies coordinate-wise convexity: when the partial function $g_{i,x}$ defined in (1.1) is convex, we can approximately solve the subproblems. This situation is for instance encountered for nonnegative matrix factorization [CZA08].

My work of the last years was focused on another challenge of coordinate descent methods: the treatment of convex but non-differentiable functions.

Example 4 (non-differentiable convex function [Aus76])**.** $f(x^{(1)}, x^{(2)}) = |x^{(1)} - x^{(2)}| - \min(x^{(1)}, x^{(2)}) + I_{[0,1]^2}(x)$ *where $I_{[0,1]^2}$ is the convex indicator of $[0,1]^2$. f is convex but not differentiable. If we nevertheless try to run exact coordinate descent, the algorithm proceeds as $x_1^{(1)} = \arg\min_z f(z, x_0^{(2)}) = x_0^{(2)}$, $x_2^{(2)} = \arg\min_z f(x_1^{(1)}, z) = x_0^{(2)}$, and so on. Thus exact coordinate descent converges in two iterations to $(x_0^{(2)}, x_0^{(2)})$: the algorithm is stuck on a non-differentiability point on the line $\{x^{(1)} = x^{(2)}\}$ and does not reach the minimiser $(1,1)$.*

We can see that for non-differentiable convex functions, exact coordinate descent may not return the expected solution. On this example, using the notation of (1.1), even though $0 \in \partial g_{i,x}(0)$, for all i, we

have $0 \notin \partial f(x)$. Said otherwise, even when 0 is not in the subdifferential of f at x, for all direction i there may exist a subgradient $q \in \partial f(x)$ such $q^{(i)} = 0$.

A classical workaround is to restrict the attention to composite problems involving the sum of a differentiable function and a *separable* nonsmooth functions [Tse01].

Definition 1. *A function f is said to be separable if it can be written as*

$$f(x) = \sum_{i=1}^{n} f_i(x^{(i)}) \, .$$

My contribution includes faster algorithms to deal with non-differentiable separable functions, smoothing techniques for non-differentiable non-separable functions and primal-dual algorithms. A substantial amount of my research has been driven by the wish to extend successful optimization techniques to coordinate descent. I also have a great interest in applications of coordinate descent that involve the resolution of optimization problems in large dimensions.

1.4 A negative result on universal coordinate descent

The universal gradient algorithm introduced by Nesterov [Nes13b] is an algorithm that is able to minimize smooth as well as nonsmooth convex functions without any a priori knowledge on the level of smoothness of the function. This a particularly desirable feature. First of all, the same algorithm may be used for a large class of problems and, having no parameter to tune, it is very robust. Secondly, the adaptive process that "discovers" the level of smoothness of the function may take profit of a locally favorable situation, even though the function is difficult to minimize at the global scope.

Our aim in the paper [FR14] is to design and analyze a *universal coordinate descent method* for the problem of minimizing a convex composite function:

$$\min_{x \in \mathbb{R}^N} \left[F(x) \equiv f(x) + \Psi(x) \right], \tag{1.2}$$

where Ψ is convex and has a simple proximal operator. The function $f(x)$ is convex and can be smooth or nonsmooth. It is still interesting to consider the composite framework because we can take profit of the proximal operator of Ψ.

Classical coordinate descent algorithm may get stuck at a non-stationary point if applied to a general nonsmooth convex problem. However, several coordinate-descent-type methods have been proposed for nonsmooth problems. An algorithm based on the averaging of past subgradient coordinates is presented in [TKCW12] and a successful subgradient-based coordinate descent method for problems with sparse subgradients is proposed by Nesterov [Nes12b]. An important feature of these algorithms is that at each point x, one subgradient $\nabla f(x)$ is selected and then the updates are performed according to the i^{th} coordinate $\nabla_i f(x)$ of the subgradient. This is different to partial subgradients. A coordinate descent algorithm based on smoothing was proposed in [FR17] for the minimization of nonsmooth functions with a max-structure. However if one tries to use one of these algorithms on a smooth problem, one would get a very slow algorithm with iteration complexity in $O(1/\epsilon^2)$.

The adaptive procedure of the universal gradient method is based on a line search, the parameter of which estimates either the Lipschitz constant of the function (if it is nonsmooth) or the Lipschitz constant of the gradient of the function (if it is smooth). We designed such a line search procedure for universal coordinate descent. On top of being able to deal with nonsmooth functions, it covers the composite framework with a nonsmooth regularizer and uses only partial derivatives evaluation.

We extend the theory of parallel coordinate descent developed in [RT15] to the universal coordinate descent method. In particular, we define a non-quadratic expected separable overapproximation for partially separable functions that allows us to run independent line searches on all the coordinates to be minimized at a given iteration. This is a result of independent interest as this is the first time that a line search procedure is introduced for parallel coordinate descent.

We design a universal accelerated coordinate descent method with optimal rates of convergence $O(1/\sqrt{\epsilon})$ for smooth problems and $O(1/\epsilon^2)$ for nonsmooth problems. We recover many previous results. In particular we recover the universal primal gradient method [Nes13b] when we consider one single block (in our notation, $n = 1$) and we recover the accelerated coordinate descent method [FR15]. Moreover, the line search procedure allows us not to bother about the coordinatewise Lipschitz constant.

Algorithm 3 Universal coordinate descent

Choose $(L_0^j)_{j\in[n]}$ and accuracy ϵ.
For $k \geq 0$ do:

1. Select a subgradient $\nabla f(x_k)$
2. Select block j_k at random.
3. Find the smallest $s_k \in \mathbb{N}$ such that for

$$x_k^+ = \arg\min_{z\in\mathbb{R}_j} \Psi_{j_k}(z) + f(x_k) + \langle \nabla_j f(x_k), z - x_k^{(j_k)}\rangle + \frac{1}{2}2^{s_k}L_k^{j_k}\|z - x_k^{(j_k)}\|_{(j_k)}^2,$$

we have $\frac{1}{2}\frac{1}{2^{s_k-1}L_k^{j_k}}\left(\|\nabla_{j_k} f(x_k) - \nabla_{j_k} f(x_k^+)\|_{(j_k)}^*\right)^2 \leq \frac{2^{s_k-1}L_k^{j_k}}{2}\|x_k^{(j_k)} - (x_k^+)^{(j_k)}\|_{(j_k)}^2 + \frac{\epsilon}{2n}$.

4. Set $x_{k+1} = x_k^+$ and $L_{k+1}^{j_k} = 2^{s_k}L_k^{j_k}$

However, as one can see on Figure 1.3, universal coordinate descent (Algorithm 3) is quite slow, much slower than the universal gradient method. This is consistent with the dependence in n of the iteration complexity of the algorithm which is of order n^2 when $\nu = 0$, as stated in Theorem 1 below.

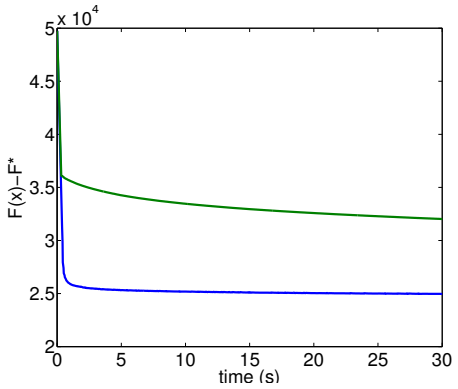

Figure 1.3: Universal coordinate descent (green) is much slower than universal gradient (blue) for the nonsmooth problem $\min_x \|Ax - b\|_1 + \|x\|_1$.

Theorem 1. *Let f be a convex function with Hölder continuous gradient, that is such that there exists $\nu \in [0,1]$ such that for all $x, y \in \mathbb{R}^n$, for all $\nabla f(x) \in \partial f(x)$ and for all $\nabla f(y) \in \partial f(y)$,*

$$\|\nabla f(y) - \nabla f(x)\|^* \leq M_\nu \|x - y\|^\nu.$$

Let us denote $\mathcal{R}_M(x_*) = \max_{x\in D}\|x - x_*\|_M$ (usual weighted 2-norm) where D is a set containing x_k for all k and $\bar{\Lambda} = \frac{1}{n}\sum_{j=1}^n \log_2(\frac{2M^j}{L_0^j})$. Algorithm 3 converges to an ϵ-solution with iteration complexity

$$\min_{1\leq l\leq k} \mathbf{E}[f(x_l) - f(x_*)] \leq \frac{1}{k}\sum_{l=1}^k \mathbf{E}[f(x_l) - f(x_*)] \leq \frac{n}{k}R_0^2(x_*) + \frac{2n^{\frac{2}{1+\nu}}}{k\epsilon^{\frac{1-\nu}{1+\nu}}}\bar{\Lambda}\mathcal{R}_{M_\nu^{\frac{2}{1+\nu}}}(x_*)^2 + \frac{\epsilon}{2}\,.$$

This study on universal coordinate descent showed that it is indeed possible to design a coordinate descent method based solely on subgradients but in order to really gain from the coordinate descent framework when minimizing a nonsmooth function, one should use more information on this function.

1.5 Plan of the thesis

This thesis summarizes my research work between 2012 and 2018 while I was working at the University of Edinburgh and at Télécom Paristech. My publications follow three main tracks:

Chapter 2. **Fast algorithms for composite differentiable-separable functions**

Coordinate descent methods have shown very efficient for problems of the form $\min_x f(x) + \sum_i \psi_i(x_i)$ with f differentiable. This chapter is based on the following papers.

[FQRT14] Olivier Fercoq, Zheng Qu, Peter Richtárik, and Martin Takáč. Fast distributed coordinate descent for minimizing non-strongly convex losses. In *IEEE International Workshop on Machine Learning for Signal Processing*, 2014.

[FR15] Olivier Fercoq and Peter Richtárik. Accelerated, parallel and proximal coordinate descent. *SIAM Journal on Optimization*, 25(4):1997–2023, 2015.

[FR16] Olivier Fercoq and Peter Richtárik. Optimization in high dimensions via accelerated, parallel, and proximal coordinate descent. *SIAM Review*, 58(4):739–771, 2016.

[QRTF16] Zheng Qu, Peter Richtárik, Martin Takác, and Olivier Fercoq. Sdna: stochastic dual newton ascent for empirical risk minimization. In *International Conference on Machine Learning*, pages 1823–1832, 2016.

[FQ16] Olivier Fercoq and Zheng Qu. Restarting accelerated gradient methods with a rough strong convexity estimate. *arXiv preprint arXiv:1609.07358*, 2016.

[FQ17] Olivier Fercoq and Zheng Qu. Adaptive restart of accelerated gradient methods under local quadratic growth condition. *arXiv preprint arXiv:1709.02300*, 2017.

[FQ18] Olivier Fercoq and Zheng Qu. Restarting the accelerated coordinate descent method with a rough strong convexity estimate. *arXiv preprint arXiv:1803.05771*, 2018.

Chapter 3. **Coordinate descent methods for saddle point problems**

In this line of work, I have been trying to show that coordinate descent methods can be applied successfully to a much wider class of functions that what was previously thought. I have works on primal-dual methods, smoothing theory and their relations to coordinate descent.

[FR13] Olivier Fercoq and Peter Richtárik. Smooth Minimization of Nonsmooth Functions by Parallel Coordinate Descent. *arXiv:1309.5885*, 2013.

[Fer13] Olivier Fercoq, Parallel coordinate descent for the Adaboost problem In *International Conference on Machine Learning and Applications*, 2013.

[FR14] Olivier Fercoq and Peter Richtárik. Universal coordinate descent and line search for parallel coordinate descent. Technical report, The University of Edinburgh, 2014.

[FB15] Olivier Fercoq and Pascal Bianchi. A Coordinate Descent Primal-Dual Algorithm with Large Step Size and Possibly Non Separable Functions. *arXiv preprint arXiv:1508.04625*, 2015. Accepted for publication in SIAM Journal on Optimization.

[BF16] Pascal Bianchi and Olivier Fercoq. Using Big Steps in Coordinate Descent Primal-Dual Algorithms. In *Proc. of the Conference on Decision and Control*, 2016.

[VNFC17] Quang Van Nguyen, Olivier Fercoq, and Volkan Cevher. Smoothing technique for nonsmooth composite minimization with linear operator. *arXiv preprint arXiv:1706.05837*, 2017.

[ADFC17] Ahmet Alacaoglu, Quoc Tran Dinh, Olivier Fercoq, and Volkan Cevher. Smooth primal-dual coordinate descent algorithms for nonsmooth convex optimization. In *Advances in Neural Information Processing Systems*, pages 5852–5861, 2017.

[TDFC18] Quoc Tran-Dinh, Olivier Fercoq, and Volkan Cevher. A smooth primal-dual optimization framework for nonsmooth composite convex minimization. *SIAM Journal on Optimization*, 28(1):96–134, 2018.

[YFLC18] Alp Yurtsever, Olivier Fercoq, Francesco Locatello, and Volkan Cevher. A conditional gradient framework for composite convex minimization with applications to semidefinite programming. In *ICML*, 2018.

Chapter 4. **Applications to statistics**

Statistical estimators are often defined as the solution of an optimization problem. Moreover, the structure of the functions encountered in statistics and the massive amount of data to deal with makes it a natural playground for coordinate descent methods. I have contributions in safe screening rules, a concomitant lasso solver, joint quantile regression and ϵ-insensitive losses.

[FGS15] Olivier Fercoq, Alexandre Gramfort, and Joseph Salmon. Mind the duality gap: safer rules for the lasso. In *ICML*, pages 333–342, 2015.

[NFGS15] Eugène Ndiaye, Olivier Fercoq, Alexandre Gramfort, and Joseph Salmon. GAP safe screening rules for sparse multi-task and multi-class models. *NIPS*, pages 811–819, 2015.

[NFGS16] Eugène Ndiaye, Olivier Fercoq, Alexandre Gramfort, and Joseph Salmon. GAP safe screening rules for Sparse-Group Lasso. *NIPS*, 2016.

[SFdB16] Maxime Sangnier, Olivier Fercoq, and Florence d'Alché Buc. Joint quantile regression in vector-valued rkhss. In *Advances in Neural Information Processing Systems*, pages 3693–3701, 2016.

[NFGS17] Eugene Ndiaye, Olivier Fercoq, Alexandre Gramfort, and Joseph Salmon. Gap safe screening rules for sparsity enforcing penalties. *Journal of Machine Learning Research*, 18(1):4671–4703, 2017.

[MFGS17] Mathurin Massias, Olivier Fercoq, Alexandre Gramfort, and Joseph Salmon. Heteroscedastic concomitant lasso for sparse multimodal electromagnetic brain imaging. In *Proc. of the 21st International Conference on Artificial Intelligence and Statistics*, 1050:27, 2017.

[SFdB17] Maxime Sangnier, Olivier Fercoq, and Florence d'Alché Buc. Data sparse nonparametric regression with ε-insensitive losses. In *Asian Conference on Machine Learning*, pages 192–207, 2017.

[NFT+18] Eugène Ndiaye, Olivier Fercoq, Joseph Salmon, Ichiro Takeuchi, and Le Tam. Safe grid search with optimal complexity. Technical report, Télécom ParisTech and Nagoya Institute of Technology, 2018.

1.6 Publications related to my PhD thesis

[BCF+12] Frédérique Billy, Jean Clairambault, Olivier Fercoq, Stéphane Gaubert, Thomas Lepoutre, Thomas Ouillon, and Shoko Saito. Synchronisation and control of proliferation in cycling cell population models with age structure. *Mathematics and Computers in Simulation*, 2012.

[Fer12a] Olivier Fercoq. Convergence of Tomlin's HOTS algorithm. *arXiv:1205.6727*, 2012.

[Fer12b] Olivier Fercoq. PageRank optimization applied to spam detection. In *6th International conference on NETwork Games, COntrol and OPtimization (Netgcoop)*, pages 124–131, 2012.

[Fer13] Olivier Fercoq. Perron vector optimization applied to search engines. *Applied Numerical Mathematics*, 2013. doi :10.1016/j.apnum.2012.12.006.

[FABG13] Olivier Fercoq, Marianne Akian, Mustapha Bouhtou, and Stéphane Gaubert. Ergodic Control and Polyhedral approaches to PageRank Optimization. *IEEE TAC*, 58:134–148, 2013.

[CF16] Jean Clairambault and Olivier Fercoq. Physiologically structured cell population dynamic models with applications to combined drug delivery optimisation in oncology. *Mathematical Modelling of Natural Phenomena*, 11(6), 2016.

[FerPhD] Olivier Fercoq. *Optimization of Perron eigenvectors: from web ranking to chronotherapeutics.* PhD thesis, Ecole Polytechnique, 2012.

Chapter 2

Fast algorithms for composite differentiable-separable functions

In this line of work we focus on the solution of convex optimization problems with a huge number of variables of the form

$$\min_{x \in \mathbb{R}^N} \quad f(x) + \Psi(x). \tag{2.1}$$

Here $x = (x^{(1)}, \ldots, x^{(n)}) \in \mathbb{R}^N$ is a decision vector composed of n blocks with $x^{(i)} \in \mathbb{R}^{N_i}$, and $N = \sum_i N_i$. We assume that $\Psi : \mathbb{R}^N \to \mathbb{R} \cup \{+\infty\}$ is a convex (and lower semicontinous) block separable regularizer (e.g., the $L1$ norm).

Despite the major limitation on the nonsmooth part of the objective, this setup has had a lot of applications. For instance, a breakthrough was made in large scale sparse regression when coordinate descent was shown to be the most efficient method for the resolution of the Lasso problem [FHHT07].

2.1 Applications

In this section we describe four applications areas for accelerated coordinate descent, all motivated and building on the work [FR15] where we developed the Accelerated, Parallel and Proximal coordinate descent method APPROX (see Table 2.1). Empirical risk minimization is a natural framework for coordinate descent methods but acceleration increased its range of applicability to other domains.

2.1.1 Empirical risk minimization

Empirical risk minimization (ERM) is a powerful and immensely popular paradigm for training statistical (machine) learning models [SSBD14]. In statistical learning, one wishes to "learn" an unknown function $h^* : \mathcal{X} \to \mathcal{Y}$, where $\mathcal{X}$ (set of samples) and $\mathcal{Y}$ (set of labels) are arbitrary domains. Roughly speaking, the goal of statistical learning is to find a function (predictor, hypothesis) $h : \mathcal{X} \to \mathcal{Y}$ from some predefined set (hypothesis class) $\mathcal{H}$ of predictors which in some statistical sense is the best approximation of h^*. In particular, we assume that there is an *unknown* distribution $\mathcal{D}$ over $\xi \in \mathcal{X}$. Given a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, we define the *risk* (generalization error) associated with predictor $h \in \mathcal{H}$ to be

$$L_{\mathcal{D}}(h) = \mathbb{E}_{\xi \sim \mathcal{D}}\, \ell(h(\xi), h^*(\xi)). \tag{2.2}$$

Application	Paper	Section
Empirical risk minimization	[FQRT14, LLX15]	2.1.1
Submodular optimization	Ene and Nguyen [EN15]	2.1.2
Packing and covering linear programs	Allen-Zhu & Orecchia [AZO15]	2.1.3
Least-squares semidefinite programming	Sun, Toh and Yang [STY16]	2.1.4

Table 2.1: Selected applications of APPROX, developed by others after the publication of [FR15].

The goal of statistical learning is to find $h \in \mathcal{H}$ of minimal risk:

$$\min_{h \in \mathcal{H}} L_{\mathcal{D}}(h). \tag{2.3}$$

A natural albeit in general intractable choice of a loss function in some applications is $\ell(y, y') = 0$ if $y = y'$ and $\ell(y, y') = 1$ otherwise.

Let $\mathcal{X}$ be a collection of images, $\mathcal{Y} = \{-1, 1\}$ and let $h^*(\xi)$ be 1 of image ξ contains an image of a cat, and $h^*(\xi) = -1$ otherwise. If we are able to learn h^*, we will be able to detect images of cats. Problems where $\mathcal{Y}$ consist of two elements are called *classification problems*. The domain set can instead represent a video collection, a text corporus, a collection of emails or any other collection of objects which we can represent mathematically. If $\mathcal{Y}$ is a finite set consisting of more than two elements, we speak of *multi-class classification*. If $\mathcal{Y} = \mathbb{R}$, we speak of *regression*.

One of the fundamental issues making (2.3) difficult to solve is the fact that the distribution $\mathcal{D}$ is not known. ERM is a paradigm for overcoming this obstacle, assuming that we have access to independent samples from $\mathcal{D}$. In ERM, we first collect a *training set* of i.i.d. samples and their labels; that is, $\mathcal{S} = \{(\xi_j, y_j) \in \mathcal{X} \times \mathcal{Y} \ : \ j = 1, 2, \ldots, m\}$, where $y_j = h^*(\xi_j)$. Subsequently, we replace the expectation in (2.2) defining the risk, by a sample average approximation, which defines the *empirical risk*:

$$L_{\mathcal{S}}(h) = \frac{1}{m} \sum_{j=1}^{m} \ell(h(\xi_j), y_j).$$

The ERM paradigm is to solve the *empirical risk minimization* problem

$$\min_{h \in \mathcal{H}} L_{\mathcal{S}}(h) \tag{2.4}$$

instead of the harder risk minimization problem (2.3). In practice, $\mathcal{H}$ is often chosen to be a parametric class of functions described by a parameter $x \in \mathbb{R}^d$. For instance, let $\mathcal{X} \subseteq \mathbb{R}^d$ (d = number of features) and $\mathcal{Y} = \mathbb{R}$, and consider the class of *linear predictors*: $\mathcal{H} = \{h \ : \ h(\xi) = x^\top \xi\}$. Clearly, h is uniquely defined by $x \in \mathbb{R}^d$. Defining $\ell_j : \mathbb{R} \to \mathbb{R}$ via $\ell_j(t) = \frac{1}{m}\ell(t, y_j)$, and setting $f_j(x) = \ell_j(\xi_j^T x)$, we have

$$f(x) = \sum_{j=1}^{m} f_j(x) = \sum_{j=1}^{m} \ell_j(\xi_j^T x) = L_{\mathcal{S}}(h).$$

Hence, the ERM problem fits our framework (2.1), with $\psi \equiv 0$. However, in practice one often uses nonzero ψ, which is interpreted as a regularizer, and is included in order to prevent overfitting and hence allow the estimator to generalize to future, unobserved samples.

ERM has a tight connection with coordinate descent methods. Indeed, coordinate descent methods became very popular when their efficiency was proved for the Lasso problem [FHHT07]. It is a regression problem where the goal is to find sparse solutions. It can be written as

$$\min_{x \in \mathbb{R}^n} \frac{1}{2}\|Ax - b\|_2^2 + \lambda \|x\|_1$$

where $y \in \mathbb{R}^m$ is the signal, made out of m observations, A is a matrix of features of size $m \times n$ and x is the parameter vector that we seek to estimate so that it has a few nonzero coefficients. This is a non-differentiable problem but it can be decomposed in a smooth part $\frac{1}{2}\|Ax - b\|_2^2$ and a separable part $\lambda\|x\|_1 = \lambda \sum_{i=1}^{n} |x_i|$. It is thus amenable to coordinate descent methods.

If the number of features is larger than the number of examples ($d \gg m$), which is typical for Lasso problems, randomized coordinate descent is an efficient algorithm for solving (2.1) [RT14, RT15, SSZ13]. If the training set $\mathcal{S}$ is so large that it does not fit the memory (or disk space) of a single machine, one needs to employ a distributed computing system and solve ERM via a distributed optimization algorithm. One option is the use of distributed coordinate descent [RT16c, MRT15], known as Hydra. APPROX has been successfully applied in the distributed setting, leading to the Hydra2 method [FQRT14]. In this work, the authors solve an ERM problem involving a training set of several terabytes in size, and 50 billion features.

If the number of examples in the training set is larger than the number of features ($m \gg d$), it is typically not efficient to employ randomized coordinate descent, to the ERM problem directly. Instead, the state of the art methods are variants of randomized coordinate descent applied to the *dual* problem.

The (Fenchel) dual of the regularized ERM problem for linear predictors considered above has the form

$$\min_{y\in\mathbb{R}^m} \psi^*\left(\frac{1}{m}\sum_{j=1}^m y_j\xi_j\right) + \frac{1}{m}\sum_{j=1}^m \ell_j^*(-y_j),$$

where ψ^* (resp. ℓ_j^*) is the Fenchel conjugate of ψ (resp. ℓ_j). The function $y \mapsto \psi^*(\frac{1}{m}\sum_j y_j\xi_j)$ has Lipschitz gradient if we assume that ψ is strongly convex, and $y \mapsto \frac{1}{m}\sum_{j=1}^m \ell_j^*(-y_j)$ is separable. This also fits the framework (2.1), f corresponding to the first part of the objective (and consisting of a single summand), and ψ corresponding to the second part of the objective (block separability is implied by separability).

We developed APPROX with ERM as an application in mind and hence our numerical experiments consider two key ERM problems: the Lasso problem and the Support Vector Machine (SVM) problem.

Following our paper, Lin, Lu and Xiao [LLX15] proposed a version of APPROX designed for *strongly* convex problems. Their motivation was that practitioners often choose regularizers that are at the same time separable *and* strongly convex. This leads to problems for which we have a good lower bound on the strong convexity parameter. With this additional knowledge, they showed that the rate of convergence of a properly modified APPROX algorithm, applied to the dual problem, leads to state of the art complexity for a class of ERM problems.

2.1.2 Submodular optimization

Ene and Nguyen [EN15] showed how the APPROX algorithm leads to a state-of-the-art method for minimizing *decomposable submodular functions*. Submodular minimization has a vast and growing array of applications, including image segmentation [RKB04, EN15], graphical model structure learning, experimental design, Bayesian variable selection and minimizing matroid rank functions [Bac13].

We now briefly introduce the notion of submodularity. Let $V = \{1, 2, \ldots, d\}$ be a finite ground set. A real-valued set function $\phi : 2^V \to \mathbb{R}$ is called *modular*, if $\phi(\emptyset) = 0$ and there exists a vector $w \in \mathbb{R}^d$ such that $\phi(A) = \sum_{i\in A} w_i$ for all $\emptyset \neq A \subseteq V$. It is called *submodular* if $\phi(A) + \phi(B) \geq \phi(A \cap B) + \phi(A \cup B)$ for any two sets $A, B \subseteq V$. An equivalent and often more intuitive characterization of submodularity is the following "diminishing returns property": ϕ is submodular if and only if for all $A \subseteq B \subseteq V$ and $k \in V$ such that $k \notin B$, we have $\phi(A \cup \{k\}) - \phi(A) \geq \phi(B \cup \{k\}) - \phi(B)$.

Ene and Nguyen [EN15] consider the *decomposable* submodular minimization problem

$$\min_{A\subseteq V} \sum_{i=1}^n \phi_i(A), \tag{2.5}$$

where $\phi_i : 2^V \to \mathbb{R}$ are simple submodular functions (simplicity refers to the assumption that it is simple to minimize ϕ_i plus a modular function). Instead of solving (2.5) directly, one can focus on solving the unconstrained convex minimization problem

$$\min_{z\in\mathbb{R}^d} \sum_{i=1}^n \left(\hat{\phi}_i(z) + \frac{1}{2n}\|z\|^2\right), \tag{2.6}$$

where $\|\cdot\|$ is the standard Euclidean norm, and $\hat{\phi}_i : \mathbb{R}^d \to \mathbb{R}$ is the Lovász extension of ϕ_i (i.e., the support function of the base polytope $P_i \subset \mathbb{R}^d$ of ϕ_i). Given a solution z, one recovers the solution of (2.5) by setting

$$A = A(z) = \{k \in V \ : \ z_k \geq 0\}. \tag{2.7}$$

Further, instead of solving (2.6), one focuses on its (Fenchel) dual:

$$\min_{x^{(1)}\in P_1,\ldots,x^{(n)}\in P_n} f(x) = \frac{1}{2}\left\|\sum_{i=1}^n x^{(i)}\right\|^2. \tag{2.8}$$

It can be shown that if $x = (x^{(1)}, \ldots, x^{(n)}) \in \mathbb{R}^{nd} = \mathbb{R}^N$ solves (2.8), then

$$z = -\sum_{i=1}^n x^{(i)} \tag{2.9}$$

solves (2.6). Note that f is a convex quadratic function. If we let Ψ be the indicator function of the set $P = P_1 \times \cdots \times P_n \subseteq \mathbb{R}^N$, i.e., $\Psi(x) = 0$ if $x \in P$ and $\Psi(x) = +\infty$ otherwise, then (2.8) is of the form (2.1), where $N_i = d$ for all i. It remains to apply the APPROX method to this problem, and transform the solution back via (2.9) and then (2.7) to obtain solution of the original problem (2.5).

2.1.3 Packing and covering linear programs

Packing and covering problems are a pair of mutually-dual linear programming problems of the form

$$\begin{aligned} \text{Packing LP:} \quad & \max_{x \geq 0} \{1^T x \ : \ Ax \leq 1\} \\ \text{Covering LP:} \quad & \min_{y \geq 0} \{1^T y \ : \ A^T y \leq 1\}, \end{aligned}$$

where A is a real matrix, and 1 denotes the vector of appropriate dimension with all entries equal to 1. These problems become more difficult as the size of A grows since each iteration of interior-point solvers becomes more expensive. Allen-Zhu and Orecchia [AZO15] developed algorithms for these problems whose complexity is $O(N_A \log(N_A) \log(\epsilon^{-1})/\epsilon)$ for packing and $O(N_A \log(N_A) \log(\epsilon^{-1})\epsilon^{-1.5})$ for covering LP, respectively. This complexity is nearly linear in the size of the problem $N_A = \text{nnz}(A)$ (number of nonzero entries in A), does not depend on the magnitude of the elements of A and has a better dependence on the accuracy ϵ than other nearly-linear time approaches. The improvement in the complexity is due to the use of accelerated proximal coordinate descent techniques such as those developed in our paper, combined with extra techniques, such as the use of an exponential penalty.

2.1.4 Least squares semi-definite programming

Semidefinite programming have a very important role in optimization due to its ability to model and efficiently solve a wide array of problems appearing in fields such as control, network science, signal processing and computer science [VB96, Tod01, BTN01, WSV10]. In semidefinite programming, one aims to minimize a linear function in a matrix variable, subject to linear equality and inequality constraints and the additional requirement that the matrix variable be positive semidefinite.

Sun, Toh and Yang [STY16] consider the canonical semidefinite program (SDP)

$$\begin{aligned} \min_{X \in \mathcal{S}^n_+,\ s \in \mathbb{R}^{m_I}} \quad & \langle C, X \rangle \\ \text{subject to} \quad & \mathcal{A}_E(X) = b_E,\ \mathcal{A}_I(X) = s,\ L \leq X \leq U,\ l \leq s \leq u, \end{aligned}$$

where $\langle C, X \rangle$ is the trace inner product, $\mathcal{S}^n_+$ is the cone of $n \times n$ symmetric positive semidefinite matrices, $\mathcal{A}_E : \mathcal{S}^n_+ \to \mathbb{R}^{m_E}$ and $\mathcal{A}_I : \mathcal{S}^n_+ \to \mathbb{R}^{m_I}$ are linear maps, $L \leq U$ are given positive semidefinite matrices and $l \leq u$ are given vectors in $\mathbb{R}^{m_I}$.

The above SDP can be solved by a proximal point algorithm (PPA) of Rockafellar [Roc76a, Roc76b]. In each iteration of PPA, one needs to solve a *least-squares semidefinite program (LS-SDP)* of the form

$$\begin{aligned} (X^{k+1}, s^{k+1}) = \arg \min_{X \in \mathcal{S}^n_+, s \in \mathbb{R}^{m_I}} \quad & \langle C, X \rangle + \frac{1}{2\sigma_k}(\|X - X^k\|^2 + \|s - s^k\|^2) \\ \text{subject to} \quad & \mathcal{A}_E(X) = b_E,\ \mathcal{A}_I(X) = s,\ L \leq X \leq U,\ l \leq s \leq u, \end{aligned}$$

where (X^k, s^k) is the previous iterate, and $\sigma_k > 0$ a regularization parameter. Sun, Toh and Yang [STY16] observe that the dual of LS-SDP has block-separable constraints and can hence be written in the form (2.1), with either 2 or 4 blocks ($n = 2$ or $n = 4$). They used this observation as a starting point to propose new algorithms for LS-SDP that combine advanced linear algebra techniques, a fine study of the errors made by each inner solver, and coordinate descent ideas. They consider APPROX and block coordinate descent as natural competitors to their specialized methods. They implemented the methods using 4 blocks, each equipped with a nontrivial norm defined by a well chosen positive semi-definite matrix $B_i \in \mathbb{R}^{N_i \times N_i}$ and approximate solutions to the proximity operators. Finally, Sun, Toh and Yang conducted extensive experiments on 616 SDP instances coming from relaxations of combinatorial problems. It is worth noting that on these instances, APPROX is vastly faster than standard (non-accelerated) block coordinate descent.

2.2 APPROX

2.2.1 Description of the algorithm

Algorithm 4 APPROX: **A**ccelerated **P**arallel **Prox**imal Coordinate Descent Method

1: Choose $x_0 \in \mathbb{R}^N$ and set $z_0 = x_0$ and $\theta_0 = \frac{\tau}{n}$
2: **for** $k \geq 0$ **do**
3: $y_k = (1-\theta_k)x_k + \theta_k z_k$
4: Generate a random set of blocks $S_k \sim \hat{S}$
5: $z_{k+1} = z_k$
6: **for** $i \in S_k$ **do**
7: $z_{k+1}^{(i)} = \arg\min_{z \in \mathbb{R}^{N_i}} \left\{ \langle \nabla_i f(y_k), z - y_k^{(i)} \rangle + \frac{n\theta_k v_i}{2\tau} \|z - z_k^{(i)}\|_{(i)}^2 + \Psi_i(z) \right\}$
8: **end for**
9: $x_{k+1} = y_k + \frac{n}{\tau}\theta_k(z_{k+1} - z_k)$
10: $\theta_{k+1} = \frac{\sqrt{\theta_k^4 + 4\theta_k^2} - \theta_k^2}{2}$
11: **end for**

The method starts from $x_0 \in \mathbb{R}^N$ and generates three vector sequences denoted $\{x_k, y_k, z_k\}_{k \geq 0}$. In Step 3, y_k is defined as a convex combination of x_k and z_k, which may in general be full dimensional vectors. This is not efficient; but following ideas by [LS13], we showed that it is possible to implement the method in such a way that it not necessary to ever form y_k.

The strength of the algorithm is its excellent worst case bound stated below. It relies on the assumption of the existence of an expected separable overapproximation, which is a generalization of gradient Lipschitz continuity tailored for parallel coordinate descent methods.

Assumption 1 (Expected Separable Overapproximation [RT15, FR17]). *1. f is convex and differentiable.*

2. $\hat{S}$ is a uniform block sampling. That is, $\hat{S}$ is a random subset of $[n] = \{1, 2, \ldots, n\}$ with the property[1] *that $\mathbb{P}(i \in \hat{S}) = \mathbb{P}(j \in \hat{S})$ for all $i, j \in [n]$. Let $\tau = \mathbb{E}[|\hat{S}|]$.*

3. There are computable constants $v = (v_1, \ldots, v_n) > 0$ for which the pair $(f, \hat{S})$ admits the Expected Separable Overapproximation (ESO):

$$\mathbb{E}\left[f(x + h_{[\hat{S}]})\right] \leq f(x) + \frac{\tau}{n}\left(\langle \nabla f(x), h \rangle + \frac{1}{2}\|h\|_v^2\right), \quad x, h \in \mathbb{R}^N, \tag{2.10}$$

where $h_{[\hat{S}]}^{(i)} = \begin{cases} h^{(i)} & \text{if } i \in \hat{S} \\ 0 & \text{otherwise} \end{cases}$

If the above inequality holds, for simplicity we will write $(f, \hat{S}) \sim \mathrm{ESO}(v)$.

Theorem 2. *Let Assumption 1 be satisfied, with $(f, \hat{S}) \sim \mathrm{ESO}(v)$, where $\tau = \mathbb{E}[|\hat{S}|] > 0$. Let $x_0 \in \operatorname{dom}\Psi$, and assume that the random sets S_k in Algorithm 4 are chosen independently, following the distribution of $\hat{S}$. Let x_* be any optimal point of problem (2.1). Then the iterates $\{x_k\}_{k \geq 1}$ of APPROX satisfy:*

$$\mathbb{E}[F(x_k) - F(x_*)] \quad \leq \quad \frac{4n^2 C_*}{((k-1)\tau + 2n)^2}, \tag{2.11}$$

where

$$C_* \quad = \quad \left(1 - \frac{\tau}{n}\right)(F(x_0) - F(x_*)) + \frac{1}{2}\|x_0 - x_*\|_v^2. \tag{2.12}$$

In other words, for any $0 < \epsilon \leq C_*$, the number of iterations for obtaining an ϵ-solution in expectation does not exceed

$$k \quad = \quad \left\lceil \frac{2n}{\tau}\left(\sqrt{\frac{C_*}{\epsilon}} - 1\right) + 1 \right\rceil. \tag{2.13}$$

[1] It is easy to see that if $\hat{S}$ is a uniform sampling, then necessarily, $\mathbb{P}(i \in \hat{S}) = \frac{\mathbb{E}[|\hat{S}|]}{n}$ for all $i \in [n]$.

The main novelty in the analysis of APPROX was to show that the iterates remain within the constraint set, although they are defined as an overrelaxation of admissible points in Step 9.

Lemma 2. *Let $\{x_k, z_k\}_{k\geq 0}$ be the iterates of Algorithm 4. Then for all $k \geq 0$ we have*

$$x_k = \sum_{l=0}^{k} \gamma_k^l z_l, \tag{2.14}$$

where the constants $\gamma_k^0, \gamma_k^1, \ldots, \gamma_k^k$ are non-negative and sum to 1. That is, x_k is a convex combination of the vectors $z_1, \ldots, z_k$. In particular, the constants are defined recursively in k by setting $\gamma_0^0 = 1$, $\gamma_1^0 = 0$, $\gamma_1^1 = 1$ and for $k \geq 1$,

$$\gamma_{k+1}^l = \begin{cases} (1-\theta_k)\gamma_k^l, & l = 0, \ldots, k-1, \\ \theta_k(1 - \frac{n}{\tau}\theta_{k-1}) + \frac{n}{\tau}(\theta_{k-1} - \theta_k), & l = k, \\ \frac{n}{\tau}\theta_k, & l = k+1. \end{cases} \tag{2.15}$$

Moreover, for all $k \geq 0$, the following identity holds

$$\gamma_{k+1}^k + \frac{n-\tau}{\tau}\theta_k = (1-\theta_k)\gamma_k^k. \tag{2.16}$$

2.2.2 Comparison with previous approaches

APPROX is the first randomized block coordinate descent method (APPROX) which is simultaneously *accelerated*, *parallel* and *proximal*. In fact, we are not aware of any published results on accelerated coordinate descent which would either be proximal *or* parallel. Our method is *accelerated* in the sense that it achieves an $O(1/k^2)$ convergence rate, where k is the iteration counter. The first *gradient* method with this convergence rate is due to Nesterov [Nes83]; see also [Tse08, BT09]. Accelerated randomized coordinate descent method, for convex minimization without constraints, was originally proposed in 2010 by Nesterov [Nes12a].

Paper	Eff	Blck	Prx	Par	Acc	Notable feature
Leventhal & Lewis '08 [LL10]	✓	×	×	×	×	quadratic f
S-Shwartz & Tewari '09 [SST11]	✓	×	ℓ_1	×	×	1st ℓ_1-regularized
Nesterov '10 [Nes12a]	×	✓	×	×	✓	1st blck & 1st acc
Richtárik & Takáč '11 [RT14]	✓	✓	✓	×	×	1st proximal
Bradley et al '12 [BKBG11]	✓	×	ℓ_1	✓	×	ℓ_1-regularized parallel
Richtárik & Takáč '12 [RT15]	✓	✓	✓	✓	×	1st general parallel
S-Shwartz & Zhang '12 [SSZ12]	✓	✓	✓	×	×	1st primal-dual
Necoara et al '12 [NNG12]	✓	✓	×	×	×	2-coordinate descent
Takáč et al '13 [TBRS13]	✓	×	×	✓	×	1st primal-d. & parallel
Tappenden et al '13 [TRG16a]	✓	✓	✓	×	×	1st inexact
Necoara & Clipici '13 [NC13]	✓	✓	✓	×	×	coupled constraints
Lin & Xiao '13 [LX15b]	×	✓	×	×	✓	improvements on [Nes12a, RT14]
Fercoq & Richtárik '13 [FR17]	✓	✓	✓	✓	×	1st nonsmooth f
Lee & Sidford '13 [LS13]	✓	×	×	×	✓	1st efficient accelerated
Richtárik & Takáč '13 [RT16a]	✓	×	✓	✓	×	1st distributed
Liu et al '13 [LWR$^+$15]	✓	×	×	✓	×	1st asynchronous
S-Shwartz & Zhang '13 [SSZ14]	✓	×	✓	×	✓	acceleration in the primal
Richtárik & Takáč '13 [RT16b]	✓	×	×	✓	×	1st arbitrary sampling
This paper '13	✓	✓	✓	✓	✓	**5 times ✓**

Table 2.2: An overview of selected recent papers proposing and analyzing the iteration complexity of *randomized* coordinate descent methods. The years correspond to the time the papers were first posted online (e.g., onto arXiv), and not the eventual publication time. "Eff" = the cost of each iteration is low (in particular, independent of the problem dimension N); "Blck" = works with blocks of coordinates; "Prx" = can handle proximal setup (has ψ term); "Par" = can update more blocks per iteration; "Acc" = accelerated, i.e., achieving the optimal $O(1/k^2)$ rate for non-strongly convex objectives. Our algorithm has all these desirable properties. In the last column we highlight a single notable feature, necessarily chosen subjectively, of each work.

Several variants of proximal and parallel (but non-accelerated) randomized coordinate descent methods were proposed [BKBG11, RT15, FR17, RT16a]. In Table 2.2 we provide a list of research papers

proposing and analyzing randomized coordinate descent methods. The table substantiates our observation that while the block ("Blck" column) and proximal ("Prx" column) setup is relatively common in the literature, parallel methods ("Par" column) are much less studied, and there is just a handful of papers dealing with accelerated variants ("Acc" column). Moreover, existing accelerated methods are not efficient ("Eff" column)—with the exception of [LS13]

2.3 Accelerated coordinate descent in a distributed setting

More and more often in modern applications, the data describing the problem is so large that it does not fit into the RAM of a single computer. In such a case, unless the application at hand can tolerate slow performance due to frequent HDD reads/writes, it is often necessary to distribute the data among the nodes of a cluster and solve the problem in a distributed manner. With such big data problems it is necessary to design algorithms able to utilize modern parallel computing architectures. This resulted in an interest in parallel [RT15, FR17, RT16b] and distributed [RT16a] coordinate descent methods.

The core of the paper [FQRT14] forms the development of *new stepsizes* that improve on previous works on parallel coordinate descent using the following assumptions on the objective function

$$f(x) + \Psi(x) = \sum_{j=1}^{m} \phi_j(e_j^\top Ax) + \sum_{i=1}^{n} \Psi_i(x^{(i)}) \ .$$

We also assumed that the data is partitioned among c computers dealing with s coordinates each. We assume that each computer updates τ coordinates in parallel at each iteration. We denote $n = cs$. Let ω_j be the number of nonzeros in the jth row of A and ω_j' be the number of "partitions active at row j", i.e., the number of indexes $l \in \{1, \dots, c\}$ for which the set $\{i \in \mathcal{P}_l : A_{ji} \neq 0\}$ is nonempty. As soon as A does not have an empty row or column, we know that $1 \leq \omega_j \leq n$ and $1 \leq \omega_j' \leq c$.

The goal is to show that there exists computable constants $v_1, \dots, v_n$ such that the ESO inequality (2.10) holds:

$$\mathbb{E}\left[f(x + h_{[\hat{S}]})\right] \leq f(x) + \frac{\tau}{s}\left(\langle \nabla f(x), h\rangle + \frac{1}{2}\|h\|_v^2\right), \quad x, h \in \mathbb{R}^N.$$

Then the algorithm will consist in Algorithm 4 that uses the Expected Separable Overapproximation (ESO) inequality (2.10) required by Assumption 1. On top of allowing us to prove the convergence of the algorithm, the ESO has the following benefits:

(i) Since the overapproximation is a convex quadratic in h, *it is easy to compute* $h(x)$.

(ii) Since the overapproximation is block separable, one *can compute the updates* $h^{(i)}(x)$ *in parallel* for all $i \in \{1, 2, \dots, n\}$.

(iii) For the same reason, one *can compute the updates or* $i \in S_k$ *only*, where S_k is the sample set drawn at iteration k following the law describing $\hat{S}$.

Finding smaller constants v_i directly transfers into longer step-sizes and thus a potentially better parallelization speedup.

The first proposition corresponds to the shared memory framework.

Proposition 1. *Suppose that* $f(x) = \sum_{j=1}^{m} \phi_j(e_j^\top Ax)$ *and that* $\hat{S}$ *consists in choosing* τc *coordinates uniformly at random. Then* f *satisfies the Expected Separable Overapproximation* (2.10) *with parameters*

$$v_i = \sum_{j=1}^{m} \Big(1 + \frac{(\omega_j - 1)(\tau c - 1)}{sc - 1}\Big) A_{ji}^2.$$

The main conclusion of the study, made precise in the next proposition, is that as long as the number of processors per computer $\tau \geq 2$, the effect of partitioning the data (across the nodes) on the iteration complexity of the algorithm is negligible, and vanishes as τ increases.

Proposition 2. *Suppose that* $f(x) = \sum_{j=1}^{m} \phi_j(e_j^\top Ax)$ *Suppose that* $\hat{S}$ *consists in choosing for each computer,* τ *coordinates uniformly at random among the* s *ones it manages. Then* f *satisfies the Expected Separable Overapproximation* (2.10) *with parameters*

$$v_i = \sum_{j=1}^{n} \Big(1 + \frac{(\omega_j - 1)(\tau - 1)}{s - 1} + (\frac{\tau}{s} - \frac{\tau - 1}{s - 1})\frac{\omega'_j - 1}{\omega'_j}\omega_j\Big)A_{ji}^2.$$

In [FQRT14], we gave an extensive comparison of the ESOs available in the literature and showed that the newly proposed is much better than former ones.

2.4 Restart of accelerated gradient methods

On top of parallel processing, I also studied restarting schemes as a complementary mean of acceleration.

For a mild additional computational cost, accelerated gradient methods transform the proximal gradient method, for which the optimality gap $F(x_k) - F^\star$ decreases as $O(1/k)$, into an algorithm with "optimal" $O(1/k^2)$ complexity [Nes83]. Accelerated variants include the dual accelerated proximal gradient [Nes05b, Nes13a], the accelerated proximal gradient method (APG) [Tse08] and FISTA [BT09]. Gradient-type methods, also called first-order methods, are often used to solve large-scale problems because of their good scalability and easiness of implementation that facilitates parallel and distributed computations.

When solving a convex problem whose objective function satisfies a local quadratic error bound (this is a generalization of strong convexity), classical (non-accelerated) gradient and coordinate descent methods automatically have a linear rate of convergence, i.e. $F(x_k) - F^\star \in O((1-\mu)^k)$ for a problem dependent $0 < \mu < 1$ [NNG12, DL16], whereas one needs to know explicitly the strong convexity parameter in order to set accelerated gradient and accelerated coordinate descent methods to have a linear rate of convergence, see for instance [LS13, LMH15, LLX14, Nes12a, Nes13a]. Setting the algorithm with an incorrect parameter may result in a slower algorithm, sometimes even slower than if we had not tried to set an acceleration scheme [OC12]. This is a major drawback of the method because in general, the strong convexity parameter is difficult to estimate.

In the context of accelerated gradient method with unknown strong convexity parameter, Nesterov [Nes13a] proposed a restarting scheme which adaptively approximates the strong convexity parameter. The same idea was exploited by Lin and Xiao [LX15a] for sparse optimization. Nesterov [Nes13a] also showed that, instead of deriving a new method designed to work better for strongly convex functions, one can restart the accelerated gradient method and get a linear convergence rate. However, the restarting frequency he proposed still depends explicitly on the strong convexity of the function and so O'Donoghue and Candes [OC12] introduced some heuristics to adaptively restart the algorithm and obtain good results in practice.

The restarted algorithm is given in Algorithm 5, where APPROX$(f, \psi, \bar{x}_r, K)$ means Algorithm 4 run on the function $f + \psi$ with initial point $\bar{x}_r$ and for K iterations.

Algorithm 5 APPROX with restart

Choose $x_0 \in \operatorname{dom}\psi$ and set $\bar{x}_0 = x_0$.
Choose RestartTimes $\subseteq \mathbb{N}$.
for $r \geq 0$ **do**
  K = RestartTimes$(r+1)$ - RestartTimes(r)
  $\bar{x}_{r+1}$ = APPROX$(f, \psi, \bar{x}_r, K)$
end for

Gradient method In [FQ17], we showed that, if the objective function is convex and satisfies a local quadratic error bound, we can restart accelerated gradient methods at *any* frequency and get a linearly convergent algorithm. The rate depends on an estimate of the quadratic error bound and we show that for a wide range of this parameter, one obtains a faster rate than without acceleration. In particular, we do not require this estimate to be smaller than the actual value. In that way, our result supports and explains the practical success of arbitrary periodic restart for accelerated gradient methods.

Algorithm	Complexity bound	Assumption
Nesterov [Nes13a]	$O\Big(\frac{1}{\sqrt{\mu_F}}\ln\big(\frac{1}{\mu_F}\big)\ln\big(\frac{1}{\mu_F\epsilon}\big)\Big)$	strong convexity
Lin & Xiao [LX15a]	$O\Big(\frac{1}{\sqrt{\mu_F}}\ln\big(\frac{1}{\mu_F}\big)\ln\big(\frac{1}{\mu_F\epsilon}\big)\Big)$	strong convexity
Liu & Yang [LY17]	$O\Big(\frac{1}{\sqrt{\mu_F}}\ln\big(\frac{1}{\mu_F}\big)^2\ln\big(\frac{1}{\epsilon}\big)\Big)$	Hölderian error bound
Fercoq & Qu [FQ17]	$O\Big(\frac{1}{\sqrt{\mu_F}}\ln\big(\frac{1}{\mu_F\epsilon}\big)\Big)$	local quadratic error bound

Table 2.3: Comparison of the iteration complexity of accelerated gradient methods with adaptation to the local error bound.

Then, as the rate of convergence depends on the match between the frequency and the quadratic error bound, we design a scheme to automatically adapt the frequency of restart from the observed decrease of the norm of the gradient mapping. The approach follows the lines of [Nes13a, LX15a, LY17]. We proved that, if our current estimate of the local error bound were correct, the norm of the gradient mapping would decrease at a prescribed rate. We just need to check this decrease and when the test fails, we have a certificate that the estimate was too large.

Our algorithm has a better theoretical bound than previously proposed methods for the adaptation to the quadratic error bound of the objective. In particular, we can make use of the fact that our study shows that the norm of the gradient mapping will decrease even when we had a wrong estimate of the local error bound.

Coordinate descent In [FQ16, FQ18], we studied the case of accelerated coordinate descent. The adaptive restart of randomized accelerated coordinate descent methods is more complex than in the deterministic case. As the complexity bound holds in expectation only, one cannot rely on this bound to estimate whether the rate of convergence is in line with our estimate of the local error bound, as was done in the deterministic case.

We considered four setups, the first one in [FQ16], the three other ones in [FQ18]:

1. In the case where the objective satisfies a global quadratic error bound, we proposed a fixed restarting scheme. We considered restarting at a point which is a convex combination of all previous iterates and showed linear convergence.

   Theorem 3. *Let γ_k^i be the coefficients defined in* (2.15) *and*

   $$\mathring{x}_k = \frac{1}{\sum_{i=0}^{k-1}\frac{\gamma_k^i}{\theta_{i-1}^2}+\frac{1}{\theta_0\theta_{k-1}}-\frac{1-\theta_0}{\theta_0^2}}\left(\sum_{i=0}^{k-1}\frac{\gamma_k^i}{\theta_{i-1}^2}x_i+\left(\frac{1}{\theta_0\theta_{k-1}}-\frac{1-\theta_0}{\theta_0^2}\right)x_k\right)$$

   a convex combination of the k first iterates of APPROX. Let $\sigma\in[0,1]$, $\bar{x}_k=\sigma x_k+(1-\sigma)\mathring{x}_k$. Denote $\Delta(x):=\frac{1-\theta_0}{\theta_0^2}(F(x)-F(x_*))+\frac{1}{2\theta_0^2}\operatorname{dist}_v(x,\mathcal{X})^2$ and $m_k(\mu):=\frac{\mu\theta_0^2}{1+\mu(1-\theta_0)}\left(\sum_{i=0}^{k-1}\frac{\gamma_k^i}{\theta_{i-1}^2}+\frac{1}{\theta_0\theta_{k-1}}-\frac{1-\theta_0}{\theta_0^2}\right)$. We have

   $$\mathbb{E}[\Delta(\bar{x}_k)]\leq\max\left(\sigma,1-\sigma m_k(\mu_F(v))\right)\Delta(x_0).$$

2. If the local quadratic error bound coefficient μ of the objective function is known, then we show that setting a restarting period as $O(1/\sqrt{\mu})$ yields an algorithm with optimal rate of convergence. More precisely restarted APPROX admits the same theoretical complexity bound as the accelerated coordinate descent methods for strongly convex functions developed in [LLX14], is applicable with milder assumptions and exhibits better performance in numerical experiments.

   Proposition 3. *Let x_k be the iterate of APPROX applied to the objective function $F=f+\psi$ with quadratic error bound coefficient μ on the level set $\{x:F(x)\leq F(x_0)\}$. Denote: $\tilde{x}=x_k1_{F(x_k)\leq F(x_0)}+x_01_{F(x_k)>F(x_0)}$. We have*

   $$\mathbb{E}[F(\tilde{x})-F^\star]\leq\theta_{k-1}^2\left(\frac{1-\theta_0}{\theta_0^2}+\frac{1}{\theta_0^2\mu}\right)(F(x_0)-F^\star).$$

   Moreover, given $\alpha<1$, if $k\geq\frac{2}{\theta_0}\left(\sqrt{\frac{1+\mu_F(v,x_0)}{\alpha\mu_F(v,x_0)}}-1\right)+1$, then $\mathbb{E}[F(\tilde{x})-F^\star]\leq\alpha(F(x_0)-F^\star)$.

3. If the objective function is strongly convex, we show that we can restart the accelerated coordinate descent method at the last iterate at any frequency and get a linearly convergent algorithm. The rate depends on an estimate of the local quadratic error bound and we show that for a wide range of this parameter, one obtains a faster rate than without acceleration. In particular, we do not require the estimate of the error bound coefficient to be smaller than the actual value. The difference with respect to [FQ16] is that in this section, we show that there is no need to restart at a complex combination of previous iterates.

   Theorem 4. *Denote* $\Delta(x) = \frac{1-\theta_0}{\theta_0^2}(F(x) - F^\star) + \frac{1}{2\theta_0^2}\operatorname{dist}_v(x, \mathcal{X})^2$. *Assume that* F *is* μ_F *strongly convex. Then the iterates of APPROX satisfy*

$$\mathbb{E}[\Delta(x_K)] \leq \frac{1 + (1-\theta_0)\mu_F}{1 + \frac{\theta_0^2}{2\theta_{K-1}^2}\mu_F}\Delta(x_0)$$

4. If the local error bound coefficient is not known, we introduce a variable restarting periods and show that up to a $\log(\log 1/\epsilon)$ term, the algorithm is as efficient as if we had known the local error bound coefficient.

   Theorem 5. *We define the sequence*

$$K_0 = K_0 \quad K_1 = 2^1 K_0 \quad K_2 = K_0 \quad K_3 = 2^2 K_0 \quad K_4 = K_0 \quad K_5 = 2^1 K_0 \quad K_6 = K_0 \quad K_7 = 2^3 K_0 \dots$$

   such that $K_{2^j-1} = 2^j K_0$, $\forall j \in \mathbb{N}$ *and* $|\{l \leq 2^J - 1 \mid K_l = 2^j K_0\}| = 2 \times |\{l \leq 2^J - 1 \mid K_l = 2^{j-1} K_0\}|$ *for all* $j \in \{1, \dots J-1\}$, $J \in \mathbb{N}$.

   We denote $\delta_0 = F(\tilde{x}_0) - F^*$ *and* $K(\mu_F) = \frac{2}{\theta_0}\left(\sqrt{\frac{1+\mu_F}{e^{-2}\mu_F}} - 1\right) + 1$, *where* μ_F *is the (unknown) quadratic error bound coefficient* μ *of* F.

   Suppose that the restart times are given by the variable restart periods defined above. Then the iterates of restarted APPROX satisfy $\mathbb{E}(F(x_k) - F(x_*)) \leq \epsilon$ *as soon as*

$$k \geq \left(\max\left(\log\left(\frac{K(\mu_F)}{K_0}\right), 0\right) + \log_2\left(\frac{\log(\frac{\delta_0}{\epsilon})}{2}\right)\right)\frac{\log(\frac{\delta_0}{\epsilon})}{2}\max(K(\mu_F), K_0)\ .$$

On Figure 2.1, we can see that restarting the accelerated coordinate descent clearly outperforms both coordinate descent and plain accelerated coordinate descent. Moreover, using a version of accelerated coordinate descent designed for strongly convex functions [LLX14] is not safe: it may fail to converge or be very slow depending on the estimate of strong convexity that we are using.

2.5 Using second order information

There have been several attempts at designing methods that combine randomization with the use of curvature (second-order) information. For example, methods based on running coordinate ascent in the dual such as [RT15, FR17, FR15, FQRT14, QRZ15] use curvature information contained in the diagonal of a bound on the Hessian matrix. Block coordinate descent methods, when equipped with suitable data-dependent norms for the blocks, use information contained in the block diagonal of the Hessian [TRG16b]. A more direct route to incorporating curvature information was taken by [SYG07] in their stochastic L-BFGS method and by [BHNS14] and [SDPG14] in their stochastic quasi-Newton methods. Complexity estimates are not easy to find. An exception in this regard is the work of [BBG09], who give a $O(1/\epsilon)$ complexity bound for a Quasi-Newton SGD method.

An alternative approach is to consider block coordinate descent methods with overlapping blocks [TY09a, FT15]. While typically efficient in practice, none of the methods mentioned above are equipped with complexity bounds (bounds on the number of iterations). Tseng and Yun [TY09a] only showed convergence to a stationary point and focus on non-overlapping blocks for the rest of their paper. Numerical evidence that this approach is promising is provided in [FT15] with some mild convergence rate results but no iteration complexity.

The main contribution of the work [QRTF16] is the analysis of stochastic block coordinate descent methods with overlapping blocks. We then instantiate this to get a new algorithm—Stochastic Dual

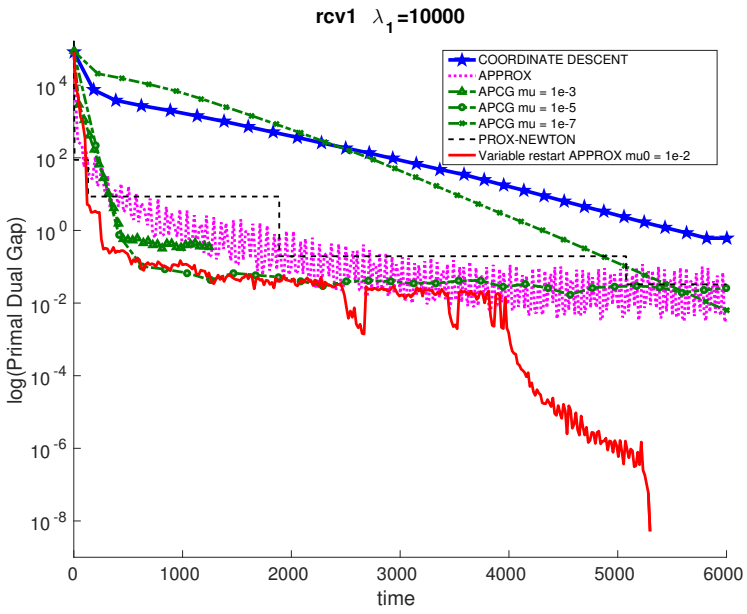

Figure 2.1: Comparison of (accelerated) coordinate descent algorithms for the logistic regression problem on the dataset RCV1: coordinate descent, APPROX, APCG [LLX14] with $\mu \in \{10^{-3}, 10^{-5}, 10^{-7}\}$, Prox-Newton [LSS12] and APPROX with variable restart initiated with $K_0 = K(10^{-2})$.

Newton Ascent (SDNA)—for solving a regularized Empirical Risk Minimization problem with smooth loss functions and a strongly convex regularizer (primal problem). Our method is stochastic in nature and has the capacity to utilize all curvature information inherent in the data.

SDNA in each iteration picks a random subset of the dual variables (which corresponds to picking a minibatch of examples in the primal problem), following an arbitrary probability law, and maximizes, exactly, the dual objective restricted to the random subspace spanned by the coordinates. Equivalently, this can be seen as the solution of a proximal subproblem involving a random principal submatrix of the Hessian of the quadratic function. Hence, SDNA utilizes all curvature information available in the random subspace in which it operates. Note that this is very different from the update strategy of parallel / minibatch coordinate descent methods. Indeed, while these methods also update a random subset of variables in each iteration, they instead only utilize curvature information present in the diagonal of the Hessian.

In the case of quadratic loss, and when viewed as a primal method, SDNA can be interpreted as a variant of the Iterative Hessian Sketch algorithm [PW14].

SDCA-like methods need *more* passes through data to converge as the minibatch size increases. However, SDNA enjoys the opposite behavior: with increasing minibatch size, up to a certain threshold, SDNA needs fewer passes through data to converge. This observation is confirmed by our numerical experiments. In particular, we show that the expected duality gap decreases at a geometric rate which i) is better than that of SDCA-like methods such as SDCA [SSZ13] and QUARTZ [QRZ15], and ii) improves with increasing minibatch size. This improvement does not come for free: as we increase the minibatch size, the subproblems grow in size as they involve larger portions of the Hessian. We find through experiments that for some, especially dense problems, even relatively small minibatch sizes lead to dramatic speedups in actual runtime. For instance on Figure 2.2 a batch size of 16 is enough to take a lot of profit from second order information.

We assume a strong convexity and a smoothness assumption involving data-dependent norms:

$$f(x) + \langle \nabla f(x), h \rangle + \tfrac{1}{2}\langle \mathbf{G}h, h \rangle \leq f(x+h), \tag{2.17}$$

$$f(x+h) \leq f(x) + \langle \nabla f(x), h \rangle + \tfrac{1}{2}\langle \mathbf{M}h, h \rangle. \tag{2.18}$$

We consider a sampling of the coordinates $\hat{S}$ and real numbers $v_1, \ldots, v_n$ satisfying a generalization of Assumption 1 to nonuniform probabilities such that $\mathbb{P}(i \in \hat{S}) = p_i > 0$ for all i. A large part of the

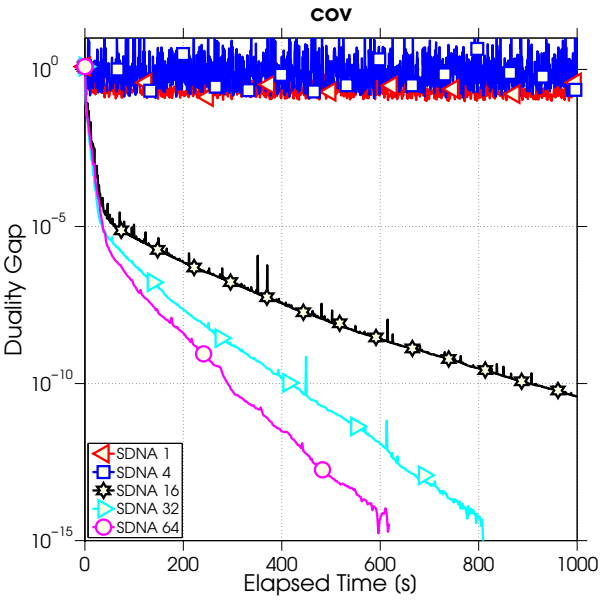

Figure 2.2: Runtime of SDNA for minibatch sizes $\tau = 1, 4, 16, 32, 64$ for a L_2-regularized least squares problem on the `cov` dataset: $d = 54$, $n = 522,911$).

paper consists in comparing three quantities:

$$\sigma_1 := \lambda_{\min}\left(\mathbf{G}^{1/2}\mathbb{E}\left[\left(\mathbf{M}_{\hat{S}}\right)^{\dagger}\right]\mathbf{G}^{1/2}\right), \tag{2.19}$$

$$\sigma_2 := \lambda_{\min}\left(\mathbf{G}^{1/2}\mathbf{D}(p)\left(\mathbb{E}\left[\mathbf{M}_{\hat{S}}\right]\right)^{-1}\mathbf{D}(p)\mathbf{G}^{1/2}\right), \tag{2.20}$$

$$\sigma_3 := \lambda_{\min}\left(\mathbf{G}^{1/2}\mathbf{D}(p)\mathbf{D}(v^{-1})\mathbf{G}^{1/2}\right), \tag{2.21}$$

where $\lambda_{\min}$ stands for the smallest eigenvalue. We show that SDNA converges linearly as $\mathbb{E}[f(x^{k+1}) - f(x^*)] \leq (1-\sigma_1)\mathbb{E}[f(x^k) - f(x^*)]$ while the parallel coordinate descent method (PCDM) converges as $\mathbb{E}[f(x^{k+1}) - f(x^*)] \leq (1-\sigma_3)\mathbb{E}[f(x^k) - f(x^*)]$. Moreover, as $\sigma_1 \geq \sigma_2 \geq \sigma_3$, the worst case complexity indicates that SDNA should be faster than PCDM. We went even further by showing that as the number of processors τ increases, $\sigma_1(\tau) \geq \tau\sigma_1(1) = \tau\sigma_3(1) \geq \sigma_3(\tau)$. In particular, SDNA enjoys superlinear speedup in τ, to be compared with the sublinear speedup of PCDM.

Chapter 3

Coordinate descent methods for saddle point problems

We now turn to convex optimization problems of the form

$$\min_{x\in\mathcal{X}} \; f(x) + g(x) + h(Mx), \tag{3.1}$$

where $x = (x^{(1)}, \ldots, x^{(n)}) \in \mathcal{X} = \mathcal{X}_1 \times \ldots \times \mathcal{X}_n$ is a decision vector composed of n blocks f is a differentiable convex function with coordinate-wise Lipschitz gradients, $g : \mathcal{X} \to \mathbb{R} \cup \{+\infty\}$ and $h : \mathcal{Y} \to \mathbb{R} \cup \{+\infty\}$ are convex and lower semicontinous functions and $M : \mathcal{X} \to \mathcal{Y}$ is a continuous linear operator. We also assume that $\mathcal{Y} = \mathcal{Y}_1 \times \ldots \times \mathcal{Y}_p$ for some integer p.

Under the standard qualification condition $0 \in \mathrm{ri}(M \operatorname{dom} g - \operatorname{dom} h)$ (where dom and ri are the domain and the relative interior), a point $x \in \mathcal{X}$ is a minimizer of (3.1) if and only if there exists $y \in \mathcal{Y}$ such that (x, y) is a saddle point of the Lagrangian function

$$L(x, y) = f(x) + g(x) + \langle y, Mx\rangle - h^\star(y)$$

where $h^\star : y \mapsto \sup_{z\in\mathcal{Y}} \langle y, z\rangle - h(z)$ is the Fenchel-Legendre transform of h.

Problem (3.1) is much more general than Problem (2.1) that we studied in the previous chapter. It allows us to consider nonsmooth nonseparable objectives, containing for instance linear equality and inequality constraints. Indeed, it provides a unified formulation for a broad set of applications in various disciplines, see, e.g., [BT89, BV04, CBS14, CRPW12, MCTD+14, NW06, Wai14]. While problem (3.1) is presented in the unconstrained form, it automatically covers constrained settings by means of indicator functions. For example, (3.1) covers the following prototypical optimization template via $h(z) := \delta_{\{b\}}(z)$ (i.e., the indicator function of the convex set $\{b\}$):

$$\min_{x\in\mathcal{X}} \big\{f(x) + g(x) + \delta_{\{b\}}(Ax)\big\} = \min_{x\in\mathcal{X}} \big\{f(x) + g(x) : Ax = b\big\}, \tag{3.2}$$

Note that (3.2) is sufficiently general to cover standard convex optimization subclasses, such as conic programming, monotropic programming, and geometric programming, as specific instances [BTN01, Ber96, BPC+11b].

In the first section of this chapter, we study a primal dual coordinate descent method based on a fixed point operator. The other sections are devoted to methods based on a smoothing technique. We study parallel coordinate descent for smoothed nonsmooth functions, a generalization of Nesterov's smoothing technique to linear equality constraints and a second primal-dual coordinate descent with a very good worst case guarantee.

3.1 A coordinate descent version of the Vũ Condat method with long step sizes

3.1.1 Introduction

There is a rich literature on *primal-dual* algorithms searching for a saddle point of L (see [TC14] and references therein). In the special case where $f = 0$, the alternating direction method of multipliers

(ADMM) proposed by Glowinsky and Marroco [GM75], Gabay and Mercier [GM76] and the algorithm of Chambolle and Pock [CP11] are amongst the most celebrated ones. Based on an elegant idea also used in [HY12], Vũ [Vũ13] and Condat [Con13] separately proposed a primal-dual algorithm allowing as well to handle ∇f explicitly, and requiring one evaluation of the gradient of f at each iteration. Hence, the ∇f is handled explicitly in the sense that the algorithm does *not* involve, for instance, the call of a proximity operator associated with f. A convergence rate analysis is provided in [CP15a] (see also [TC14]). A related splitting method has been recently introduced by [DY15].

The paper [FB19] introduces a *coordinate descent* (CD) version of the Vũ-Condat algorithm. By coordinate descent, we mean that only a subset of the coordinates of the primal and dual iterates is updated at each iteration, the other coordinates being maintained to their past value. Coordinate descent was historically used in the context of coordinate-wise minimization of a unique function in a Gauss-Seidel sense [War63, BT89, Tse01]. Tseng *et al.* [LT02, TY09a, TY09b] and Nesterov [Nes12a] developped CD versions of the gradient descent. In [Nes12a] as well as in this paper, the updated coordinates are randomly chosen at each iteration. The algorithm of [Nes12a] has at least two interesting features. Not only it is often easier to evaluate a single coordinate of the gradient vector rather than the whole vector, but the conditions under which the CD version of the algorithm is provably convergent are generally weaker than in the case of standard gradient descent. The key point is that the *step size* used in the algorithm when updating a given coordinate i can be chosen to be inversely proportional to the *coordinate-wise* Lipschitz constant of ∇f along its ith coordinate, rather than the global Lipschitz constant of ∇f (as would be the case in a standard gradient descent). Hence, the introduction of coordinate descent allows to use *longer step sizes* which potentially results in a more attractive performance. The random CD gradient descent of [Nes12a] was later generalized by Richtárik and Takáč [RT14] to the minimization of a sum of two convex functions $f + g$ (that is, $h = 0$ in problem (3.1)). The algorithm of [RT14] is analyzed under the additional assumption that function g is *separable* in the sense that for each $x \in \mathcal{X}$, $g(x) = \sum_{i=1}^{n} g_i(x^{(i)})$ for some functions $g_i : \mathcal{X}_i \to]-\infty, +\infty]$.

In the literature, several papers seek to apply the principle of coordinate descent to primal-dual algorithms. In the case where $f = 0$, h is separable and smooth and g is strongly convex, Zhang and Xiao [ZX14] introduce a stochastic CD primal-dual algorithm and analyze its convergence rate (see also [Suz14] for related works). In 2013, Iutzeler et al. [IBCH13] proved that random coordinate descent can be successfully applied to fixed point iterations of firmly non-expansive (FNE) operators. According to [Gab83], the ADMM can be written as a fixed point algorithm of a FNE operator, which led the authors of [IBCH13] to propose a coordinate descent version of ADMM with application to distributed optimization. The key idea behind the convergence proof of [IBCH13] is to establish the so-called stochastic Fejér monotonicity of the sequence of iterates as noted by [CP15b]. In a more general setting than [IBCH13], Combettes *et al.* in [CP15b] and Bianchi *et al.* [BHF14] extend the proof to the so-called α-averaged operators, which include FNE operators as a special case. This generalization allows to apply the coordinate descent principle to a broader class of primal-dual algorithms which is no longer restricted to the ADMM or the Douglas Rachford algorithm. For instance, Forward-Backward splitting is considered in [CP15b] and particular cases of the Vũ-Condat algorithm are considered in [BHF14, PR15]. Nevertheless, the above approach has two major limitations.

First, in order to derive a converging coordinate descent version of a given deterministic algorithm, the latter must write as a fixed point algorithm over some product Hilbert space of the form $H = H_1 \times \cdots H_q$ where the inner product in H is the sum of the inner products in the H_i's. Unfortunately, this condition does *not* hold in general for the Vũ-Condat method, because the inner product over H involves the coupling linear operator M. A workaround was proposed in [BHF14] but for a particular example only.

Second and even more importantly, the approach of [IBCH13, CP15b, BHF14, PR15] needs "small" step sizes. More precisely, the convergence conditions are identical to the ones of the brute method, the one without coordinate descent. These conditions involve the global Lipschitz constant of the gradient ∇f instead than its coordinate-wise Lipschitz constants. In practice, it means that the application of coordinate descent to primal-dual algorithm as suggested by [CP15b] and [BHF14] is restricted to the use of potentially small step sizes. One of the major benefits of coordinate descent is lost.

Some recent works also focused on designing primal-dual coordinate descent methods with a guaranteed convergence rate. In [GXZ19] and [CERS18], a $O(1/k)$ rate is obtained for the ergodic mean of the sequences. The rates are given in terms of feasibility and optimality or Bregman distance. Those two papers require all the dual variables to be updated at each iteration, which may not be efficient if there are more than a few dual variables. In the present paper, we will have much more flexibility in the

variables we choose to update at each iteration, while retaining a provable convergence rate.

Our main contribution is to provide a CD primal-dual algorithm with a broad range of admissible step sizes. Our numerical experiments show that remarkable performance gains can be obtained when using larger step sizes. We also identify two setups for which the structure of the problem is favorable to coordinate descent algorithms. Finally, we prove a sublinear rate of convergence in general and a linear rate of convergence if the objective enjoys strong convexity properties.

3.1.2 Main algorithm and convergence theorem

Consider Problem (3.1). We note $M = (M_{j,i} : j \in \{1,\dots,p\},\, i \in \{1,\dots,n\})$ where $M_{j,i} : \mathcal{X}_i \to \mathcal{Y}_j$ are the block components of M. For each $j \in \{1,\dots,p\}$, we introduce the set $I(j) := \Big\{ i \in \{1,\dots,n\} \,:\, M_{j,i} \neq 0 \Big\}$. Otherwise stated, the jth component of vector Mx only depends on x through the coordinates $x^{(i)}$ such that $i \in I(j)$. We denote by $m_j := \mathrm{card}(I(j))$ the number of such coordinates. Without loss of generality, we assume that $m_j \neq 0$ for all j. We also denote $\pi_j := \frac{1}{\mathrm{card}(I(j))}$.

For all $i \in \{1,\dots,n\}$, we define $J(i) := \Big\{ j \in \{1,\dots,p\} \,:\, M_{j,i} \neq 0 \Big\}$. Note that for every pair (i,j), the statements $i \in I(j)$ and $j \in J(i)$ are equivalent.

Let $\sigma = (\sigma_1,\dots,\sigma_p)$ and $\tau = (\tau_1,\dots,\tau_n)$ be two tuples of positive real numbers. Consider an independent and identically distributed sequence $(i_k : k \in \mathbb{N}^*)$ with uniform distribution on $\{1,\dots,n\}$ (the results of this paper easily extend to the selection of several primal coordinates at each iteration with a uniform samplings of the coordinates, using the techniques introduced in [RT15]). The proposed primal-dual coordinate descent algorithm consists in updating two sequences $x_k \in \mathcal{X}$, $y_k \in \mathcal{Y}$. It is provided in Algorithm 6 below.

Algorithm 6 Coordinate-descent primal-dual algorithm

Initialization: Choose $x_0 \in \mathcal{X}$, $y_0 \in \mathcal{Y}$.

Iteration k: Define:

$$\begin{aligned}
\overline{y}_{k+1} &= \mathrm{prox}_{\sigma,h^\star}\left(y_k + D(\sigma)Mx_k\right)\\
\overline{x}_{k+1} &= \mathrm{prox}_{\tau,g}\Big(x_k - D(\tau)\left(\nabla f(x_k) + 2M^\star \overline{y}_{k+1} - M^\star y_k\right)\Big)\,.
\end{aligned}$$

For $i = i_{k+1}$ and for each $j \in J(i_{k+1})$, update:

$$\begin{aligned}
x_{k+1}^{(i)} &= \overline{x}_{k+1}^{(i)}\\
y_{k+1}^{(j)} &= y_k^{(j)} + \pi_j(\overline{y}_{k+1}^{(j)} - y_k^{(j)})\,.
\end{aligned}$$

Otherwise, set $x_{k+1}^{(i')} = x_k^{(i')}$, and $y_{k+1}^{(j')} = y_k^{(j')}$.

Our convergence result holds under the following assumptions.

Assumption 2. *1. The functions f, g, h are closed proper and convex.*

2. The function f is differentiable on $\mathcal{X}$.

3. For every $i \in \{1,\dots,n\}$, there exists $\beta_i \geq 0$ such that for any $x \in \mathcal{X}$, any $u \in \mathcal{X}_i$,

$$f(x + U_i u) \leq f(x) + \langle \nabla f(x), U_i u\rangle + \frac{\beta_i}{2}\|u\|_{\mathcal{X}_i}^2\,.$$

4. The random sequence $(i_k)_{k\in\mathbb{N}^*}$ is independent, uniformly distributed on $\{1,\dots,n\}$.

5. The step sizes $\tau = (\tau_1,\dots,\tau_n)$ and $\sigma = (\sigma_1,\dots,\sigma_p)$ satisfy for all $i \in \{1,\dots,n\}$,

$$\tau_i < \frac{1}{\beta_i + \rho\left(\sum_{j\in J(i)}(2-\pi_j)m_j\sigma_j M_{j,i}^\star M_{j,i}\right)}\,.$$

We denote by $\mathcal{S}$ the set of saddle points of the Lagrangian function L. Otherwise stated, a couple $(x_*, y_*) \in \mathcal{X} \times \mathcal{Y}$ lies in $\mathcal{S}$ if and only if it satisfies the following inclusions

$$0 \in \nabla f(x_*) + \partial g(x_*) + M^\star y_* \tag{3.3}$$
$$0 \in -M x_* + \partial h^\star(y_*)\,. \tag{3.4}$$

We shall also refer to elements of $\mathcal{S}$ as primal-dual solutions.

Theorem 6. *Let Assumption 2 hold true and suppose that $\mathcal{S} \neq \emptyset$. Let (x_k, y_k) be a sequence generated by Algorithm 6. Almost surely, there exists $(x_*, y_*) \in \mathcal{S}$ such that*

$$\lim_{k\to\infty} x_k = x_*$$
$$\lim_{k\to\infty} y_k = y_*\,.$$

In order to prove this result, we introduced the concept of duplication of dual variables. Indeed, as noted in the introduction, the squared norm with which distances are measured in Vũ and Condat's proof is not separable. This poses a major difficulty in the analysis of coordinate descent, that we circumvented by allowing nonseparable objective functions. When $m_j > 1$, i.e. the dual variable $y^{(j)}$ influences several primal variables $\{x^{(i)}, i \in I(j)\}$, we artificially duplicate the dual variable $y^{(j)}$ into a vector of size m_j denoted by $(y(j)(i))_{i\in I(j)}$. In order to get an equivalent problem, we replace h by $\tilde{h}$ that forces the constraint $y^{(j)}(i) = y^{(j)}(i')$ for all $i, i' \in I(j)$. In this equivalent problem, $m_j = 1$ for all j and we can define a separable squared norm. This allows us to prove the convergence of the algorithm with the duplicated dual space and then Theorem 6 by showing that both algorithm are in fact the same.

Efficient implementation using problem structure In Algorithm 6, it is worth noting that quantities $(\overline{x}_{k+1}, \overline{y}_{k+1})$ do not need to be explicitly calculated. At iteration k, only the coordinates

$$\overline{x}_{k+1}^{(i_{k+1})} \text{ and } \overline{y}_{k+1}^{(j)}, \quad \forall j \in J(i_{k+1})$$

are needed to perform the update. From a computational point of view, it is often the case that the evaluation of the above coordinates is less demanding than the computation of the whole vectors $\overline{x}_{k+1}$, $\overline{y}_{k+1}$. Two situations have been reported in the literature:

- If g is separable, one only needs to compute the quantities $\nabla_{i_{k+1}} f(x_k)$, $(2M^\star \bar{y}_{k+1} - M^\star y_k)^{(i_{k+1})}$ and $\operatorname{prox}_{\tau_{i_{k+1}}, g_{i_{k+1}}}$ to perform the kth iteration. A classical example of such smart residual update [Nes12b] can be found in the proximal coordinate descent gradient algorithm (case g separable and $h = 0$) [RT14]. More generally, if g (resp. $h^\star$) is block-separable, we can use this structure in the algorithm, even if this block structure does not match $\mathcal{X}_1 \times \ldots \times \mathcal{X}_n$ (resp. $\mathcal{Y}_1 \times \ldots \times \mathcal{Y}_p$).

  We used this idea in Figure 3.1 to deal efficiently with the proximal operator of the $\ell_{2,1}$ norm.

- If g is the indicator of the consensus constraint $\{x_1 = \cdots = x_n\}$, f is separable and $h = 0$, we recover MISO [Mai15]. In that case, we can store $\nabla f(x_k)$ and update its average. Thanks to the separability of f, only one coordinate of $\nabla f(x_k)$ needs to be updated at each iteration.

  We used similar ideas in Figure 3.2 to deal efficiently with the projection onto the subspace orthogonal to a vector.

To illustrate the importance of these implementation tricks, we give in Table 3.1 a comparison of the number of operations to compute the updates of the standard Vũ-Condat method against the proposed algorithm.

3.1.3 Convergence rate

We present below the convergence guarantee for Algorithm 6. It is of the same order as what can be obtained by other primal-dual methods like the ADMM [DY17], *i.e.* $O(1/\sqrt{k})$ in general. Note that the result holds for the sequence $(\bar{x}_k)$ only, since x_k may not be feasible by design.

Problem / Dimension of data	Vũ-Condat	Our algorithm
Total Variation + ℓ_1 regularization $A \in \mathbb{R}^{m\times n}$: dense; $M \in \mathbb{R}^{3n\times n}$: nnz$(M) = 6n$	$O(mn + 6n)$	$O(m + 12)$
Support Vector Machines $A \in \mathbb{R}^{m\times n}$: sparse	$O(\text{nnz}(A) + n)$	$O(\text{nnz}(Ae_i) + 1)$

Table 3.1: Number of operations per iteration for the proposed algorithm and for the standard Vũ-Condat algorithm - The use cases are the ones described in the numerical section. The numbers 6 and 12 highlight the (mild) overhead of duplication in the Total Variation + ℓ_1 regularized least squares problem.

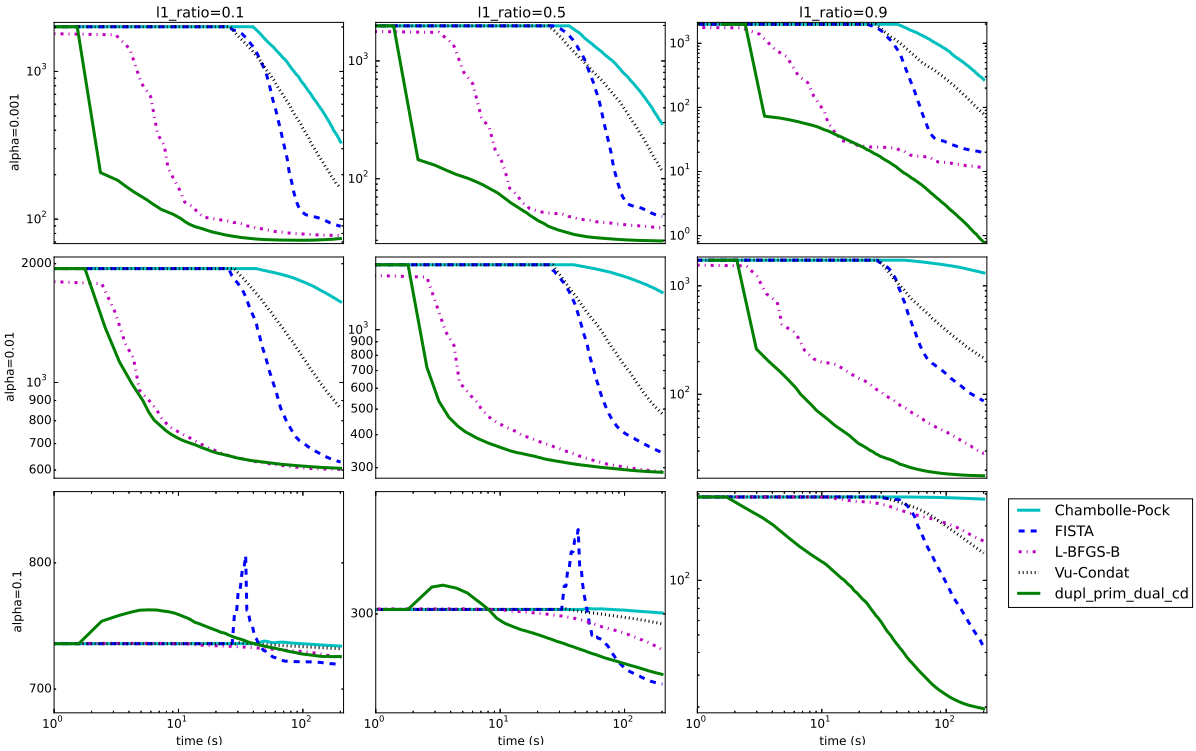

Figure 3.1: Comparison of algorithms for TV+L_1-regularized regression at various regularization parameters. The problem is given by $\min_{x\in\mathbb{R}^n} \frac{1}{2}\|Ax - b\|_2^2 + \alpha\big(r\|x\|_1 + (1-r)\|Mx\|_{2,1}\big)$ and the data comes from fMRI data. Our method is `dupl_prim_dual_cd` (solid green curve). Note for the choices of regularization parameters such that $\alpha(1-r)$ is larger, the problem is more difficult to solve because the total variation regularizer is dominant. This is in fact the most challenging part of the objective because it is non-differentiable and non-separable.

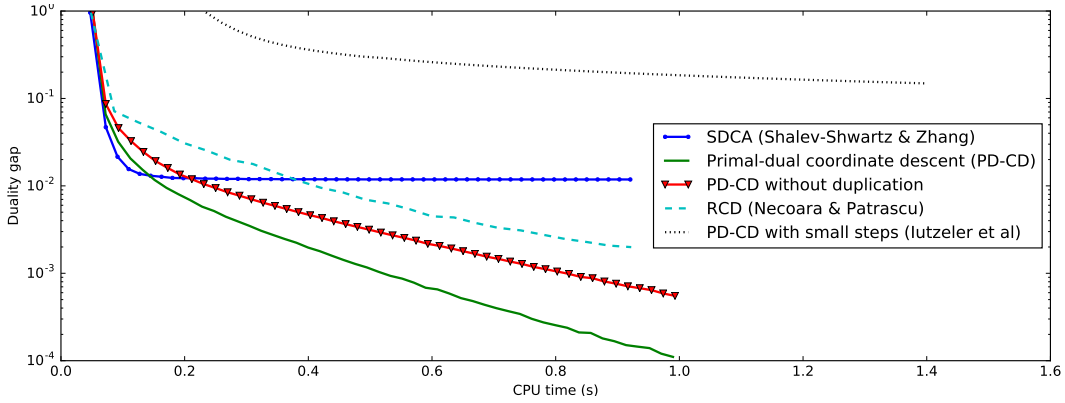

Figure 3.2: Comparison of algorithms for the resolution of the dual of linear SVM $\max_{x\in\mathbb{R}^n} -\frac{1}{2\lambda}\|AD(b)x\|_2^2 + e^T x - \sum_{i=1}^n I_{[0,C_i]}(x_i) - I_{\{0\}}(\langle b, x\rangle)$ on the RCV1 dataset. We report the value of the duality gap after a post-processing to recover feasible primal and dual variables. We stopped each algorithm after 100 passes through the data: note that the cost per iteration of the 5 algorithms is similar but that the algorithm of [IBCH13] needs first to compute the Lipschitz constant of the gradient. SDCA does not converge to the minimum because it does not consider the linear equality constraint.

Theorem 7. *Define for $\alpha \geq 1$,*

$$C_{1,\alpha} = \max_{1\leq i\leq n} \frac{\tau_i^{-1} + \tau_i^{-1/2}\rho(\sum_{j\in J(i)} m_j\sigma_j M^\star_{j,i}M_{j,i})^{1/2}}{\tau_i^{-1} - \rho(\sum_{j\in J(i)} m_j\sigma_j M^\star_{j,i}M_{j,i})}(1+\frac{n}{\alpha})$$

$$C_{2,\alpha} = \left(1 + \max_{1\leq i\leq n} \frac{\alpha^{-1}(n(n-1)+1)+1}{\tau_i^{-1} - \beta_i - \rho(\sum_{j\in J(i)}(2-\boldsymbol{\pi}_j(i))m_j\sigma_j M^\star_{j,i}M_{j,i})}\beta_i\right).$$

We have that $C_{1,\alpha}$ and $C_{2,\alpha}$ are nonincreasing with respect to α, and thus bounded. Let us denote $S_{0,*} = f(x_0) - f(x_*) - \langle\nabla f(x_*), x_0 - x_*\rangle$ and $V(z) = V(x,y) = \frac{1}{2}\|x\|^2_{\tau^{-1}} + \langle Mx, y\rangle + \frac{1}{2}\|y\|^2_{\sigma^{-1}}$.

Define the number of iterations $\hat{K} \in \{1,\dots,k\}$ as a random variable, independent of $\{i_1,\dots,i_k\}$ and such that $\Pr(\hat{K} = l) = \frac{1}{k}$ for all $l \in \{1,\dots,k\}$.

If h is $L(h)$-Lipschitz in the norm $\|\cdot\|_{D(m)\sigma}$, then for all $k \geq 0$,

$$\mathbb{E}(f(\bar{x}_{\hat{K}}) + g(\bar{x}_{\hat{K}}) + h(M\bar{x}_{\hat{K}}) - f(x_*) - g(x_*) - h(Mx_*)) \leq \frac{C_{2,\sqrt{k}} + 2C_{1,k}}{\sqrt{k}} n(S_{0,*} + V(z_0 - z_*)) + \frac{4}{\sqrt{k}}L(h)^2.$$

If $h = I_{\{b\}}$, then for all $k \geq 0$,

$$\mathbb{E}(f(\bar{x}_{\hat{K}}) + g(\bar{x}_{\hat{K}}) - f(x_*) - g(x_*)) \leq \frac{C_{2,\sqrt{k}} + 2C_{1,k}}{\sqrt{k}} n\big(S_{0,*} + V(z_0 - z_*)\big) + \|y_*\|\mathbb{E}(\|M\bar{x}_{\hat{K}} - b\|)$$

$$\mathbb{E}(\|M\bar{x}_{\hat{K}} - b\|_{D(m)\sigma}) \leq \frac{2}{\sqrt{k}}\Big(\sqrt{C_{2,\sqrt{k}} + 2C_{1,k}} + \sqrt{2C_{1,k}}\Big)\big(n(S_{0,*} + V(z_0 - z_*))\big)^{1/2}\Big)$$

3.2 Smooth minimization of nonsmooth functions with parallel coordinate descent methods

A major question when designing a Parallel Coordinate Descent Method (PCDM) is: how should we combine the updates computed by the various processors? One may simply compute the updates in the same way as in the single processor case, and apply them all. However, this strategy is doomed to fail: the method may end up oscillating between sub-optimal points [TBRS13]. Indeed, although the individual updates are safe, there is no reason why adding them all up for should decrease the

function value. In order to overcome this difficulty, Richtárik and Takáč [RT15] introduced the concept of Expected Separable Overapproximation (ESO). Thanks to this bound on the expected decrease after one iteration of the algorithm, they could define safe values for the amount of damping one should apply to the updates in order to have a converging algorithm.

They could prove a nearly linear theoretical parallelization speedup for a composite and partially separable function. This means that the objective function is the sum of a separable nonsmooth function and a differentiable function of the form $f(x) = \sum_{J\in\mathcal{J}} f_J(x^{(J)})$ where each function f_J depends only on a small number of coordinates $x^{(J)}$. They also showed that the way coordinates are sampled has a huge impact on the performance. Indeed, PCDM implemented with a so-called τ-nice sampling can be faster than PCDM implemented with a more general uniform sampling by a factor $O(\sqrt{n})$, where n is the number of variables.

The goal of the paper [FR17] is to study a class of nonsmooth functions on which similar parallelization speedups can be proved for parallel coordinate descent methods. This class of functions will be the class of convex functions with max structure, which is very closely related to saddle points problems. Our approach is based on the smoothing technique introduced by Nesterov in [Nes05b]. Indeed, if the function to optimize has a max-structure, then one can define a smooth approximation of the function and minimize the approximation by any method available for smooth optimization, including coordinate descent.

We wish to solve the problem

$$\min_x g(x) + h(Ax) = \min_{x\in\mathbb{R}^N} \max_{z\in Q} g(x) + \langle Ax, z\rangle - h^*(z), \tag{3.5}$$

where we assume that Q is bounded. Such problems were coined by Nesterov as problems with max structure in [Nes05b]. The method we use for solving the smoothed composite problem (3.5) is given in Algorithm 7. It relies on a smooth approximation of $h(Ax)$ given by [Nes05b]

$$h_\mu(y) = \max_{z\in Q}\{\langle y, z\rangle - h^*(z) - \mu d(z)\}, \tag{3.6}$$

where d is a function defined on Q, called the prox function, which is strongly convex with respect to the norm $\|z\|_v = \left(\sum_{j=1}^m v_j^p |z_j|^p\right)^{1/p}$.

Algorithm 7 Smoothed Parallel Coordinate Descent Method (SPCDM)

Input: initial iterate $x_0 \in \mathbb{R}^N$, $\beta > 0$ and $w = (w_1, \dots, w_n) > 0$

for $k \geq 0$ **do**

Step 1. Generate a random set of blocks $S_k \subseteq \{1, 2, \dots, n\}$ following the law of $\hat{S}$

Step 2. In parallel for $i \in S_k$, compute

$$s_k^{(i)} = \arg\min_{t\in\mathbb{R}^{N_i}} \left\{ \langle (A^\top \nabla h_\mu(Ax_k))^{(i)}, t\rangle + \frac{\beta w_i}{2}\langle B_i t, t\rangle + g_i(x_k^{(i)} + t) \right\}$$

Step 3. In parallel for $i \in S_k$, update $x_{k+1}^{(i)} \leftarrow x_k^{(i)} + s_k^{(i)}$ and set $x_{k+1}^{(j)} \leftarrow x_k^{(j)}$ for $j \notin S_k$

end for

On top of the random sampling, the algorithm depends on parameters $\beta > 0$ and $w \in \mathbb{R}^n_+$. These parameters are determined in such a way that the function $H_\mu = h_\mu \circ A$ satisfies an *Expected Separable Overapproximation* (ESO) defined for a function ϕ as

$$\mathbb{E}\left[\phi(x + s_{[\hat{S}]})\right] \leq \phi(x) + \frac{\mathbb{E}[|\hat{S}|]}{n}\left(\langle \nabla\phi(x), s\rangle + \frac{\beta}{2}\sum_{i=1}^n w_i \langle B_i s^{(i)}, s^{(i)}\rangle\right), \qquad x, s \in \mathbb{R}^N, \tag{3.7}$$

where $s_{[\hat{S}]}$ is defined in Assumption 1. When (3.7) holds, we say that ϕ admits a (β, w)-ESO with respect to $\hat{S}$. For simplicity, we may sometimes write $(\phi, \hat{S}) \sim \mathrm{ESO}(\beta, w)$.

It is important to understand whether choosing $\tau > 1$ (several processors), as opposed to $\tau = 1$ (one single processor), leads to acceleration in terms of an improved complexity bound. By analogy with proximal gradient descent, we can see that $\frac{1}{\beta}$ can be interpreted as a stepsize. We would hence wish to choose small β, but not too small so that the method does not diverge. The issue of the computation

Table 3.2: Summary of iteration complexity results

	strong convexity	convexity
Nonsmooth problem with max-structure Problem (3.5)	$\frac{n}{\tau} \times \frac{\frac{2\beta(\tau)D}{\epsilon\sigma}+\sigma_\Psi}{\sigma_{f_\mu}+\sigma_\Psi} \times \log(\frac{1}{\epsilon})$	$\frac{n\beta(\tau)}{\tau} \times \frac{8DDiam^2}{\sigma\epsilon^2} \times \log(\frac{1}{\epsilon})$

of a good (small) parameter β is very intricate for several reasons and is at the heart of the design of a randomized parallel coordinate descent method. As can be seen from Table 3.2, the number of iterations required to obtain an ϵ-solution for Problem (3.5) is of the form $k \geq C(\epsilon)\frac{\beta(\tau)}{\tau}$, where $C(\epsilon)$ does not depend on τ. Hence, parallelization speedup occurs when the function $T(\tau) = \frac{\beta(\tau)}{\tau}$ is decreasing. This has been proved for smooth partially separable functions in [RT15]. In [FR17], we proved it for Nesterov separable functions.

Definition 2 (Nesterov separability)**.** *We say that $H = h \circ A$ is* Nesterov (block) separable *of degree ω if h has the form* (3.6) *and*

$$\max_{1\leq j\leq m} |\{i \ : \ A_{ji} \neq 0\}| \leq \omega. \tag{3.8}$$

Theorem 8 (ESO for τ-nice sampling)**.** *Let $H = h \circ A$ be Nesterov separable of degree ω, $\hat{S}$ be τ-nice (i.e. for all $S \subset \{1, \dots n\}$, if $|S| = \tau$, then $\mathbb{P}(\hat{S} = S) = 1/\binom{n}{\tau}$), and w^* be chosen as*

$$w_i^* = \max\{(\|A_i t\|_v^*)^2 \ : \ t \in \mathbb{R}^{N_i}, \ \|t\|_E = 1\}, \quad i = 1, 2, \dots, n.$$

Then

$$(H_\mu, \hat{S}) \sim \mathrm{ESO}(\beta, w^*),$$

where $\beta = \frac{\beta'}{\mu\sigma}$ and, if the dual norm $\|\cdot\|_v$ is defined with $p = 2$,

$$\beta' = \beta_2' = 1 + \frac{(\omega - 1)(\tau - 1)}{\max(1, n - 1)}$$

or, if $p = 1$,

$$\beta' = \beta_3' = \sum_{k=1}^{k_{max}} \min\left\{1, \frac{mn}{\tau} \sum_{l=\max\{k,k_{min}\}}^{k_{max}} c_l \pi_l\right\} \tag{3.9}$$

where c_l, π_l, k_{min} and k_{max} are defined by: $k_{min} = \max\{1, \tau - (n - \omega)\}$, $k_{max} = \min\{\tau, \omega\}$, $c_l = \max\left\{\frac{l}{\omega}, \frac{\tau - l}{n - \omega}\right\} \leq 1$ if $\omega < n$, $c_l = \frac{l}{\omega} \leq 1$ otherwise, and $\pi_l = \frac{\binom{\omega}{k}\binom{n-\omega}{\tau-k}}{\binom{n}{\tau}}, k_{min} \leq l \leq k_{max}$.

Formula (3.9) may look complicated at first glance but it is in fact just a sum of a few easily computable terms. Computing β_3' has a negligible cost compared to the rest of the algorithm.

One can also easily show that using $\beta = \beta_1' = \min\{\omega, \tau\}$ always leads to a converging algorithm for any τ-uniform sampling. However, the performance may be disappointing for an algorithm using parallel processing. In Figure 3.3, we can see that our analysis allows us to prove a much better theoretical parallelization speedup than the more basic result thanks to a fine study of τ-nice samplings.

3.3 A Smooth Primal-Dual Optimization Framework for Nonsmooth Composite Convex Minimization

3.3.1 Introduction

In [TDFC18], we introduce a new analysis framework for designing primal-dual optimization algorithms to obtain numerical solutions to (3.1). We first focused on the case $f = 0$. Associated with the primal problem (3.1), we define the following dual formulation:

$$\max_{y\in\mathcal{Y}} \Big\{D(y) := -g^*(-A^\top y) - h^*(y)\Big\}, \tag{3.10}$$

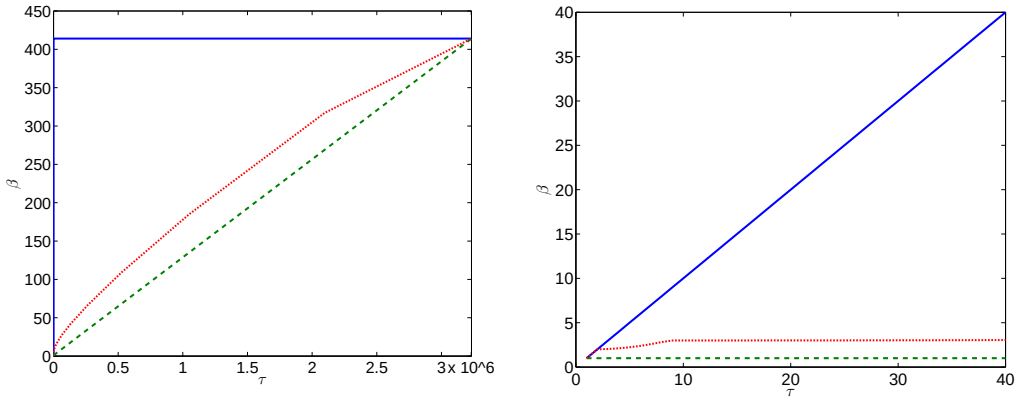

Figure 3.3: Comparison of three formulae for β' as a function of the number of processors τ (smaller β' is better). **Left:** Large number of processors. **Right:** Zoom for smaller number of processors. We have used matrix $A \in \mathbb{R}^{m\times n}$ with $m = 2,396,130$, $n = 3,231,961$ and $\omega = 414$. **Blue solid line:** τ-uniform sampling, $\beta'_1 = \min\{\omega, \tau\}$. **Green dashed line:** τ-nice sampling and $p = 2$, $\beta'_2 = 1 + \frac{(\omega-1)(\tau-1)}{\max\{1,n-1\}}$ (Theorem 8). **Red dash-dotted line:** τ-nice sampling and $p = 1$, β'_3 follows (3.9) in Theorem 8. Note that β'_1 reaches its maximal value ω quickly, whereas β'_2 increases slowly. When τ is small compared to n, this means that β'_2 remains close to 1. Recall that small values of β' directly translate into better complexity and parallelization speedup.

where g^* and h^* are the Fenchel conjugate of g and h, respectively.

Among classical convex optimization methods, the primal-dual approach is perhaps one of the best candidates to solve the primal-dual pair (3.1)-(3.10). Theory and methods along this approach have been developed for several decades and have led to a diverse set of algorithms. A comparison between some famous primal-dual methods and our approach in this paper is given in Tables 3.3 and 3.4. There are several reasons for our emphasis on first-order primal-dual methods for (3.1)-(3.10), with the most obvious one being their scalability. Coupled with recent demand for low-to-medium accuracy solutions in applications, these methods indeed provide important trade-offs between the per-iteration complexity and the iteration-convergence rate along with the ability to distribute and decentralize the computation.

Associated with the primal problem (3.1) and the dual one (3.10), we define

$$G(w) := P(x) - D(y), \tag{3.11}$$

as a primal-dual gap function, where $w := (x, y)$ is the concatenated primal-dual variable. The gap function G in (3.11) is convex in terms of w. Under strong duality, we have $G(w^\star) = 0$ if and only if $w^\star = (x^\star, y^\star)$ is a primal-dual solution of (3.1) and (3.10).

In stark contrast with the existing literature, our analysis relies on a novel combination of three classical concepts in convex optimization applied to the primal-dual gap function: Nesterov's *smoothing* technique, the *accelerated* proximal gradient descent method and *homotopy* in a nontrivial manner. While some combinations of these techniques have already been studied in the literature, their full combination is important for the desiderata and had not been studied before.

Smoothing: We can obtain a smoothed estimate of the gap function within Nesterov's smoothing technique applied to g and h [BT12, Nes05b]. In the sequel, we denote the smoothed gap function by $G_{\gamma\beta}(w) := P_\beta(x) - D_\gamma(y)$ to approximate the primal-dual gap function $G(w)$, where P_β is a smoothed approximation to P depending on the smoothness parameter $\beta > 0$, and D_γ is a smoothed approximation to D depending on the smoothness parameter $\gamma > 0$. By smoothed approximation, we mean the same max-form approximation as [Nes05b]. However, it was previously unclear how to properly update these smoothness parameters in primal-dual methods.

Acceleration: Using an accelerated scheme, we will design new primal-dual decomposition methods that satisfy the following smoothed gap reduction model:

$$G_{\gamma_{k+1}\beta_{k+1}}(\bar{w}^{k+1}) \le (1 - \tau_k) G_{\gamma_k\beta_k}(\bar{w}^k) + \ \psi_k, \tag{3.12}$$

where $\{\bar{w}^k\}$ and the parameters are generated by the algorithms with $\tau_k \in [0, 1)$ and $\max(\psi_k, 0)$ converges

Table 3.3: A comparison of convergence rates between ASGARD and ADSGARD algorithms (in the case $f = 0$) and selected existing methods for solving (3.1) and (3.2). Here, all algorithms do not involve any large matrix inversion or "complex" convex subproblem, and $w_k := \frac{1}{k}\sum_{l=1}^{k} w^l$; K is the iteration budget; and σ is the step-size in [CP11].

Paper	dom g bounded and h Lipschitz	h Lipschitz	$h = \delta_{\{c\}}$ (optimality and feasibility)
Nesterov [Nes05b]	$P(x^k) \leq \frac{2\sqrt{\bar{L}_A D_{\mathcal{X}} D_{\mathcal{Y}}}}{K}\left(1+\frac{K^2}{k^2}\right)$	$P(x^k) \leq \frac{\epsilon}{2} + \frac{8\bar{L}_A \|x^0 - x^\star\|^2 D_{\mathcal{Y}}}{\epsilon(k+1)^2}$	not applicable
Chambolle-Pock [CP11]	$G(w_k) \leq \frac{\sigma \bar{L}_A D_{\mathcal{X}} + \sigma^{-1} D_{\mathcal{Y}}}{k}$	convergence	convergence
ASGARD [TDFC18]	$P(x^k) \leq \frac{2\sqrt{2}\sqrt{\bar{L}_A D_{\mathcal{Y}} D_{\mathcal{X}}}}{k}$	$P(x^k) \leq \frac{\bar{L}_A}{2\beta_1 k}\|\bar{x}^0 - x^\star\|^2 + \frac{2\beta_1}{k} D_{\mathcal{Y}}$	$\lvert g(x^k) - g^\star\rvert \leq$ $\frac{\bar{L}_A}{\beta_1 k}\|\bar{x}^0 - x^\star\|^2 + \frac{3\beta_1}{k}\|\dot{y} - y^\star\|^2 + \frac{\beta_1}{k}\|y^\star\|^2$ $\|Ax^k - c\|_{\mathcal{Y},*} \leq$ $\frac{\beta_1}{k+1}\left(2\|\dot{y} - y^\star\| + \frac{\sqrt{\bar{L}_A}}{\beta_1}\|\bar{x}^0 - x^\star\|\right)$
ADSGARD [TDFC18]	$G(w^k) \leq \frac{2\sqrt{\bar{L}_A D_{\mathcal{Y}} D_{\mathcal{X}}}}{k}$	$G(w^k) \leq \frac{\gamma_1}{k+1}\|\dot{x} - x^\star\|^2 + \frac{2\bar{L}_A}{\gamma_1 k} D_{\mathcal{Y}}$	$\lvert g(x^k) - g^\star\rvert \leq$ $\frac{3\gamma_1}{k}\|\dot{x} - x^\star\|^2 + \frac{2\bar{L}_A}{\gamma_1 k}\|\dot{y} - y^\star\|^2 + \frac{L_A}{\gamma_1 k}\|y^\star\|^2$ $\|Ax^k - c\|_{\mathcal{Y},*} \leq$ $\frac{\bar{L}_A}{\gamma_1 k}\left(2\|\dot{y} - y^\star\| + 2\gamma_1\|\dot{x} - x^\star\|\right)$

to zero. Similar ideas have been proposed before; for instance, Nesterov's excessive gap technique [Nes05a] is a special case of the gap reduction model (3.12) when $\psi_k \leq 0$ (see [TDC14]).

Homotopy: We will design algorithms to maintain (3.12) while simultaneously updating β_k, γ_k and τ_k to zero to achieve the best known convergence rate based on the assumptions imposed on the problem template. This strategy will also allow our theoretical guarantees not to depend on the diameter of the feasible set of (3.2). A similar technique is also proposed in [Nes05a], but only for symmetric primal-dual methods. It is also used in conjunction with Nesterov's smoothing technique in [BH12] for unconstrained problem but had only an $\mathcal{O}(\ln(k)/k)$ convergence rate.

Note that without homotopy, we can directly apply Nesterov's accelerated methods to minimize the smoothed gap function $G_{\gamma\beta}$ for given $\gamma > 0$ and $\beta > 0$. In this case, these smoothness parameters must be fixed a priori depending on the desired accuracy and the prox-diameter of both the primal and dual problems, which may not be applicable to (3.2) due to the unboundedness of the dual feasible domain.

Table 3.4: A comparison of convergence rates between our algorithms and selected existing methods for solving (3.2). Here, all algorithms may involve "complex" convex subproblems or matrix inversions; and ρ is the penalty parameter in [LMon] and [MS12].

Paper	$g = \delta_{\{c\}}$
ALM [LMon]	$\lvert g(x^k) - g^\star\rvert \leq \frac{6}{\rho\sqrt{k}}\|y^\star\|\|y^0 - y^\star\|$ $\|Ax^k - c\|_{\mathcal{Y},*} \leq \frac{3}{\rho\sqrt{k}}\|y^0 - y^\star\|$
ADMM [MS12]	$\lvert g_1((x_1)_k) + g_2((x_2)_k) - g_1(x_1^\star) - g_2(x_2^\star)\rvert \leq \frac{6+4\sqrt{2}}{k}\left(\frac{1}{\rho}\|y^0 - y^\star\|^2 + \rho\|x_1^0 - x_1^\star\|^2_{A_1^* A_1}\right)$ $\|A_1(x_1)_k + A_2(x_2)_k - c\| \leq \frac{2}{k}\sqrt{\frac{1}{\rho^2}\|y^0 - y^\star\|^2 + \|x_1^0 - x_1^\star\|^2_{A_1^* A_1}}$
ASALGARD [TDFC18]	$\lvert g(x^k) - g^\star\rvert \leq \frac{10\|y^\star\|_{\mathcal{Y}}\|\dot{y} - y^\star\|_{\mathcal{Y}}}{\gamma_0(k+1)^2}$ $\|Ax^k - c\|_{\mathcal{Y},*} \leq \frac{8\|\dot{y} - y^\star\|_{\mathcal{Y}}}{\gamma_0(k+2)^2}$

3.3.2 Technical results

Lemma 3 is fundamental to our theory since it relates the decrease in the smoothed gap to the objective value and the feasibility gap in the very important case of equality constraints. It shows that when the smoothed gap and the smoothing parameters are small at the same time, then objective value gap and feasibility gap are both small.

This question had been open for several years in the area of convergence rates for primal-dual methods. Previous works either made assumptions that forbade equality constraints or proved results in terms of quantities that are not natural measures of optimality (for instance the restricted duality gap in [CP11]). Concurrently to our work, alternative approaches have been proposed by Y. Drori in [Dro14] and by Y. Xu in [Xu17]. Note that although the rates obtained are the same, our approach using smoothing is more principled.

Lemma 3. *Let $G_{\gamma\beta}$ be the smoothed gap function and $S_\beta(x;\dot{y}) := P_\beta(x;\dot{y}) - P(x^\star) = f(x) + g_\beta(Ax;\dot{y}) - P(x^\star)$ be the smoothed objective residual. Then, we have*

$$S_\beta(x;\dot{y}) \leq G_{\gamma\beta}(w;\dot{w}) + \gamma b_{\mathcal{X}}(x^\star, \dot{x}) \quad \text{and} \quad \frac{1}{2}\|y^*_\beta(Ax;\dot{y}) - y^\star\|^2_{\mathcal{Y},*} \leq b_{\mathcal{Y}}(y^\star,\dot{y}) + \frac{1}{\beta}S_\beta(x;\dot{y}). \tag{3.13}$$

Suppose that $g(\cdot) := \delta_{\{c\}}(\cdot)$. Then, for any $y^\star \in y^\star$ and $x \in \mathcal{X}$, one has

$$-\|y^\star\|_{\mathcal{Y}}\|Ax - c\|_{\mathcal{Y},*} \leq f(x) - f(x^\star) \tag{3.14}$$

and the following primal objective residual and feasibility gap estimates hold for (3.2)*:*

$$\begin{cases} f(x) - f(x^\star) & \leq S_\beta(x;\dot{y}) - \langle y^\star, Ax - c\rangle + \beta b_{\mathcal{Y}}(y^\star,\dot{y}), \\ \|Ax - c\|_{\mathcal{Y},*} & \leq \beta L_{b_{\mathcal{Y}}}\Big[\|y^\star - \dot{y}\|_{\mathcal{Y}} + \big(\|y^\star - \dot{y}\|^2_{\mathcal{Y}} + 2L^{-1}_{b_{\mathcal{Y}}}\beta^{-1}S_\beta(x;\dot{y})\big)^{1/2}\Big], \end{cases} \tag{3.15}$$

where the quantity in the square root is always nonnegative.

Using the smoothed gap machinery, we defined a new algorithm, that we called the Accelerated Smoothed GAp ReDuction algorithm (ASGARD). The idea is at each iteration to run one step of accelerated gradient on the smoothed gap function and then to decrease the smoothing parameter β_k as much as our analysis allows us to do while maintaining (3.12).

$$\begin{cases} \hat{x}^k & = (1-\tau_k)\bar{x}^k + \tau_k\tilde{x}^k, \\ y^*_{\beta_{k+1}}(A\hat{x}^k;\dot{y}) & := \arg\max_{y\in\mathcal{Y}} \langle A\hat{x}^k, y\rangle - h^*(y) - \beta_{k+1}b_{\mathcal{Y}}(y,\dot{y}), \\ \bar{x}^{k+1} & = \operatorname{prox}_{\beta_{k+1}\bar{L}_A^{-1}g}\left(\hat{x}^k - \beta_{k+1}\bar{L}_A^{-1}A^\top y^*_{\beta_{k+1}}(A\hat{x}^k;\dot{y})\right), \\ \tilde{x}^{k+1} & = \tilde{x}^k - \tau_k^{-1}(\hat{x}^k - \bar{x}^{k+1}), \\ \tau_{k+1} \in (0,1) & \text{ is the unique positive root of } \tau^3/L_{b_{\mathcal{Y}}} + \tau^2 + \tau_k^2\tau - \tau_k^2 = 0, \\ \beta_{k+2} & = \frac{\beta_{k+1}}{1+L^{-1}_{b_{\mathcal{Y}}}\tau_{k+1}} \end{cases} \tag{ASGARD}$$

We prove an $\mathcal{O}(1/k)$ convergence rate on the objective residual $P(\bar{x}^k) - P^\star$ of (3.1) for the algorithm, which is the best known in the literature for the fully nonsmooth setting. For the constrained case (3.2), we also prove the convergence of the algorithm in terms of the primal objective residual and the feasibility violation, both achieve an $\mathcal{O}(1/k)$ convergence rate, and are independent of the prox-diameters unlike existing smoothing techniques [BT12, Nes05a, Nes05b]. As the optimality measure is different in each case, we state two distinct theorems.

Theorem 9. *Suppose that $f = 0$ and $h = \delta_{\{c\}}$. Let $\beta_1 > 0$ and $b_{\mathcal{Y}}$ be chosen such that $L_{b_{\mathcal{Y}}} = 1$. Let $\{\bar{x}^k\}$ be the primal sequence generated by Algorithm ASGARD. Then the following bounds hold for* (3.2)*:*

$$\begin{cases} g(\bar{x}^k) - g^\star & \geq -\|y^\star\|_{\mathcal{Y}}\|A\bar{x}^k - c\|_{\mathcal{Y},*}, \\ g(\bar{x}^k) - g^\star & \leq \frac{1}{k}\frac{\bar{L}_A}{2\beta_1}\|\bar{x}^0 - x^\star\|^2_{\mathcal{X}} + \|y^\star\|_{\mathcal{Y}}\|A\bar{x}^k - c\|_{\mathcal{Y},*} + \frac{2\beta_1}{k+1}b_{\mathcal{Y}}(y^\star,\dot{y}), \\ \|A\bar{x}^k - c\|_{\mathcal{Y},*} & \leq \frac{\beta_1}{k+1}\Big[\|y^\star - \dot{y}\|_{\mathcal{Y}} + \big(\|y^\star - \dot{y}\|^2_{\mathcal{Y}} + \beta_1^{-2}\bar{L}_A\|\bar{x}^0 - x^\star\|^2_{\mathcal{X}}\big)^{1/2}\Big]. \end{cases} \tag{3.16}$$

Clearly, the choice of β_1 in Theorem 9 trades off between $\|\bar{x}^0 - x^\star\|_{\mathcal{X}}^2$ and $\|y^\star - \dot{y}\|_{\mathcal{Y}}^2$ on the primal objective residual $g(\bar{x}^k) - g^\star$ and on the feasibility gap $\|A\bar{x}^k - c\|_{\mathcal{Y},*}$.

Theorem 10. *Suppose that $f = 0$ and h is Lipschitz continuous, so that $D_{\mathcal{Y}} = \sup_{y \in \operatorname{dom} g^*} b_{\mathcal{Y}}(y, \dot{y}) < +\infty$. Let $\beta_1 > 0$ and $b_{\mathcal{Y}}$ be chosen such that $L_{b_{\mathcal{Y}}} = 1$. Let $\{\bar{x}^k\}$ be the primal sequence generated by Algorithm ASGARD. Then, the primal objective residual of* (3.1) *satisfies*

$$P(\bar{x}^k) - P(x^\star) \leq \frac{\bar{L}_A}{2\beta_1 k}\|\bar{x}^0 - x^\star\|_{\mathcal{X}}^2 + \frac{2\beta_1}{k+1}D_{\mathcal{Y}}, \quad \textit{for all } k \geq 1. \tag{3.17}$$

3.3.3 Extensions

The flexibility of the framework allowed us to develop several variants of ASGARD.

Dual algorithm We developed in [TDFC18] a dual version of ASGARD, called ADSGARD, that updates a sequence of primal-dual vectors $(\bar{x}^k, \bar{y}^k)$. Its main feature is that, although it is a dual algorithm, it has guarantees on the smoothed duality gap, from which we can derive a convergence rate in the primal space.

Moreover, by considering the non-strongly convex smoothing $b_{\mathcal{X}}(x, \dot{x}) = \frac{1}{2}\|Ax - A\dot{x}\|^2$, we could define ASALGARD, a method similar to the Augmented Lagrangian method but with improved worst case guarantee of the order $O(1/k^2)$.

When the objective function g is μ-strongly convex, we showed that it is not necessary to smooth its Fenchel conjugate g^*. Taking $\gamma_k = 0$ in the algorithm yields a faster algorithm with rate $O(1/k^2)$.

Heuristic restart Similar to other accelerated gradient algorithms in [GB14, OC12, SBC14, FQ16], restarting ASGARD and ADSGARD may lead to a better performance in practice.

If we consider ASGARD, then, when a restart takes place, we perform the following steps:

$$\left\{\begin{array}{lcl} \tilde{x}^{k+1} & \leftarrow & \bar{x}^{k+1}, \\ \dot{y} & \leftarrow & y^*_{\beta_{k+1}}(A\bar{x}^{k+1}; \dot{y}), \\ \beta_{k+1} & \leftarrow & \beta_1, \\ \tau_{k+1} & \leftarrow & 1. \end{array}\right. \tag{3.18}$$

Extension to Problem 3.1 In [VNFC17], we showed that ASGARD can deal with differentiable functions through their gradient. The step-sizes need to be adapted accordingly. We also proposed a novel line search that adapts the step-sizes and the smoothing parameters together. Thus, if the function h happened to be smooth near the optimum, the algorithm will automatically switch to accelerated gradient with $O(1/k^2)$ rate.

Going from a purely proximal to a forward-backward type method was a requirement in order to design a coordinate descent version of ASGARD that would take profit of longer step-sizes.

Conditional gradient with linear constraints In [YFLC18], we showed how to use the smoothing framework with a conditional gradient method. We obtain an algorithm that, using only gradients and linear minimization oracles, can solve complex problems, like for instance semi-definite programs, and has a convergence guarantee of order $O(1/\sqrt{k})$. This is the first method with such a rate under the setup of conditional gradient with linear constraints and the method has very good practical performance.

3.4 A primal-dual coordinate descent method based on smoothing

The paper [ADFC17] develops a random coordinate descent method to solve the composite problem (3.1):

$$F^\star = \min_{x \in \mathbb{R}^p}\{F(x) = f(x) + g(x) + h(Ax)\},$$

where $f : \mathbb{R}^p \to \mathbb{R}$, $g : \mathbb{R}^p \to \mathbb{R} \cup \{+\infty\}$, and $h : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ are proper, closed and convex functions, $A \in \mathbb{R}^{n\times p}$ is a given matrix. As explained before in this thesis, the optimization template (3.1) covers many important applications including support vector machines, sparse model selection, logistic regression, etc. It is also convenient to formulate generic constrained convex problems by choosing an appropriate h.

Before this work, there was no coordinate descent method for the general three-composite form (3.1) within our structure assumptions studied here that had rigorous convergence speed guarantees. In particular, we had not derived the convergence rate of Algorithm 6 yet. For such a theoretical development, coordinate descent algorithms require specific assumptions on the convex optimization problems [Nes12a, FR15, NC13]. As a result, to rigorously handle the three-composite case, we assume that (i) f is smooth, (ii) g is non-smooth but decomposable (each component has an "efficiently computable" proximal operator), and (iii) h is non-smooth.

We generalize [FR15, QR16] to the three composite case (3.1). For this purpose, we combine several classical and contemporary ideas: We exploit the smoothing technique in [Nes05b], the efficient implementation technique in [Nes12a, FR15], the homotopy strategy in [TDC15], and the nonuniform coordinate selection rule in [QR16] in our algorithm, achieving the best known complexity estimate.

Surprisingly, the combination of these ideas are achieved in a very natural and elementary primal-dual gap-based framework. However, the extension is indeed not trivial since it requires to deal with a composition of a non-smooth function h and a linear operator A.

We propose a new smooth primal-dual randomized coordinate descent method (Algorithm 8) for solving (3.1) where f is smooth, g is nonsmooth, separable and has a block-wise proximal operator, and h is a general nonsmooth function. Under such a structure, we show that our algorithm achieves the best known $\mathcal{O}(n/k)$ convergence rate, where k is the iteration count.

We instantiate our algorithm to solve special cases of (3.1) including the case $g = 0$ and constrained problems. We analyze the convergence rate guarantee of these variants individually and discuss the choice of nonuniform distribution to achieve the best convergence rate. Exploiting the strategy in [FR15], we show that our algorithm can be implemented in parallel by breaking up the full vector updates. We also provide a restart strategy to enhance practical performance.

Algorithm 8 (SMooth, Accelerate, Randomize The Coordinate Descent (SMART-CD))

Require: Choose $\beta_1 > 0$ and $\alpha \in [0, 1]$ as two input parameters.

Set $B_i^0 := \hat{L}_i + \frac{\|A_i\|^2}{\beta_1}$ for $i \in [n]$. Compute $S_\alpha := \sum_{i=1}^n (B_i^0)^\alpha$ and $q_i := \frac{(B_i^0)^\alpha}{S_\alpha}$ for all $i \in [n]$.

Set $\tau_0 := \min\{q_i \ : \ 1 \le i \le n\} \in (0, 1]$ for $i \in [n]$.

for $k \leftarrow 0, 1, \cdots, k_{\max}$ **do**

  Update $\hat{x}^k := (1 - \tau_k)\bar{x}^k + \tau_k \tilde{x}^k$ and compute $\hat{u}^k := A\hat{x}^k$.

  Compute the dual step $y^*_{\beta_{k+1}}(\hat{u}^k) := \operatorname{prox} \beta_{k+1}^{-1} h^* \dot{y} - \beta_{k+1}^{-1}\hat{u}^k$.

  Select a block coordinate $i_k \in [n]$ according to the probability distribution q.

  Set $\tilde{x}^{k+1} := \tilde{x}^k$, and compute the primal i_k-block coordinate:

$$\tilde{x}_{i_k}^{k+1} := \operatorname*{argmin}_{x_{i_k} \in \mathbb{R}^{p_{i_k}}} \Big\{ \langle \nabla_{i_k} f(\hat{x}^k) + A_{i_k}^T y^*_{\beta_{k+1}}(\hat{u}^k), x_{i_k} - \hat{x}_{i_k} \rangle_{(i_k)} + g_{i_k}(x_{i_k}) + \frac{\tau_k B_{i_k}^k}{2\tau_0} \|x_{i_k} - \tilde{x}_{i_k}^k\|_{(i_k)}^2 \Big\}.$$

  Update $\bar{x}^{k+1} := \hat{x}^k + \frac{\tau_k}{\tau_0}(\tilde{x}^{k+1} - \tilde{x}^k)$.

  Compute $\tau_{k+1} \in (0, 1)$ as the unique positive root of $\tau^3 + \tau^2 + \tau_k^2 \tau - \tau_k^2 = 0$.

  Update $\beta_{k+2} := \frac{\beta_k}{1+\tau_{k+1}}$ and $B_i^{k+1} := \hat{L}_i + \frac{\|A_i\|^2}{\beta_{k+2}}$ for $i \in [n]$.

end for

As the proof is using Lemma 3, our convergence result is split into the two important cases of Lipschitz h and linear equality constraint.

Theorem 11. *Let $x^\star$ be an optimal point of (3.1) where h is D_{h^*}-Lipschitz continuous and let $\beta_1 > 0$ be given. Let $\tau_0 := \min\{q_i \ : \ i \in [n]\} \in (0, 1]$ and $\beta_0 := (1 + \tau_0)\beta_1$ be given parameters. For all $k \ge 1$, the sequence $\{\bar{x}^k\}$ generated by Algorithm 8 satisfies:*

$$\mathbb{E}[F(\bar{x}^k) - F(x^\star)] \le \frac{C^\star}{\tau_0(k-1)+1} + \frac{\beta_1(1+\tau_0)D_{h^*}^2}{2(\tau_0 k + 1)}, \tag{3.19}$$

where $C^\star := (1-\tau_0)(F_{\beta_0}(x^0) - F(x^\star)) + \sum_{i=1}^n \frac{\tau_0 B_i^0}{q_i}\|x_i^\star - \tilde{x}_i^0\|_{(i)}^2$.

Theorem 12. *Let* $\{\bar{x}^k\}$ *be the sequence generated by Algorithm 8 for solving* (3.1) *where* $h = \delta_{\{b\}}$. *Then, we have the following estimate:*

$$\begin{cases} \mathbb{E}[F(\bar{x}^k) - F(x^\star)] & \leq \frac{(1-\tau_0)C^*}{\tau_0(k-1)+1} + \frac{\beta_1\|y^\star - \dot{y}\|^2}{2(\tau_0(k-1)+1)} + \|y^*\|\mathbb{E}[\|A\bar{x}^k - b\|], \\ \mathbb{E}\|A\bar{x}^k - b\| & \leq \frac{\beta_1}{\tau_0(k-1)+1}\left[\|y^* - \dot{y}\| + \left(\|y^* - \dot{y}\|^2 + 4\beta_1^{-1}C^*\right)^{1/2}\right], \end{cases} \tag{3.20}$$

where $C^* := (1-\tau_0)(F_{\beta_0}(x^0) - F(x^\star)) + \sum_{i=1}^n \frac{\tau_0 B_i^0}{q_i}\|x_i^\star - \tilde{x}_i^0\|_i^2$. *We note that the following lower bound always holds* $-\|y^\star\|_*\mathbb{E}[\|A\bar{x}^k - b\|] \leq \mathbb{E}[F(\bar{x}^k) - F(x^\star)]$.

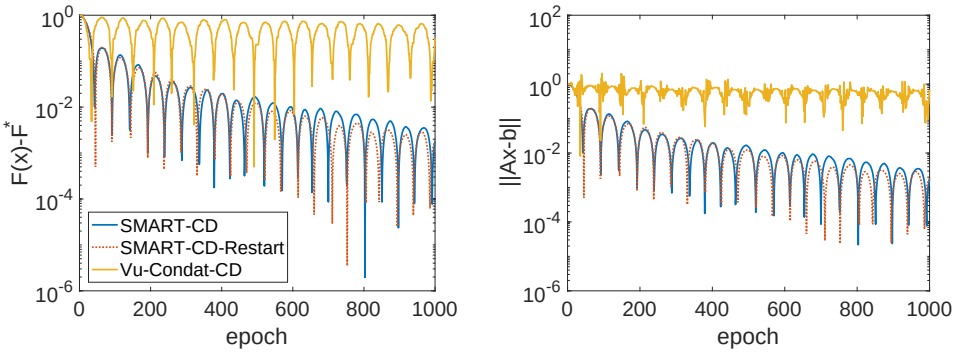

Figure 3.4: The convergence behavior of SMART-CD (Algorithm 8) and Vu-Condat-CD (Algorithm 6) for a degenerate linear program with repeated constraints. We observe that degeneracy of the problem prevents Vu-Condat-CD from making any progress towards the solution (we only proved $O(1/\sqrt{k})$ speed of convergence for Algorithm 6), while SMART-CD preserves $O(1/k)$ rate as predicted by theory.

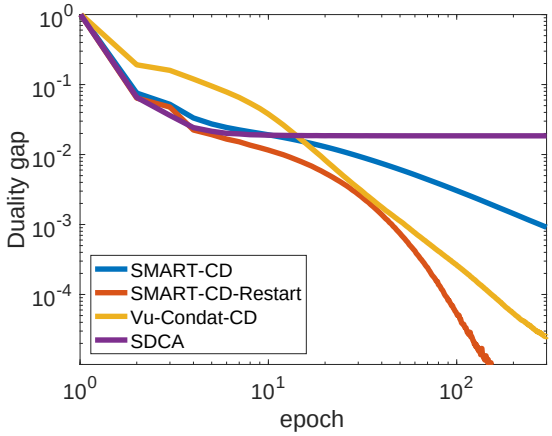

Figure 3.5: Comparison of coordinate descent methods for dual SVM with bias on `rcv1` dataset: 47,236 examples, 20,242 features – $\min_x \left\{\frac{1}{2\lambda}\|AD(b)x\|^2 - \sum_{i=1}^n x_i \;\text{ s.t. } 0 \leq x_i \leq C_i,\; i = 1, \cdots, n,\; b^\top x = 0\right\}$

Chapter 4

Applications to statistics

4.1 Gap Safe screening rules for sparsity enforcing penalties

4.1.1 Introduction

The computational burden of solving high dimensional regularized regression problem has led to a vast literature on improving algorithmic solvers in the last two decades. With the increasing popularity of ℓ_1-type regularization ranging from the Lasso [Tib96] or group-Lasso [YL06] to regularized logistic regression and multi-task learning, many algorithmic methods have emerged to solve the associated optimization problems [KKB07, BJM+12]. Although for the simple ℓ_1 regularized least square a specific algorithm (e.g. the LARS [EHJT04]) can be considered, for more general formulations, penalties, and possibly larger dimensions, (block) coordinate descent has proved to be an efficient strategy [FHT10].

Our main objective in this work is to propose a technique that can speed-up any iterative solver for such learning problems, and that is particularly well suited for (block) coordinate descent method as this type of method can easily ignore useless coordinates.

The *safe rules* introduced by [EVR12] for generalized ℓ_1 regularized problems, is a set of rules allowing to eliminate features whose associated coefficients are guaranteed to be zero at the optimum, even before starting any algorithm. Relaxing the safe rule, one can obtain some additional speed-up at the price of possible mistakes. Such heuristic strategies, called *strong rules* by [TBF+12] reduce the computational cost using an active set strategy, but require difficult post-processing to check for features possibly wrongly discarded.

Another road to speed-up screening method has been pursued following the introduction of *sequential safe rules* [EVR12, WWY12, XWR14, WZL+14]. The idea is to improve the screening thanks to the computation done for a previous regularization parameter as in homotopy/continuation methods. This scenario is particularly relevant in machine learning, where one computes solutions over a grid of regularization parameters, so as to select the best one, e.g. by cross-validation. Nevertheless, the aforementioned methods suffer from the same problem as strong rules, since relevant features can be wrongly disregarded. Indeed, sequential rules usually rely on the exact knowledge of certain theoretical quantities that are only known approximately. Especially, for such rules to work one needs the exact dual optimal solution from the previous regularization parameter, a quantity (almost) never available to the practitioner.

The introduction of *dynamic safe rules* by [BERG15, BERG14] has opened a promising venue by performing variable screening, not only before the algorithm starts, but also along the iterations. This screening strategy can be applied for any standard optimization algorithm such as FISTA [BT09], primal-dual [CP11], augmented Lagrangian [BPC+11a]. Yet, it is particularly relevant for strategies that can benefit from support reduction or active sets[KWGA11, JG15], such as coordinate-descent [Fu98, FHHT07, FHT10].

We shall present the methods introduced first for the Lasso in [FGS15] and then for ℓ_1/ℓ_2 norms in [NFGS15] as well as for Sparse Group Lasso in [NFGS16], and summarized in [NFGS17]. Our so-called Gap Safe rules (because the screening rules rely on duality gap computations), improved on dynamic safe rules for a broad class of learning problems with the following benefits:

- Gap Safe rules are easy to insert in existing solvers,

- they are proved to be safe and unify sequential and dynamic rules,
- they are more efficient in practice than previously known safe rules,
- they achieve fast variable identifications.

Furthermore, it is worth noting that strategies also leveraging dual gap computations have recently been considered to safely discard irrelevant coordinates: [SKHT16] have considered screening rules for learning tasks with both feature sparsity and sample sparsity, such as for ℓ_1-regularized SVM. In this case, some interesting developments have been proposed, namely *safe keeping* strategies, which allow to identify features and samples that are guaranteed to be active. Constrained convex problems such as minimum enclosing ball can also be included as shown in [ROG+16]. The Blitz algorithm by [JG15] aims to speed up working set methods using duality gaps computations; significant gains were also obtained in limited-memory and distributed settings.

4.1.2 Safe Screening rules

We propose to estimate the vector of parameters β by solving

$$\hat{\beta}^{(\lambda)} \in \arg\min_{\beta\in\mathbb{R}^p} P_\lambda(\beta), \text{ for } P_\lambda(\beta) := F(\beta) + \lambda\Omega(\beta) := \sum_{i=1}^{n} f_i(x_i^\top\beta) + \lambda\Omega(\beta) \ , \tag{4.1}$$

where all $f_i : \mathbb{R} \mapsto \mathbb{R}$ are convex and differentiable functions with $1/\gamma$-Lipschitz gradient and $\Omega : \mathbb{R}^p \mapsto \mathbb{R}_+$ is a norm that is group-decomposable, i.e. $\Omega(\beta) = \sum_{g\in\mathcal{G}} \Omega_g(\beta_g)$ where each Ω_g is a norm on $\mathbb{R}^{n_g}$. The λ parameter is a non-negative constant controlling the trade-off between the data fitting term and the regularization term. We shall denote the dual norm of Ω by $\Omega^D(z) = \max_{g\in\mathcal{G}} \Omega_g^D(z_g)$.

A dual formulation of 4.1 is given by

$$\hat{\theta}^{(\lambda)} = \arg\max_{\theta\in\Delta_X} -\sum_{i=1}^{n} f_i^*(-\lambda\theta_i) =: D_\lambda(\theta), \tag{4.2}$$

where $\Delta_X = \{\theta \in \mathbb{R}^n : \forall g \in \mathcal{G}, \Omega_g^D(X_g^\top\theta) \leq 1\}$. Moreover, the Fermat's rule reads:

$$\forall i \in [n],\ \hat{\theta}_i^{(\lambda)} = -\nabla f_i(x_i^\top\hat{\beta}^{(\lambda)})/\lambda \quad \text{(link equation)}, \tag{4.3}$$

$$\forall g \in \mathcal{G},\ X_g^\top\hat{\theta}^{(\lambda)} \in \partial\Omega_g(\hat{\beta}_g^{(\lambda)}) \quad \text{(sub-differential inclusion)}. \tag{4.4}$$

For any $\theta \in \mathbb{R}^n$ let us introduce $G(\theta) := [\nabla f_1(\theta_1), \ldots, \nabla f_n(\theta_n)]^\top \in \mathbb{R}^n$. Then the primal/dual link equation can be written $\hat{\theta}^{(\lambda)} = -G(X\hat{\beta}^{(\lambda)})/\lambda$.

Contrarily to the primal, the dual problem has a unique solution under our assumption on the f_i's. Indeed, the dual function is strongly concave.

Screening rules rely on a direct consequence of Fermat's rule. If $\hat{\beta}(\lambda)_g \neq 0$, then $\Omega_g^D(X_g^\top\hat{\theta}^{(\lambda)}) = 1$ thanks to the formula for the subdifferential of a norm

$$\partial\Omega(x) = \begin{cases} \{z \in \mathbb{R}^d : \Omega^D(z) \leq 1\} = \mathcal{B}_{\Omega^D}, & \text{if } x = 0, \\ \{z \in \mathbb{R}^d : \Omega^D(z) = 1 \text{ and } z^\top x = \Omega(x)\}, & \text{otherwise.} \end{cases} \tag{4.5}$$

Since $\hat{\theta}^{(\lambda)} \in \Delta_X$, it implies, by contrapositive, that if $\Omega_g^D(X_g^\top\hat{\theta}^{(\lambda)}) < 1$ then $\hat{\beta}(\lambda)_g = 0$. This relation means that the g-th group can be discarded whenever $\Omega_g^D(X_g^\top\hat{\theta}^{(\lambda)}) < 1$. However, since $\hat{\theta}^{(\lambda)}$ is unknown, this rule is of limited use. Fortunately, it is often possible to construct a set $\mathcal{R} \subset \mathbb{R}^n$, called a *safe region*, that contains $\hat{\theta}^{(\lambda)}$. This observation leads to the following result.

Proposition 4 (Safe screening rule [EVR12])**.** *If $\hat{\theta}^{(\lambda)} \in \mathcal{R}$, and $g \in \mathcal{G}$:*

$$\max_{\theta\in\mathcal{R}} \Omega_g^D(X_g^\top\theta) < 1 \Longrightarrow \Omega_g^D(X_g^\top\hat{\theta}^{(\lambda)}) < 1 \Longrightarrow \hat{\beta}(\lambda)_g = 0 \ . \tag{4.6}$$

The so-called *safe screening* rule consists in removing the g-th group from the problem whenever the previous test is satisfied, since then $\hat{\beta}(\lambda)_g$ is guaranteed to be zero. Should $\mathcal{R}$ be small enough to screen many groups, one can observe considerable speed-ups in practice as long as the testing can be performed efficiently. A natural goal is to find safe regions as narrow as possible: smaller safe regions can only increase the number of screened out variables. To have useful screening procedures one needs:

- the safe region $\mathcal{R}$ to be as small as possible (and to contain $\hat{\theta}^{(\lambda)}$),
- the computation of the quantity $\max_{\theta \in \mathcal{R}} \Omega_g^D(X_g^\top \theta)$ to be cheap.

Various shapes have been considered in practice for the safe region $\mathcal{R}$ such as balls [EVR12], domes [FGS15] or more refined sets (see [XWR14] for a survey). Here we consider for simplicity the so-called "sphere regions" (following the terminology introduced by [EVR12]) choosing a ball $\mathcal{R} = B(\theta_c, r)$ as a safe region. Thanks to the triangle inequality, we have:

$$\max_{\theta \in B(\theta_c, r)} \Omega_g^D(X_g^\top \theta) \leq \Omega_g^D(X_g^\top \theta_c) + \max_{\theta \in B(\theta_c, r)} \Omega_g^D(X_g^\top(\theta - \theta_c)),$$

and denoting $\Omega_g^D(X_g) := \sup_{u \neq 0} \frac{\Omega_g^D(X_g^\top u)}{\|u\|_2}$ the operator norm of X_g associated to $\Omega_g^D(\cdot)$, we deduce from Proposition 4.6 the screening rule for the g-th group:

$$\text{Safe sphere test:} \qquad \text{If} \quad \Omega_g^D(X_g^\top \theta_c) + r\Omega_g^D(X_g) < 1, \quad \text{then} \quad \hat{\beta}(\lambda)_g = 0 \ . \tag{4.7}$$

Finding a center To create a useful center for a safe sphere, one needs to be able to create dual feasible points, i.e. points in the dual feasible set Δ_X. One such point is $\theta_{\max} := -G(0)/\lambda_{\max}$ which leads to the original static safe rules proposed by [EVR12]. Yet, it has a limited interest, being helpful only for a range of small regularization parameters λ. A more generic way of creating a dual point that will be key for creating our safe rules is to rescale any point $z \in \mathbb{R}^n$ such that it is in the dual set Δ_X. The rescaled point is denoted by $\Theta(z)$ and is defined by

$$\Theta(z) := \begin{cases} z, & \text{if } \Omega^D(X^\top z) \leq 1, \\ \frac{z}{\Omega^D(X^\top z)}, & \text{otherwise.} \end{cases} \tag{4.8}$$

This choice guarantees that $\forall z \in \mathbb{R}^n, \Theta(z) \in \Delta_X$. A candidate often considered for computing a dual point consists in starting by the (generalized) residual term $z = -G(X\beta)/\lambda$. This choice is motivated by the primal-dual link equation (4.3) i.e. $\hat{\theta}^{(\lambda)} = -G(X\hat{\beta}(\lambda))/\lambda$. We also have the following theoretical guarantee.

Proposition 5 (Convergence of the dual points)**.** *Let β_k be the current estimate of $\hat{\beta}^{(\lambda)}$ and $\theta_k = \Theta(-G(X\beta_k)/\lambda)$ be the current estimate of $\hat{\theta}^{(\lambda)}$. Then $\lim_{k \to +\infty} \beta_k = \hat{\beta}^{(\lambda)}$ implies $\lim_{k \to +\infty} \theta_k = \hat{\theta}^{(\lambda)}$.*

Finding a radius Now that we have seen how to create a center candidate for the sphere, we need to find a proper radius, that would allow the associated sphere to be safe. The following theorem proposes a way to obtain a radius using the duality gap:

Theorem 13 (Gap Safe sphere)**.** *Assuming that F has $1/\gamma$-Lipschitz gradient, we have*

$$\forall \beta \in \mathbb{R}^p, \forall \theta \in \Delta_X, \quad \|\hat{\theta}(\lambda) - \theta\|_2 \leq \sqrt{\frac{2(P_\lambda(\beta) - D_\lambda(\theta))}{\gamma \lambda^2}} =: r_\lambda(\beta, \theta) \ . \tag{4.9}$$

Hence $\mathcal{R} = B(\theta, r_\lambda(\beta, \theta))$ is a safe region for any $\beta \in \mathbb{R}^n$ and $\theta \in \Delta_X$.

Safe Active Set Note that any time a safe rule is performed thanks to a safe region $\mathcal{R} = B(\theta, r)$, one can associate a *safe active set* $\mathcal{A}_{\theta,r}$, consisting of the features that cannot be removed yet by the test in Equation 4.7. Hence, the safe active set contains the true support of $\hat{\beta}^{(\lambda)}$.

When choosing $z = -G(X\beta)/\lambda$ as proposed in Section 4.1.2 as the current residual, the computation of $\theta = \Theta(z)$ in Equation 4.8 involves the computation of $\Omega^D(X^\top z)$. A straightforward implementation would cost $\mathcal{O}(np)$ operations. This can be avoided: when using a safe rule one knows that the index achieving the maximum for this norm is in $\mathcal{A}(\theta, r)$. Indeed, by construction of the safe active set, it is easy to see that $\Omega^D(X^\top z) = \max_{g \in \mathcal{A}(\theta,r)} \Omega_g^D(X_g^\top z)$. In practice the evaluation of the dual gap is therefore $\mathcal{O}(nq)$ where q is the size of $\mathcal{A}(\theta, r)$. In other words, using a safe screening rule also speeds up the evaluation of the stopping criterion.

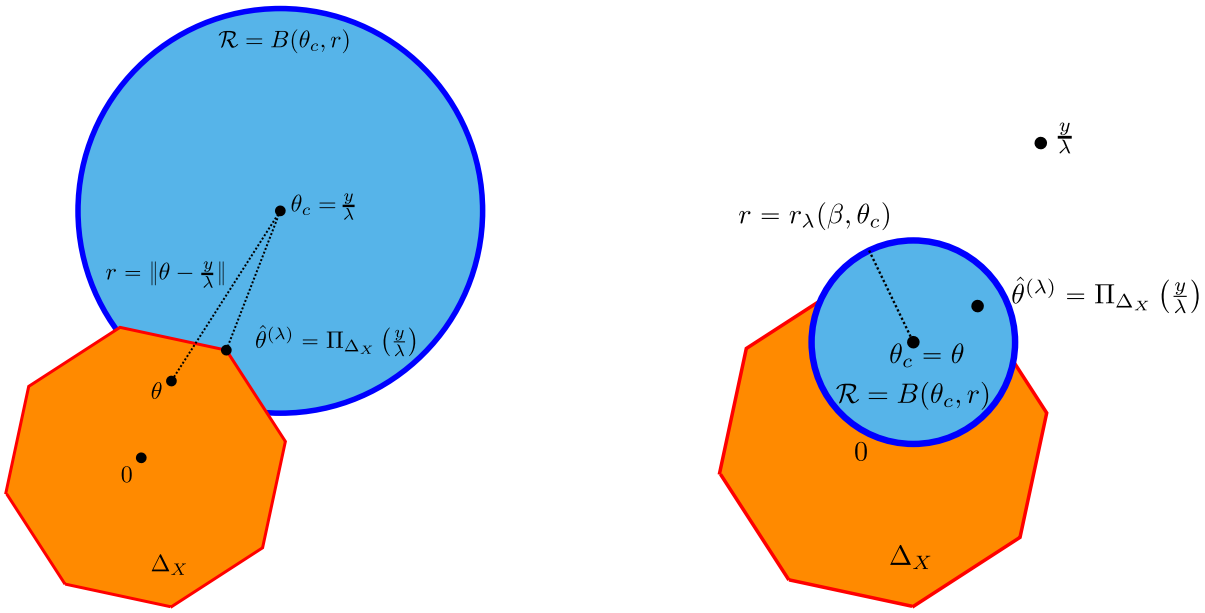

Figure 4.1: Illustration of safe region differences between Bonnefoy et al [BERG15] (left) and Gap Safe (right) strategies for the Lasso case. Here β is a primal point, θ is a dual feasible point (the feasible region Δ_X is in orange, while the respective safe balls $\mathcal{R}$ are in blue), and $r_\lambda(\beta, \theta)$ is defined by Equation 4.9.

4.1.3 Experiments

The algorithm is given in Algorithm 9. It improves an iterative solver with updates SolverUpdate by sequentially improving the safe active set. For our experiments, we chose a frequency such that $f^{\text{ce}} = 10$. The rationale for this frequency is that computing the dual variable amounts to one matrix-vector multiply by X, which is comparable to one iteration of gradient descent or n iterations of coordinate descent.

Algorithm 9 Iterative solver with GAP safe rules: Solver $(X,\ y,\ \beta,\ \epsilon,\ K,\ f^{\text{ce}},\ \lambda)$

Require: $X,\ y,\ \beta,\ \epsilon,\ K,\ f^{\text{ce}},\ \lambda$
  for $k \in [K]$ **do**
    if $k \mod f^{\text{ce}} = 1$ **then**
      Compute a dual variable $\theta = -G(X\beta)/\max(\lambda, \Omega^D(X^\top G(X\beta)))$
      Stop if $\text{Gap}_\lambda(\beta, \theta) \leq \epsilon$
      $r = \sqrt{\frac{2\text{Gap}_\lambda(\beta,\theta)}{\gamma\lambda^2}}$ Get Gap Safe radius as in Equation 4.9
      $\mathcal{A} = \left\{g \in \mathcal{G} : \Omega_g^D(X_g^\top \theta) + r\Omega_g^D(X_g) \geq 1\right\}$ Get Safe active set
    end if
    $\beta_\mathcal{A} = \text{SolverUpdate}\,(X_\mathcal{A},\ y,\ \beta_\mathcal{A},\ \lambda)$ Solve on current Safe active set
  end for

4.1.4 Alternative Strategies: a Brief Survey

The Seminal Safe Regions The first Safe Screening rules introduced by [EVR12] can be generalized to Problem 4.1 as follows. Take $\hat{\theta}^{(\lambda_0)}$ the optimal solution of the dual problem 4.2 with a regularization parameter λ_0. Since $\hat{\theta}^{(\lambda)}$ is optimal for problem 4.2 one obtains $\hat{\theta}^{(\lambda)} \in \{\theta : D_\lambda(\theta) \geq D_\lambda(\hat{\theta}^{(\lambda_0)})\}$. This set was proposed as a safe region by [EVR12]. In the regression case (where $f_i(z) = (y_i - z)^2/2$), it is straightforward to see that it corresponds to the safe sphere $B(y/\lambda, \|y/\lambda - \hat{\theta}^{(\lambda_0)}\|_2)$.

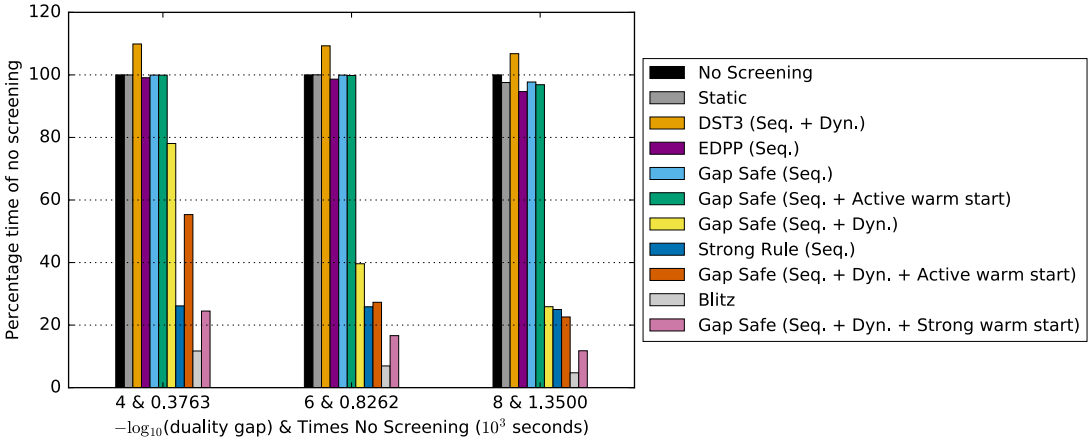

Figure 4.2: Lasso on financial data E2006-log1p (sparse data with $n = 16087$ observations and $p = 1668737$ features). Computation times needed to solve the Lasso regression path to desired accuracy for a grid of 10 values of λ from $\lambda_{\max}$ to $\lambda_{\max}/20$. We compare several refinements of our Gap Safe technique with the alternative strategies described in the literature. Seq (sequential) means that we only compute the active set at the first iteration using the last point returned for the previous λ. Dyn (dynamic) means that we update the active set every f^{ce} iterations. We can see that dynamic updates yield much better performance (up to a 10 times speedup).

ST3 and Dynamic ST3 A refined sphere rule can be obtained in the regression case by exploiting geometric informations in the dual space. This method was originally proposed in [XXR11] and extended in [BERG14] with a dynamic refinement of the safe region.

Let $g_\star \in \arg\max_{g\in\mathcal{G}} \Omega_g^D(X^\top y)$ (note that $\Omega_{g_\star}^D(X^\top y) = \lambda_{\max}$), and let us define

$$\mathcal{V}_\star := \{\theta \in \mathbb{R}^n : \Omega_{g_\star}^D(X_{g_\star}^\top \theta) \leq 1\} \text{ and } \mathcal{H}_\star := \{\theta \in \mathbb{R}^n : \Omega_{g_\star}^D(X_{g_\star}^\top \theta) = 1\}.$$

Let $\eta := X_{g_\star} \nabla\Omega_{g_\star}^D(X_{g_\star}^\top y/\lambda_{\max})$ be the vector normal to $\mathcal{V}_\star$ at $y/\lambda_{\max}$ and define

$$\theta_c := \Pi_{\mathcal{H}_\star}\left(\frac{y}{\lambda}\right) = \frac{y}{\lambda} - \frac{\langle \frac{y}{\lambda}, \eta\rangle - 1}{\|\eta\|_2^2}\eta \text{ and } r_\theta := \sqrt{\|\frac{y}{\lambda} - \theta\|_2^2 - \|\frac{y}{\lambda} - \theta_c\|_2^2}\ ,$$

where $\theta \in \Delta_X$ is any dual feasible vector. One can show that $\hat{\theta}^{(\lambda)} \in B(\theta_c, r_\theta)$. The special case where $\theta = y/\lambda_{\max}$ corresponds to the original ST3 introduced in [XXR11] for the Lasso.

Dual Polytope Projection In the regression case, [WWY12] explore other geometric properties of the dual solution. Their method is based on the non-expansiveness of projection operators[1]. Indeed, for $\hat{\theta}^{(\lambda)}$ (resp. $\hat{\theta}^{(\lambda_0)}$)) being optimal dual solution of 4.2 with parameter λ (resp. λ_0), one has: $\|\hat{\theta}^{(\lambda)} - \hat{\theta}^{(\lambda_0)}\|_2 = \|\Pi_{\Delta_X}(y/\lambda) - \Pi_{\Delta_X}(y/\lambda_0)\|_2 \leq \|y/\lambda - y/\lambda_0\|_2$ and hence $\hat{\theta}^{(\lambda)} \in \mathcal{B}(\hat{\theta}^{(\lambda_0)}, \|y/\lambda - y/\lambda_0\|_2)$. Unfortunately, those regions are intractable since they required the *exact* knowledge of the optimal solution $\hat{\theta}^{(\lambda_0)}$ which is not available in practice (except for $\lambda_0 = \lambda_{\max}$).

Strong rules The Strong rules were introduced in [TBF+12] as a *heuristic* extension of the safe rules. It consists in relaxing the *safe* properties to discard features more aggressively, and can be formalized as follows. Assume that the gradient of the data fitting term ∇F is group-wise non-expansive w.r.t. the dual norm $\Omega_g^D(\cdot)$ along the regularization path i.e. that for any $g \in \mathcal{G}$, any $\lambda > 0, \lambda' > 0$, $\Omega_g^D\big(\nabla_g F(\hat{\beta}^{(\lambda)}) - \nabla_g F(\hat{\beta}^{(\lambda')})\big) \leq |\lambda - \lambda'|$. When choosing two regularization parameters such that $\lambda < \lambda'$ this assumption leads to: $\lambda\Omega_g^D(X_g^\top\hat{\theta}^{(\lambda)}) \leq \lambda'\Omega_g^D(X_g^\top\hat{\theta}^{(\lambda')}) + \lambda' - \lambda$. Combining this with the screening rule (4.6), one obtains: $\Omega_g^D(X_g^\top\hat{\theta}^{(\lambda')}) < \frac{2\lambda - \lambda'}{\lambda'} \Longrightarrow \hat{\beta}_g^{(\lambda)} = 0$.

[1] The authors also proved an enhanced version of this safe region by using the firm non-expansiveness of the projection operator.

The set of variables not eliminated is called the *strong active set*. Note that Strong rules are un-safe because the non-expansiveness condition on the (gradient of the) data fitting term is usually not satisfied without stronger assumptions on the design matrix X; see discussion in [TBF+12]. It requires the exact knowledge of $\hat{\theta}^{(\lambda')}$ which is not available in practice. Using such rules, the authors advised to check the KKT condition a posteriori, to avoid removing wrongly some features.

Correlation Based Rule Previous works in statistics have proposed various model-based screening methods to select important variables. Those methods discard variables with small correlation between the features and response variables. For instance Sure Independence Screening (SIS) by [FL08] reads: for a chosen critical threshold γ remove the feature if the correlation with the observation is smaller than γ. It is a marginal oriented variable selection method and it is worth noting that SIS can be recast as a static sphere test in linear regression scenarios:

$$\text{If } \Omega_g^D(X_g^\top y) < \gamma = \lambda \left(1 - r\Omega_g^D(X_g)\right) \text{ then } \hat{\beta}_g^{(\lambda)} = 0 \,(\text{remove } X_g).$$

Other refinements can also be found in the literature such as iterative screening (ISIS) [FL08], that bears some similarities with dynamic sphere safe tests.

4.2 Efficient Smoothed Concomitant Lasso Estimation for High Dimensional Regression

In the context of high dimensional regression where the number of features is greater than the number of observations, standard least squares need some regularization to both avoid over-fitting and ease the interpretation of discriminant features. Among the least squares with sparsity inducing regularization, the Lasso [Tib96], using the ℓ_1 norm as a regularizer, is the most standard one. It hinges on a regularization parameter governing the trade-off between data fitting and sparsity of the estimator, and requires careful tuning. Though this estimator is well understood theoretically (we refer to [Bv11] for a review), the choice of the tuning parameter remains an open and critical question in practice as well as in theory. For the Lasso, statistical guarantees [BRT09] rely on choosing the tuning parameter proportional to the noise level, a quantity that is usually unknown to practitioners. We also want to mention that automatic tuning (e.g. by cross-validation) would be time consuming for applications like dictionary learning, where this parameter is often set once and for all [MBPS10]. Besides, the noise level is of practical interest since it is required in the computation of model selection criterions such as AIC, BIC, SURE or in the construction of confidence sets. As shown in [FGH12], this is a challenging task in high dimension because of the spurious correlation phenomenon.

A convenient way to estimate both the regression coefficient and the noise level is to perform a joint estimation, for instance by performing the penalized maximum likelihood of the joint distribution. Unfortunately, a direct approach leads to a non-convex formulation (though one can recover a jointly convex formulation through a change of variable [SBv10]).

Another road for this joint estimation was inspired by the robust theory developed by Huber [Hub81], particularly in the context of location-scale estimation. Indeed, Owen [Owe07] extended it to handle sparsity inducing penalty, leading to the jointly convex optimization formulation

$$(\hat{\beta}^{(\lambda)}, \hat{\sigma}^{(\lambda)}) \in \underset{\beta \in \mathbb{R}^p, \sigma > 0}{\arg\min} \frac{\|y - X\beta\|^2}{2n\sigma} + \frac{\sigma}{2} + \lambda \|\beta\|_1. \tag{4.10}$$

Since then, his estimator has appeared under various name, and we coined it the Concomitant Lasso. Indeed, as far as we know Owen was the first to propose such a formulation.

Later, the same formulation was mentioned in [Ant10], in a response to [SBv10], and was thoroughly analyzed in [SZ12], under the name Scaled-Lasso. Similar results were independently obtained in [BCW11] for the same estimator, though with a different formulation. While investigating pivotal quantities, Belloni et al. proposed to solve the following convex program: modify the standard Lasso by removing the square in the data fitting term. Thus, they termed their estimator the Square-root Lasso (see also [CD11]). A second approach leading to this very formulation, was proposed by [XCM10] to account for noise in the design matrix, in an adversarial scenario. Interestingly their robust construction led exactly to the Square-root Lasso formulation.

Under standard design assumption (see [BRT09]), it is proved that the Scaled/Square-root Lasso reaches optimal rates for sparse regression, with the additional benefit that the regularization parameter is independent of the noise level [BCW11, SZ12]. Moreover, a practical study [RTF13] has shown that the Concomitant Lasso estimator, or its debiased version (see for instance [BC13, Led13] for a discussion on least-squares refitting), is particularly well suited for estimating the noise level in high dimension.

Among the solutions to compute the Concomitant Lasso, two roads have been pursued so far. On the one hand, considering the Scaled-Lasso formulation, Sun and Zhang [SZ12] have proposed an iterative procedure that alternates Lasso steps and noise estimation steps, the later leading to rescaling the tuning parameter iteratively. On the other hand, considering the Square-root Lasso formulation, Belloni et al. [BCW11] have leaned on second order cone programming solvers, e.g. TFOCS [BCG11].

Despite the appealing properties listed above, among which the superiority of the theoretical results is the most striking, no consensus for an efficient solver has yet emerged for the Concomitant Lasso. Our contribution in [NFG+17] aims at providing a more numerically stable formulation, called the Smoothed Concomitant Lasso:

$$\underset{\beta\in\mathbb{R}^p,\sigma\in\mathbb{R}}{\arg\min}\ \frac{\|y-X\beta\|^2}{2n\sigma}+\frac{\sigma}{2}+\lambda\|\beta\|_1+\iota_{[\sigma_0,+\infty[}(\sigma).$$

This variant allows to obtain a fast solver: we first adapt a coordinate descent algorithm to the smooth version (in the sense given in [Nes05b]) of the original problem (4.10). Then, we apply safe rules strategies, like the ones in Section 4.1 to our estimator. Such rules allow to discard features whose coefficients are certified to be zero, either prior any computation or as the algorithm proceeds. Combined with a coordinate descent, this leads to important acceleration in practice, as illustrated for the Lasso case [FGS15]. We show similar accelerations for the Smoothed Concomitant Lasso, both on real and simulated data. Overall, our method presents the same computational cost as for the Lasso, but enjoys the nice features mentioned earlier in terms of statistical properties.

In [MFGS17], we studied an extension of the Smoothed Concomitant Lasso estimator to the case where we know that the observations can be partitioned into given groups with a different noise level. This situation occurs for instance when observations stem from different types of sensors like in M/EEG (magnetometers, gradiometers and electrodes). This leads to the following homoscedastic model

$$\underset{\substack{\mathrm{B}\in\mathbb{R}^{p\times q},\\ \sigma_1,\dots,\sigma_K\in\mathbb{R}_+\\ \sigma_k\geq\underline{\sigma}_k,\forall k\in[K]}}{\arg\min}\ \sum_{k=1}^{K}\left(\frac{\|Y^k-X^k\mathrm{B}\|^2}{2nq\sigma_k}+\frac{n_k\sigma_k}{2n}\right)+\lambda\|\mathrm{B}\|_{2,1}$$

that can also be solved using a block coordinate descent method acting on lines of B and $(\sigma_k)_{k\in[K]}$.

4.3 Safe Grid Search with Optimal Complexity

Various machine learning problems are formulated as a minimization of an empirical loss function f, regularized by a term Ω whose calibration and complexity is controlled by a non negative hyperparameter λ. The (optimal) choice of regularization parameter λ is crucial since it directly influences the generalization performance of the estimator, i.e. its score on unseen data set. One of the most popular method in such approaches is cross-validation (CV), see [AC10] for a detailed review. For simplicity, we investigate here the simplified holdout version that consists in splitting the data in two parts: on the first part (*training set*) the method is trained for a pre-defined collection of candidates $\Lambda_T := \{\lambda_0,\dots,\lambda_{T-1}\}$, and on the second part (*validation set*), the best parameter is selected. For a piecewise quadratic loss f and a piecewise linear regularization Ω, one can show [OPT00, RZ07] that the set of solutions follows a piecewise linear curve with respect to to the parameter λ. Hence there are several efficient algorithms that can generate the full path such as LARS for Lasso [EHJT04], for SVM [HRTZ04] and for the generalized linear models [PH07].

Unfortunately, these methods have a worst case complexity, i.e. the number of linear segment, that is exponential in the dimension of the problem [GJM12] leading to unpractical solutions. To overcome this issue, an approximation of the solution path up to accuracy ϵ was proposed and optimal complexity was proved to be $O(1/\epsilon)$ [GJL10] in a fairly general setting. A noticeable contribution was proposed by [MY12], who refined that bound to $O(1/\sqrt{\epsilon})$ in the Lasso case. The later result was then generalized by [GMLS12] with a lower and upper bound of order $O(1/\sqrt{\epsilon})$. A generic algorithm was also derived by

Algorithm 10 ϵ_{p}-Path on Training Set: `Training_path`

Input: $f, \Omega, \epsilon_{\text{p}}, [\lambda_{\min}, \lambda_{\max}]$
Initialize $t = 0$, $\lambda_0 = \lambda_{\max}$, $\Lambda = \{\lambda_{\max}\}$.
repeat
  Solve $\min_\beta f(X\beta) + \lambda_t \Omega(\beta)$ up to accuracy $\epsilon_{\mathbf{o}} < \epsilon_{\text{p}}$
  Compute $\rho_t^\ell = \max\{\rho \text{ s.t. } Q_{t,\mathcal{V}_{f^*}}(\rho) \leq \epsilon_{\text{p}}\}$
  Set $\lambda_{t+1} = \lambda_t \times (1 - \rho_t^\ell)$
  $\Lambda \leftarrow \Lambda \cup \{\lambda_{t+1}\}$ and $t \leftarrow t + 1$
until $\lambda_{t+1} \leq \lambda_{\min}$
Return: $\{\beta^{(\lambda_t)} \ : \ \lambda_t \in \Lambda\}$

assuming a quadratic lower bound on the objective function. Unfortunately, this assumption does not hold for a large class of problem, in particular for logistic regression.

Following such ideas, [SSKT15] have proposed, for classification problem, to approximate the regularization path for the hold-out cross-validation error. Indeed, the later is a more natural criterion to monitor when one aims at selecting a hyperparameter guaranteed to achieve the best validation error possible. The main idea is to construct an upper and lower bound on the validation error as simple functions of the regularization parameter. Hence by sequentially varying the parameters, one can estimate a range of parameter for which the validation error is smaller than an accuracy ϵ_v.

In [NFS$^+$18], we revisit the approximation and validation path results in a unified framework, under general regularity assumptions on the conjugate of the loss function f that are commonly satisfied in machine learning problems:

$$\mathcal{U}_{f^*,x}(z - x) \leq f^*(z) - f^*(x) - \langle \nabla f^*(x), z - x \rangle \ ,$$
$$\mathcal{V}_{f^*,x}(z - x) \geq f^*(z) - f^*(x) - \langle \nabla f^*(x), z - x \rangle \ .$$

These assumptions are general enough to encompass both classification (logistic loss) and regression (square loss) problems.

Suppose we have at our disposal a primal/dual pair of vector $(\beta^{(\lambda_t)}, \theta^{(\lambda_t)})$ computed as outputs of an optimization algorithm at regularization parameter λ_t, we denote

$$\text{Gap}_t := \text{Gap}_{\lambda_t}(\beta^{(\lambda_t)}, \theta^{(\lambda_t)}),$$
$$\Delta_t := f(X\beta^{(\lambda_t)}) - f(\nabla f^*(z_t)) \text{ for } z_t := -\lambda_t \theta^{(\lambda_t)}.$$

Our analysis is based on bounds on the duality gap for unseen values of the regularization parameter λ.

Lemma 4 (Bounding the Warm Start Error)**.** *Assume* $-\lambda\theta^{(\lambda_t)} \in \text{dom}(f^*)$ *and* $X^\top \theta^{(\lambda_t)} \in \text{dom}(\Omega^*)$. *For* $\rho = 1 - \lambda/\lambda_t$,

$$\text{Gap}_t + \rho \cdot (\Delta_t - \text{Gap}_t) + \mathcal{U}_{f^*,z_t}(\rho \cdot z_t) \leq \text{Gap}_\lambda(\beta^{(\lambda_t)}, \theta^{(\lambda_t)}) \leq \text{Gap}_t + \rho \cdot (\Delta_t - \text{Gap}_t) + \mathcal{V}_{f^*,z_t}(\rho \cdot z_t).$$

Denoting the upper bound $Q_{t,\mathcal{V}_{f^*}}(\rho) = \text{Gap}_t + \rho \cdot (\Delta_t - \text{Gap}_t) + \mathcal{V}_{f^*,z_t}(\rho \cdot z_t)$, we can define an algorithm (Algorithm 10) that returns an ϵ_{p}-Path, that is, for each $\lambda \in [\lambda_{\min}, \lambda_{\max}]$ a $\beta^{(\lambda)}$ such that $f(X\beta^{(\lambda)}) + \lambda\Omega(\beta^\lambda) \leq \epsilon_{\text{p}} + f(X\hat{\beta}^{(\lambda)}) + \lambda\Omega(\hat{\beta}^\lambda)$.

We then provide a complexity analysis along with optimality guarantees. We discuss the relationship between the regularity of the loss function and the complexity of the approximation path. We prove that its complexity is $O(1/\sqrt[d]{\epsilon})$ for uniformly convex loss of order $d > 0$ i.e. uniformly convex with modulus $\mu\|\cdot\|^d/d$ ($\mu > 0$) and $O(1/\sqrt{\epsilon})$ for the logistic loss thanks to a refined measure of its curvature throughout its Generalized Self-Concordant properties [STD17]. Finally, we provide an algorithm, with global convergence property, for selecting a hyperparameter with a validation error ϵ_v-close to the best possible hyparameter on a given range.

4.4 Joint Quantile regression

Given a couple (X, Y) of random variables, where Y takes scalar values, a common aim in statistics and machine learning is to estimate the conditional expectation $\mathbb{E}[Y \mid X = x]$ as a function of x.

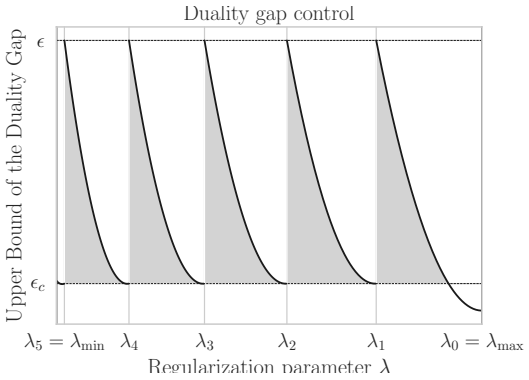

Figure 4.3: Illustration of ϵ-path for the Lasso at accuracy $\epsilon = 10^{-2}$.

In the previous setting, called regression, one assumes that the main information in Y is a scalar value corrupted by a centered noise. However, in some applications such as medicine, economics, social sciences and ecology, a more complete picture than an average relationship is required to deepen the analysis. Quantiles are natural quantities able to achieve this goal.

Since quantiles of a distribution are closely related, joint quantile regression is subsumed under the field of multi-task learning [AEP08]. As a consequence, vector-valued kernel methods are appropriate for such a task. They have already been used for various applications, such as structured classification [DOGP11] and prediction [BSdB16], manifold regularization [MBM16, BdBS11] and functional regression [KDP+15]. Quantile regression is a new opportunity for vector-valued reproducing kernel Hilbert spaces to perform in a multi-task problem, along with a loss that is different from the ℓ_2 cost predominantly used in the previous references.

Let $\mathcal{Y} \subset \mathbb{R}$ be a compact set, $\mathcal{X}$ be an arbitrary input space and $(X, Y) \in \mathcal{X} \times \mathcal{Y}$ a pair of random variables following an unknown joint distribution. For a given probability $\tau \in (0, 1)$, the conditional τ-quantile of (X, Y) is the function $\mu_\tau \colon \mathcal{X} \to \mathbb{R}$ such that $\mu_\tau(x) = \inf\{\mu \in \mathbb{R} : \mathbb{P}Y \leq \mu X = x \geq \tau\}$. Thus, given a training set $\{(x_i, y_i)\}_{i=1}^n \in (\mathcal{X} \times \mathcal{Y})^n$, the quantile regression problem aims at estimating this conditional τ-quantile function μ_τ. Following [Koe05], this can be achieved by minimization of the *pinball loss*: $\ell_\tau(r) = \max(\tau r, (\tau - 1)r)$, where $r \in \mathbb{R}$ is a residual. Using such a loss first arose from the observation that the location parameter μ that minimizes the ℓ_1-loss $\sum_{i=1}^n |y_i - \mu|$ is an estimator of the unconditional median [KB78].

Now focusing on the estimation of a conditional quantile, one can show that the target function μ_τ is a minimizer of the τ-quantile risk $R_\tau(h) = \mathbb{E}[\ell_\tau(Y - h(X))]$ [LLZ07]. However, since the joint probability of (X, Y) is unknown but we are provided with an i.i.d sample of observations $\{(x_i, y_i)\}_{i=1}^n$, we resort to minimizing the empirical risk, in the same fashion as least squares correspond to the estimation of conditional expectation. We thus wish to minimize

$$R_\tau^{\text{emp}}(h) = \frac{1}{n} \sum_{i=1}^n \ell_\tau(y_i - h(x_i)),$$

within a class $\mathcal{H} \subset (\mathbb{R})^{\mathcal{X}}$ of functions, calibrated in order to overcome the shift from the true risk to the empirical one. In particular, when $\mathcal{H}$ has the form: $\mathcal{H} = \{h = f + b : b \in \mathbb{R}, f \in (\mathbb{R})^{\mathcal{X}}, \psi(f) \leq c\}$, with $\psi \colon (\mathbb{R})^{\mathcal{X}} \to \mathbb{R}$ being a convex function and $c > 0$ a constant, [TLSS06] proved that (similarly to the unconditional case) the *quantile property* is satisfied: for any estimator $\hat{h}$, obtained by minimizing R_τ^{emp} in $\mathcal{H}$, the ratio of observations lying below $\hat{h}$ (i.e. $y_i < \hat{h}(x_i)$) equals τ to a small error (the ration of observations exactly equal to $\hat{h}(x_i)$). Moreover, under some regularity assumptions, this quantity converges to τ when the sample grows. Note that these properties are true since the intercept b is unconstrained.

In many real problems (such as medical reference charts), one is not only interested by estimating a single quantile curve but a few of them. Thus, denoting $[p]$ the range of integers between 1 and p, for several quantile levels τ_j ($j \in [p]$) and functions $h_j \in \mathcal{H}$, the empirical loss to be minimized can be written as the following separable function: $R_{\boldsymbol{\tau}}^{\text{emp}}(h_1, \ldots, h_p) = \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^p \ell_{\tau_j}(y_i - h_j(x_i))$, where $\boldsymbol{\tau}$ denotes the p dimensional vector of quantile levels.

A nice feature of multiple quantile regression is thus to extract slices of the conditional distribution of $Y|X$. However, when quantiles are estimated independently, an embarrassing phenomenon often appears: quantile functions cross, thus violating the basic principle that the cumulative distribution function should be monotonically non-decreasing. We refer to that pitfall as the *crossing problem*.

In [SFdB16], we propose to prevent curve crossing by considering the problem of multiple quantile regression as a vector-valued regression problem where outputs are not independent. An interesting feature of our method is to preserve the quantile property while most other approaches lose it when struggling to the crossing problem. Our approach is based on defining a matrix-valued kernel $K\colon (x, x') \mapsto k(x, x')\mathbf{B}$, where $k\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a scalar-valued kernel and $\mathbf{B}$ is a $p \times p$ positive definite matrix. The matrix $\mathbf{B}$ encodes the relationship between the components f_j and thus, the link between the different conditional quantile estimators. A rational choice is to consider $k(x, x') = \exp(-\|x - x'\|^2/2\sigma^2)$ and $\mathbf{B} = \big(\exp(-\gamma(\tau_i - \tau_j)^2)\big)_{1\leq i,j\leq p}$. We then define the reproducing kernel Hilbert space associated to K [MP05] with a norm denoted $\|\cdot\|_{\mathcal{K}}$ and choose $\psi = \|\cdot\|_{\mathcal{K}}$ in the definition of our class $\mathcal{H}$ of functions.

Quantile estimation, as presented in this paper, comes down to minimizing a regularized empirical risk, defined by the pinball loss $\ell_{\boldsymbol{\tau}}$. Since this loss function is non-differentiable, we introduce slack variables ξ and ξ^* to get the following primal formulation. On top of the parameters σ and γ of the kernel, we also consider a regularization parameter C to be tuned:

$$\min_{\substack{f\in\mathcal{K}_K, b\in\mathbb{R}^p,\\ \xi,\xi^*\in(\mathbb{R}^p)^n}} \frac{1}{2}\|f\|_{\mathcal{K}}^2 + C\sum_{i=1}^{n}(\langle\boldsymbol{\tau}, \xi_i\rangle_{\ell_2} + \langle\mathbb{1} - \boldsymbol{\tau}, \xi_i^*\rangle_{\ell_2}) \quad \text{s.t.} \quad \begin{cases}\forall i \in [n]\colon \xi_i \geq 0, \xi_i^* \geq 0,\\ y_i - f(x_i) - b = \xi_i - \xi_i^*.\end{cases}$$

A dual formulation of this problem is:

$$\min_{\alpha\in(\mathbb{R}^p)^n} \frac{1}{2}\sum_{i,j=1}^{n}\langle\alpha_i, K(x_i, x_j)\alpha_j\rangle_{\ell_2} - \sum_{i=1}^{n} y_i\langle\alpha_i, \mathbb{1}\rangle_{\ell_2} \quad \text{s.t.} \quad \begin{cases}\sum_{i=1}^{n}\alpha_i = 0_{\mathbb{R}^p},\\ \forall i \in [n]\colon C(\boldsymbol{\tau} - \mathbb{1}) \leq \alpha_i \leq C\boldsymbol{\tau},\end{cases}$$

We chose to solve this dual problem using the primal-dual coordinate descent method Algorithm 6. Indeed, except for the p linear equality constraints, the problem is very close to dual support vector machines. As shown on Table 4.1, our algorithmic choice is competitive against interior point based QP solvers and the Augmented Lagrangian method, especially for large scale problems

Table 4.1: CPU time (s) for training a randomly generated model

Size $n \times p$	QP	Aug. Lag.	PDCD
250×9	**8.73** ± 0.34	261.11 ± 46.69	18.69 ± 3.54
500×9	75.53 ± 2.98	865.86 ± 92.26	**61.30** ± 7.05
1000×9	621.60 ± 30.37	–	**266.50** ± 41.16
2000×9	3416.55 ± 104.41	–	**958.93** ± 107.80

Finally, in [SFdB17], by adding a group-lasso term $\epsilon\sum_{i=1}^{n}\|\alpha_i\|_{\ell_2}$ to the dual formulation, we could define an ϵ-insensitive version of the quantile regression problem. We can still use primal-dual coordinate descent to solve the model and we showed that we can obtain estimations with similar quality with the benefit of sparse solutions α. Hence, the model is easier to interpret, store and take less time to use when applied on new observations.

Chapter 5

Perspectives

According to me, the optimization community has been driven by three main research agendas:

- Write optimization models that describe practical issues. There is a compromise to make between the description power of the models and our ability to solve them.

- Discover new algorithmic paradigms that allow us to solve new, more complex models. We may cite primal-dual methods that avoid projecting onto complex sets, stochastic gradient methods that can replace the computation of an integral by the generation of random samples or line search that helps taking profit of the local smoothness of the objective function.

- Optimize the optimization algorithms available in order to be quicker and be able to solve larger instances. The tremendous amount of work put for this purpose has made optimization solvers successful on a large variety of problems.

There are still major unresolved challenges in nonconvex optimization and for the resolution of problems in large dimension. I would like to add my contribution as follows.

Primal-dual methods with optimal rate of convergence

State-of-the-art primal-dual methods have either a $O(1/k)$ rate for weakly convex functions but are rather slow in practice or they exhibit a linear rate for well behaved function but have a worst case $O(1/\sqrt{k})$ rate. My goal is to design new methods that would be efficient in both regimes. This may require to combine techniques from primal-dual algorithms and restart of accelerated methods as well as deriving new bounds for the purpose of the study.

Stochastic gradient methods for constrained convex optimization

When solving an optimization problem where the objective can be written as an expectation, stochastic gradient methods may be the preferred alternative against coordinate descent. The study of stochastic gradient for saddle point problems has only begun very recently. I plan to base on our smoothing technique to design new efficient algorithms in this context.

Stochastic compressed second order methods

Optimization algorithms using Hessians usually require much less iterations than those using gradient only, but each iteration is much more computationally intensive. We aim at developing efficient stochastic second order methods by extracting curvature information through randomized compression. This will alleviate both the cost of calculating the second order derivative matrix and the cost of manipulating the resulting matrix since it will be low rank.

Generative models for nonconvex optimization

This line of research aims at tackling nonconvex optimization problems using a generative model. The goal is design an optimization method that takes profit of a randomness to explore the set of parameters but is also able to focus on local mimima. The generative model will be trained on a set of optimization problems: the choice of training problems is critical to get good performance in practice. There is also a lot of freedom on how we parametrize the algorithm. We plan to focus on neural network architectures that have a lot of flexibility.

Analog-to-feature converter for smart sensors

This interdisciplinary project will put machine learning concerns at the heart of the design of sensors. The goal is to extract the features of the signal before digitizing it. We expect massive computational and energy gains through the use of a dedicated architecture, that should be optimized for our purpose.

Bibliography

[AC10] S. Arlot and A. Celisse. A survey of cross-validation procedures for model selection. *Statistics surveys*, 4:40–79, 2010.

[ADFC17] Ahmet Alacaoglu, Quoc Tran Dinh, Olivier Fercoq, and Volkan Cevher. Smooth primal-dual coordinate descent algorithms for nonsmooth convex optimization. In *Advances in Neural Information Processing Systems*, pages 5852–5861, 2017.

[AEP08] A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008.

[Ant10] A. Antoniadis. Comments on: ℓ_1-penalization for mixture regression models. *TEST*, 19(2):257–258, 2010.

[Aus76] Alfred Auslender. *Optimisation: Méthodes Numériques*. Masson, Paris, 1976.

[AZO15] Zeyuan Allen-Zhu and Lorenzo Orecchia. Nearly-Linear Time Packing and Covering LP Solver with Faster Convergence Rate Than $O(1/\varepsilon^2)$. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing (STOC 2015)*, 2015.

[Bac13] Francis Bach. Learning with Submodular Functions: A Convex Optimization Perspective. *Foundations and Trends® in Machine Learning*, 6(2-3):145–373, 2013.

[BBG09] Antoine Bordes, Léon Bottou, and Patrick Gallinari. SGD-QN: Careful quasi-Newton stochastic gradient descent. *JMLR*, 10:1737–1754, 2009.

[BC13] A. Belloni and V. Chernozhukov. Least squares after model selection in high-dimensional sparse models. *Bernoulli*, 19(2):521–547, 2013.

[BCG11] S. R. Becker, E. J. Candès, and M. C. Grant. Templates for convex cone problems with applications to sparse signal recovery. *Math. Program. Comput.*, 3(3):165–218, 2011.

[BCW11] A. Belloni, V. Chernozhukov, and L. Wang. Square-root Lasso: Pivotal recovery of sparse signals via conic programming. *Biometrika*, 98(4):791–806, 2011.

[BdBS11] C. Brouard, F. d'Alché Buc, and M. Szafranski. Semi-supervised Penalized Output Kernel Regression for Link Prediction. In *Proceedings of the 28th International Conference on Machine Learning*, 2011.

[Ber96] Dimitri P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods*. Athena Scientific, 1996.

[BERG14] Antoine Bonnefoy, Valentin Emiya, Liva Ralaivola, and Rémi Gribonval. A dynamic screening principle for the lasso. In *European Signal Processing Conference EUSIPCO*, 2014.

[BERG15] Antoine Bonnefoy, Valentin Emiya, Liva Ralaivola, and Rémi Gribonval. Dynamic Screening: Accelerating First-Order Algorithms for the Lasso and Group-Lasso. *IEEE Trans. Signal Process.*, 63(19):20, 2015.

[BH12] Radu Ioan Boţ and Christopher Hendrich. A variable smoothing algorithm for solving convex optimization problems. *TOP*, 23(1):124–150, 2012.

[BHF14] Pascal Bianchi, Walid Hachem, and Iutzeler Franck. A stochastic coordinate descent primal-dual algorithm and applications. In *2014 IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. IEEE, 2014.

[BHNS14] Richard H Byrd, Samantha L Hansen, Jorge Nocedal, and Yoram Singer. A stochastic quasi-Newton method for large-scale optimization. *arXiv:1401.7020*, 2014.

[BJM+12] Francis Bach, Rodolphe Jenatton, Julien Mairal, Guillaume Obozinski, et al. Convex optimization with sparsity-inducing norms. *Foundations and Trends in Machine Learning*, 4(1):1–106, 2012.

[BKBG11] Joseph K. Bradley, Aapo Kyrola, Danny Bickson, and Carlos Guestrin. Parallel Coordinate Descent for L1-Regularized Loss Minimization. In *28th International Conference on Machine Learning*, 2011.

[BPC+11a] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

[BPC+11b] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

[BRT09] P. J. Bickel, Y. Ritov, and A. B. Tsybakov. Simultaneous analysis of Lasso and Dantzig selector. *Ann. Statist.*, 37(4):1705–1732, 2009.

[BSdB16] C. Brouard, M. Szafranski, and F. d'Alché Buc. Input Output Kernel Regression: Supervised and Semi-Supervised Structured Output Prediction with Operator-Valued Kernels. *Journal of Machine Learning Research*, 17(176):1–48, 2016.

[BT89] Dimitri P Bertsekas and John N Tsitsiklis. *Parallel and distributed computation: numerical methods*. Prentice Hall, 1989.

[BT09] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[BT12] Amir Beck and Marc Teboulle. Smoothing and First Order Methods: A Unified Framework. *SIAM J. Optim.*, 22(2):557–580, 2012.

[BT13] Amir Beck and Luba Tetruashvili. On the convergence of block coordinate descent type methods. *SIAM journal on Optimization*, 23(4):2037–2060, 2013.

[BTN01] Aharon Ben-Tal and Arkadi Nemirovski. *Lectures on modern convex optimization: Analysis, algorithms, and engineering applications*, volume 3 of *MOS-SIAM Series on Optimization*. SIAM, 2001.

[BV04] S. Boyd and L. Vandenberghe. *Convex Optimization*. University Press, Cambridge, 2004.

[Bv11] P. Bühlmann and S. van de Geer. *Statistics for high-dimensional data*. Springer Series in Statistics. Springer, Heidelberg, 2011. Methods, theory and applications.

[CBS14] Volkan Cevher, Steffen Becker, and Martin Schmidt. Convex optimization for big data: Scalable, randomized, and parallel algorithms for big data analytics. *IEEE Signal Processing Magazine*, 31(5):32–43, 2014.

[CD11] S. Chrétien and S. Darses. Sparse recovery with unknown variance: a lasso-type approach. *IEEE Trans. Inf. Theory*, 2011.

[CERS18] Antonin Chambolle, Matthias J Ehrhardt, Peter Richtárik, and Carola-Bibiane Schönlieb. Stochastic primal-dual hybrid gradient algorithm with arbitrary sampling and imaging applications. *SIAM Journal on Optimization*, 28(4):2783–2808, 2018.

[Con13] Laurent Condat. A primal–dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms. *Journal of Optimization Theory and Applications*, 158(2):460–479, 2013.

[CP11] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

[CP15a] Antonin Chambolle and Thomas Pock. On the ergodic convergence rates of a first-order primal–dual algorithm. *Mathematical Programming*, pages 1–35, 2015.

[CP15b] Patrick L Combettes and Jean-Christophe Pesquet. Stochastic Quasi-Fejér Block-Coordinate Fixed Point Iterations with Random Sweeping. *SIAM Journal on Optimization*, 25(2):1221–1248, 2015.

[CRPW12] V. Chandrasekaranm, B. Recht, P. A. Parrilo, and A. S. Willsky. The convex Geometry of linear inverse problems. *Foundations of Computational Mathematics*, 12(6):805–849, 2012.

[CSS02] Michael Collins, Robert E Schapire, and Yoram Singer. Logistic regression, AdaBoost and Bregman distances. *Machine Learning*, 48(1-3):253–285, 2002.

[CZA08] Andrzej Cichocki, Rafal Zdunek, and Shun-ichi Amari. Nonnegative matrix and tensor factorization [lecture notes]. *IEEE signal processing magazine*, 25(1):142–145, 2008.

[DL16] D. Drusvyatskiy and A.S. Lewis. Error bounds, quadratic growth, and linear convergence of proximal methods. *arXiv:1602.06661*, 2016.

[DOGP11] F. Dinuzzo, C.S. Ong, P. Gehler, and G. Pillonetto. Learning Output Kernels with Block Coordinate Descent. In *Proceedings of the 28th International Conference of Machine Learning*, 2011.

[Dro14] Yoel Drori. *Contributions to the complexity analysis of optimization algorithms*. PhD thesis, University of Tel Aviv, 2014.

[DY15] Damek Davis and Wotao Yin. A three-operator splitting scheme and its optimization applications. *arXiv preprint arXiv:1504.01032*, 2015.

[DY17] Damek Davis and Wotao Yin. Faster convergence rates of relaxed Peaceman-Rachford and ADMM under regularity assumptions. *Mathematics of Operations Research*, 2017.

[EHJT04] B. Efron, T. Hastie, I. M. Johnstone, and R. Tibshirani. Least angle regression. *Ann. Statist.*, 32(2):407–499, 2004. With discussion, and a rejoinder by the authors.

[EN15] Alina Ene and Huy L Nguyen. Random coordinate descent methods for minimizing decomposable submodular functions. In *32nd International Conference on Machine Learning*, 2015.

[EVR12] L. El Ghaoui, V. Viallon, and T. Rabbani. Safe feature elimination in sparse supervised learning. *J. Pacific Optim.*, 8(4):667–698, 2012.

[FB19] Olivier Fercoq and Pascal Bianchi. A coordinate-descent primal-dual algorithm with large step size and possibly nonseparable functions. *SIAM Journal on Optimization*, 29(1):100–134, 2019.

[FGH12] J. Fan, S. Guo, and N. Hao. Variance estimation using refitted cross-validation in ultrahigh dimensional regression. *J. Roy. Statist. Soc. Ser. B*, 74(1):37–65, 2012.

[FGS15] Olivier Fercoq, Alexandre Gramfort, and Joseph Salmon. Mind the duality gap: safer rules for the lasso. In *ICML*, pages 333–342, 2015.

[FHHT07] Jerome Friedman, Trevor Hastie, Holger Höfling, and Robert Tibshirani. Pathwise coordinate optimization. *Ann. Appl. Stat.*, 1(2):302–332, 2007.

[FHT10] J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1, 2010.

[FL08] J. Fan and J. Lv. Sure independence screening for ultrahigh dimensional feature space. *J. Roy. Statist. Soc. Ser. B*, 70(5):849–911, 2008.

[FQ16] Olivier Fercoq and Zheng Qu. Restarting accelerated gradient methods with a rough strong convexity estimate. *arXiv preprint arXiv:1609.07358*, 2016.

[FQ17] Olivier Fercoq and Zheng Qu. Adaptive restart of accelerated gradient methods under local quadratic growth condition. *arXiv preprint arXiv:1709.02300*, 2017.

[FQ18] Olivier Fercoq and Zheng Qu. Restarting the accelerated coordinate descent method with a rough strong convexity estimate. *arXiv preprint arXiv:1803.05771*, 2018.

[FQRT14] Olivier Fercoq, Zheng Qu, Peter Richtárik, and Martin Takáč. Fast distributed coordinate descent for minimizing non-strongly convex losses. In *IEEE International Workshop on Machine Learning for Signal Processing*, 2014.

[FR14] Olivier Fercoq and Peter Richtárik. Universal coordinate descent and line search for parallel coordinate descent. Technical report, The University of Edinburgh, 2014.

[FR15] Olivier Fercoq and Peter Richtárik. Accelerated, parallel and proximal coordinate descent. *SIAM Journal on Optimization*, 25(4):1997–2023, 2015.

[FR17] Olivier Fercoq and Peter Richtárik. Smooth minimization of nonsmooth functions with parallel coordinate descent methods. In *Modeling and Optimization: Theory and Applications*, pages 57–96. Springer, 2017.

[FT15] Kimon Fountoulakis and Rachael Tappenden. A flexible coordinate descent method for big data applications. *arXiv preprint arXiv:1507.03713*, 2015.

[Fu98] W. J. Fu. Penalized regressions: the bridge versus the lasso. *J. Comput. Graph. Statist.*, 7(3):397–416, 1998.

[Gab83] Daniel Gabay. Chapter ix applications of the method of multipliers to variational inequalities. *Studies in mathematics and its applications*, 15:299–331, 1983.

[GB14] P. Giselsson and S. Boyd. Monotonicity and Restart in Fast Gradient Methods. In *IEEE Conference on Decision and Control*, Los Angeles, USA, December 2014. CDC.

[GJL10] J. Giesen, M. Jaggi, and S. Laue. Approximating parameterized convex optimization problems. In *European Symposium on Algorithms*, pages 524–535, 2010.

[GJM12] B. Gärtner, M. Jaggi, and C. Maria. An exponential lower bound on the complexity of regularization paths. *Journal of Computational Geometry*, 2012.

[GM75] Roland Glowinski and A Marroco. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de Dirichlet non linéaires. *Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique*, 9(2):41–76, 1975.

[GM76] Daniel Gabay and Bertrand Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1):17–40, 1976.

[GMLS12] J. Giesen, J. K. Müller, S. Laue, and S. Swiercy. Approximating concavely parameterized optimization problems. In *NIPS*, 2012.

[GXZ19] Xiang Gao, Yang-Yang Xu, and Shu-Zhong Zhang. Randomized primal–dual proximal block coordinate updates. *Journal of the Operations Research Society of China*, 7(2):205–250, 2019.

[HRTZ04] T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu. The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, 5:1391–1415, 2004.

[Hub81] P. J. Huber. *Robust Statistics*. John Wiley & Sons Inc., 1981.

[HY12] Bingsheng He and Xiaoming Yuan. Convergence analysis of primal-dual algorithms for a saddle-point problem: from contraction perspective. *SIAM Journal on Imaging Sciences*, 5(1):119–149, 2012.

[IBCH13] Franck Iutzeler, Pascal Bianchi, Philippe Ciblat, and Walid Hachem. Asynchronous distributed optimization using a randomized alternating direction method of multipliers. In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pages 3671–3676. IEEE, 2013.

[JG15] T. B. Johnson and C. Guestrin. Blitz: A principled meta-algorithm for scaling sparse optimization. In *ICML*, pages 1171–1179, 2015.

[KB78] R. Koenker and G. Bassett. Regression quantiles. *Econometrica*, 46(1):33–50, 1978.

[KDP+15] H. Kadri, E. Duflos, P. Preux, S. Canu, A. Rakotomamonjy, and J. Audiffren. Operator-valued Kernels for Learning from Functional Response Data. *Journal of Machine Learning Research*, 16:1–54, 2015.

[KKB07] K. Koh, SJ. Kim, and S. Boyd. An interior-point method for large-scale l1-regularized logistic regression. *J. Mach. Learn. Res.*, 8(8):1519–1555, 2007.

[Koe05] R. Koenker. *Quantile Regression*. Cambridge University Press, Cambridge, New York, 2005.

[KWGA11] M. Kowalski, P. Weiss, A. Gramfort, and S. Anthoine. Accelerating ISTA with an active set strategy. In *OPT 2011: 4th International Workshop on Optimization for Machine Learning*, page 7, 2011.

[Led13] J. Lederer. Trust, but verify: benefits and pitfalls of least-squares refitting in high dimensions. *arXiv preprint arXiv:1306.0113*, 2013.

[LL10] Dennis Leventhal and Adrian Lewis. Randomized Methods for Linear Constraints: Convergence Rates and Conditioning. *Mathematics of Operations Research*, 35(3):641–654, 2010.

[LLX14] Qihang Lin, Zhaosong Lu, and Lin Xiao. An accelerated proximal coordinate gradient method. In *Advances in Neural Information Processing Systems*, pages 3059–3067, 2014.

[LLX15] Qihang Lin, Zhaosong Lu, and Lin Xiao. An accelerated randomized proximal coordinate gradient method and its application to regularized empirical risk minimization. *SIAM Journal on Optimization*, 25(4):2244–2273, 2015.

[LLZ07] Y. Li, Y. Liu, and J. Zhu. Quantile Regression in Reproducing Kernel Hilbert Spaces. *Journal of the American Statistical Association*, 102(477):255–268, 2007.

[LMH15] Hongzhou Lin, Julien Mairal, and Zaid Harchaoui. A universal catalyst for first-order optimization. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 3384–3392. Curran Associates, Inc., 2015.

[LMon] G. Lan and R.D.C. Monteiro. "Iteration-complexity of first-order augmented Lagrangian methods for convex programming. *Math. Program.*, 2013 (under revision).

[LS13] Yin Tat Lee and Aaron Sidford. Efficient accelerated coordinate descent methods and faster algorithms for solving linear systems. pages 147–156, 2013.

[LSS12] Jason D Lee, Yuekai Sun, and Michael Saunders. Proximal newton-type methods for convex optimization. In *Advances in Neural Information Processing Systems*, pages 827–835, 2012.

[LT02] Zhi Quan Luo and Paul Tseng. A coordinate gradient descent method for nonsmooth separable minimization. *Journal of optimization theory and applications*, 72(1), January 2002.

[LWR+15] Ji Liu, Stephen J Wright, Christopher Ré, Victor Bittorf, and Srikrishna Sridhar. An asynchronous parallel stochastic coordinate descent algorithm. *The Journal of Machine Learning Research*, 16(1):285–322, 2015.

[LX15a] Qihang Lin and Lin Xiao. An adaptive accelerated proximal gradient method and its homotopy continuation for sparse optimization. *Computational Optimization and Applications*, 60(3):633–674, 2015.

[LX15b] Zhaosong Lu and Lin Xiao. On the complexity analysis of randomized block-coordinate descent methods. *Mathematical Programming*, 152(1-2):615–642, 2015.

[LY17] Mingrui Liu and Tianbao Yang. Adaptive accelerated gradient converging method under holderian error bound condition. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3104–3114. Curran Associates, Inc., 2017.

[Mai15] Julien Mairal. Incremental majorization-minimization optimization with application to large-scale machine learning. *SIAM Journal on Optimization*, 25(2):829–855, 2015.

[MBM16] H.Q. Minh, L. Bazzani, and V. Murino. A Unifying Framework in Vector-valued Reproducing Kernel Hilbert Spaces for Manifold Regularization and Co-Regularized Multi-view Learning. *Journal of Machine Learning Research*, 17(25):1–72, 2016.

[MBPS10] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *J. Mach. Learn. Res.*, pages 19–60, 2010.

[MCTD+14] M. B McCoy, V. Cevher, Q. Tran-Dinh, A. Asaei, and L. Baldassarre. Convexity in source separation: Models, geometry, and algorithms. *IEEE Signal Processing Magazine*, 31(3):87–95, 2014.

[MFGS17] Mathurin Massias, Olivier Fercoq, Alexandre Gramfort, and Joseph Salmon. Heteroscedastic concomitant lasso for sparse multimodal electromagnetic brain imaging. In *Proc. of the 21st International Conference on Artificial Intelligence and Statistics*, volume 1050, page 27. 2017.

[MP05] C.A. Micchelli and M.A. Pontil. On Learning Vector-Valued Functions. *Neural Computation*, 17:177–204, 2005.

[MRT15] Jakub Mareček, Peter Richtárik, and Martin Takáč. Distributed block coordinate descent for minimizing partially separable functions. In *Numerical Analysis and Optimization, Springer Proceedings in Mathematics and Statistics*, pages 261–288, 2015.

[MS12] R.D.C. Monteiro and B.F. Svaiter. Iteration-complexity of block-decomposition algorithms and the alternating direction method of multipliers. *Tech. Report.*, xx:xx, 2012.

[MY12] J. Mairal and B. Yu. Complexity analysis of the lasso regularization path. In *ICML*, 2012.

[NC13] Ion Necoara and Dragos Clipici. Efficient parallel coordinate descent algorithm for convex optimization problems with separable constraints: application to distributed MPC. *Journal of Process Control*, 23:243–253, 2013.

[Nes83] Yurii Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983.

[Nes05a] Y. Nesterov. Excessive gap technique in nonsmooth convex minimization. *SIAM J. Optimization*, 16(1):235–249, 2005.

[Nes05b] Yurii Nesterov. Smooth minimization of nonsmooth functions. *Mathematical Programming*, 103(1):127–152, 2005.

[Nes12a] Yurii Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

[Nes12b] Yurii Nesterov. Subgradient Methods for Huge-Scale Optimization Problems. *CORE DISCUSSION PAPER 2012/2*, 2012.

[Nes13a] Yurii Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.

[Nes13b] Yurii Nesterov. Universal Gradient Methods for Convex Optimization Problems. *Core discussion paper*, 2013.

[NFG+17] Eugene Ndiaye, Olivier Fercoq, Alexandre Gramfort, Vincent Leclère, and Joseph Salmon. Efficient smoothed concomitant lasso estimation for high dimensional regression. In *Journal of Physics: Conference Series*, volume 904, page 012006. IOP Publishing, 2017.

[NFGS15] Eugène Ndiaye, Olivier Fercoq, Alexandre Gramfort, and Joseph Salmon. GAP safe screening rules for sparse multi-task and multi-class models. *NIPS*, pages 811–819, 2015.

[NFGS16] Eugène Ndiaye, Olivier Fercoq, Alexandre Gramfort, and Joseph Salmon. GAP safe screening rules for Sparse-Group Lasso. *NIPS*, 2016.

[NFGS17] Eugene Ndiaye, Olivier Fercoq, Alexandre Gramfort, and Joseph Salmon. Gap safe screening rules for sparsity enforcing penalties. *The Journal of Machine Learning Research*, 18(1):4671–4703, 2017.

[NFS+18] Eugène Ndiaye, Olivier Fercoq, Joseph Salmon, Ichiro Takeuchi, and Le Tam. Safe grid search with optimal complexity. Technical report, Télécom ParisTech and Nagoya Institute of Technology, 2018.

[NNG12] Ion Necoara, Yurii Nesterov, and Francois Glineur. Efficiency of randomized coordinate descent methods on optimization problems with linearly coupled constraints. Technical report, Politehnica University of Bucharest, 2012.

[NW06] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, 2 edition, 2006.

[OC12] Brendan O'Donoghue and Emmanuel Candes. Adaptive restart for accelerated gradient schemes. *Foundations of Computational Mathematics*, pages 1–18, 2012.

[OPT00] M. R. Osborne, B. Presnell, and B. A. Turlach. A new approach to variable selection in least squares problems. *IMA J. Numer. Anal.*, 20(3):389–403, 2000.

[Owe07] A. B. Owen. A robust hybrid of lasso and ridge regression. *Contemporary Mathematics*, 443:59–72, 2007.

[PH07] M. Y. Park and T. Hastie. L1-regularization path algorithm for generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4):659–677, 2007.

[Pow73] Michael JD Powell. On search directions for minimization algorithms. *Mathematical programming*, 4(1):193–201, 1973.

[PR15] Jean-Christophe Pesquet and Audrey Repetti. A class of randomized primal-dual algorithms for distributed optimization. *Journal of Nonlinear Convex Analysis*, 16(12), 2015.

[PW14] Mert Pilanci and Martin J. Wainwright. Iterative Hessian sketch: fast and accurate solution approximation for constrained least-squares. *arXiv:1411.0347*, 2014.

[QR16] Zheng Qu and Peter Richtárik. Coordinate descent with arbitrary sampling I: algorithms and complexity. *Optimization Methods and Software*, 2016.

[QRTF16] Zheng Qu, Peter Richtárik, Martin Takác, and Olivier Fercoq. Sdna: stochastic dual newton ascent for empirical risk minimization. In *International Conference on Machine Learning*, pages 1823–1832, 2016.

[QRZ15] Zheng Qu, Peter Richtárik, and Tong Zhang. Quartz: Randomized Dual Coordinate Ascent with Arbitrary Sampling. In *NIPS 28*, pages 865–873. 2015.

[RKB04] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics (TOG)*, 23(3):309–314, 2004.

[Roc76a] R T Rockafellar. Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Mathematics of Operations Research*, 1:97–116, 1976.

[Roc76b] R T Rockafellar. Monotone operators and the proximal point algorithm. *SIAM Journal on Control and Optimization*, 14:877–898, 1976.

[ROG+16] A. Raj, J. Olbrich, B. Gärtner, B. Schölkopf, and M. Jaggi. Screening rules for convex problems. *arXiv preprint arXiv:1609.07478*, 2016.

[RT14] Peter Richtárik and Martin Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming, Ser. A*, 144(1-2):1–38, 2014.

[RT15] Peter Richtárik and Martin Takáč. Parallel coordinate descent methods for big data optimization. *Mathematical Programming*, pages 1–52, 2015.

[RT16a] Peter Richtárik and Martin Takáč. Distributed coordinate descent method for learning with big data. *The Journal of Machine Learning Research*, 17(1):2657–2681, 2016.

[RT16b] Peter Richtárik and Martin Takáč. On optimal probabilities in stochastic coordinate descent methods. *Optimization Letters*, 10(6):1233–1243, 2016.

[RT16c] Peter Richtárik and Martin Takáč. Distributed coordinate descent method for learning with big data. *The Journal of Machine Learning Research*, 17(1):2657–2681, 2016.

[RTF13] S. Reid, R. Tibshirani, and J. Friedman. A study of error variance estimation in lasso regression. *arXiv preprint arXiv:1311.5274*, 2013.

[RZ07] S. Rosset and J. Zhu. Piecewise linear regularized solution paths. *The Annals of Statistics*, pages 1012–1030, 2007.

[SBC14] W. Su, S. Boyd, and E. Candes. A differential equation for modeling Nesterov's accelerated gradient method: Theory and insights. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2510–2518, 2014.

[SBv10] N. Städler, P. Bühlmann, and S. van de Geer. ℓ_1-penalization for mixture regression models. *TEST*, 19(2):209–256, 2010.

[SDPG14] Jascha Sohl-Dickstein, Ben Poole, and Surya Ganguli. Fast large-scale optimization by unifying stochastic gradient and quasi-Newton methods. In *ICML*, 2014.

[Sei74] Ludwig Seidel. Ueber ein Verfahren, die Gleichungen, auf welche die Methode der kleinsten Quadrate führt, sowie lineäre Gleichungen überhaupt, durch successive Annäherung aufzulösen:(Aus den Abhandl. dk bayer. Akademie II. Cl. XI. Bd. III. Abth. *Verlag der k. Akademie*, 1874.

[SFdB16] Maxime Sangnier, Olivier Fercoq, and Florence d'Alché Buc. Joint quantile regression in vector-valued rkhss. In *Advances in Neural Information Processing Systems*, pages 3693–3701, 2016.

[SFdB17] Maxime Sangnier, Olivier Fercoq, and Florence d'Alché Buc. Data sparse nonparametric regression with ε-insensitive losses. In *Asian Conference on Machine Learning*, pages 192–207, 2017.

[SKHT16] A. Shibagaki, M. Karasuyama, K. Hatano, and I. Takeuchi. Simultaneous safe screening of features and samples in doubly sparse modeling. In *ICML*, pages 1577–1586, 2016.

[SSBD14] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, New York, NY, USA, 2014.

[SSKT15] A. Shibagaki, Y. Suzuki, M. Karasuyama, and I. Takeuchi. Regularization path of cross-validation error lower bounds. In *NIPS*, 2015.

[SST11] Shai Shalev-Shwartz and Ambuj Tewari. Stochastic methods for ℓ_1-regularized loss minimization. *Journal of Machine Learning Research*, 12:1865–1892, 2011.

[SSZ12] Shai Shalev-Shwartz and Tong Zhang. Proximal Stochastic Dual Coordinate Ascent. *arXiv:1211.2717*, 2012.

[SSZ13] Shai Shalev-Shwartz and Tong Zhang. Stochastic Dual Coordinate Ascent Methods for Regularized Loss Minimization. *Journal of Machine Learning Research*, 14(1):567–599, 2013.

[SSZ14] Shai Shalev-Shwartz and Tong Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. In *International Conference on Machine Learning*, pages 64–72, 2014.

[STD17] Tianxiao Sun and Quoc Tran-Dinh. Generalized self-concordant functions: a recipe for Newton-type methods. *Mathematical Programming*, pages 1–69, 2017.

[STY16] Defeng Sun, Kim-Chuan Toh, and Liuqin Yang. An efficient inexact ABCD method for least squares semidefinite programming. *SIAM Journal on Optimization*, 26(2):1072–1100, 2016.

[Suz14] Taiji Suzuki. Stochastic dual coordinate ascent with alternating direction method of multipliers. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 736–744, 2014.

[SYG07] Nicol N. Schraudolph, Jin Yu, and Simon Günter. A stochastic quasi-Newton method for online convex optimization. In *AISTATS*, pages 433–440, 2007.

[SZ12] T. Sun and C.-H.. Zhang. Scaled sparse linear regression. *Biometrika*, 99(4):879–898, 2012.

[TBF+12] R. Tibshirani, J. Bien, J. Friedman, T. Hastie, N. Simon, J. Taylor, and R. J. Tibshirani. Strong rules for discarding predictors in lasso-type problems. *J. Roy. Statist. Soc. Ser. B*, 74(2):245–266, 2012.

[TBRS13] Martin Takáč, Avleen Bijral, Peter Richtárik, and Nathan Srebro. Mini-batch primal and dual methods for SVMs. In *30th International Conference on Machine Learning*, 2013.

[TC14] Quoc Tran-Dinh and Volkan Cevher. A primal-dual algorithmic framework for constrained convex minimization. *arXiv preprint arXiv:1406.5403*, 2014.

[TDC14] Quoc Tran-Dinh and Volkan Cevher. Constrained convex minimization via model-based excessive gap. In *Proc. the Neural Information Processing Systems Foundation conference (NIPS2014)*, volume 27, pages 721–729, Montreal, Canada, December 2014.

[TDC15] Quoc Tran-Dinh and Volkan Cevher. An optimal first-order primal-dual gap reduction framework for constrained convex optimization. *Tech. Report, LIONS-EPFL*, pages 1–41, 2015.

[TDFC18] Quoc Tran-Dinh, Olivier Fercoq, and Volkan Cevher. A smooth primal-dual optimization framework for nonsmooth composite convex minimization. *SIAM Journal on Optimization*, 28(1):96–134, 2018.

[Tib96] R. Tibshirani. Regression shrinkage and selection via the lasso. *J. Roy. Statist. Soc. Ser. B*, 58(1):267–288, 1996.

[TKCW12] Qing Tao, Kang Kong, Dejun Chu, and Gaowei Wu. Stochastic Coordinate Descent Methods for Regularized Smooth and Nonsmooth Losses. *Machine Learning and Knowledge Discovery in Databases*, pages 537–552, 2012.

[TLSS06] I. Takeuchi, Q.V. Le, T.D. Sears, and A.J. Smola. Nonparametric Quantile Estimation. *Journal of Machine Learning Research*, 7:1231–1264, 2006.

[Tod01] Micheal J. Todd. Semidefinite optimization. *Acta Numerica*, 10:515–560, 2001.

[TRG16a] Rachael Tappenden, Peter Richtárik, and Jacek Gondzio. Inexact coordinate descent: complexity and preconditioning. *Journal of Optimization Theory and Applications*, pages 1–33, 2016.

[TRG16b] Rachael Tappenden, Peter Richtárik, and Jacek Gondzio. Inexact coordinate descent: complexity and preconditioning. *Journal of Optimization Theory and Applications*, 170(1):144–176, 2016.

[Tse01] Paul Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications*, 109(3):475–494, June 2001.

[Tse08] Paul Tseng. On accelerated proximal gradient methods for convex-concave optimization. *Submitted to SIAM Journal on Optimization*, 2008.

[TY09a] Paul Tseng and Sangwoon Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, 117:387–423, 2009.

[TY09b] Paul Tseng and Sangwoon Yun. Block-Coordinate Gradient Descent Method for Linearly Constrained Non Separable Optimization. *Journal of Optimization Theory and Applications*, 140:513–535, 2009.

[VB96] Lieven Vandenberghe and Stephen Boyd. Semidefinite Programming. *SIAM Review*, 38(1):49–95, 1996.

[VNFC17] Quang Van Nguyen, Olivier Fercoq, and Volkan Cevher. Smoothing technique for nonsmooth composite minimization with linear operator. *arXiv preprint arXiv:1706.05837*, 2017.

[Vũ13] Bang Công Vũ. A splitting algorithm for dual monotone inclusions involving cocoercive operators. *Advances in Computational Mathematics*, 38(3):667–681, 2013.

[Wai14] M. J. Wainwright. Structured regularizers for high-dimensional problems: Statistical and computational issues. *Annu. Rev. Stat. Appl.*, 1:233–253, 2014.

[War63] Jack Warga. Minimizing certain convex functions. *Journal of the Society for Industrial & Applied Mathematics*, 11(3):588–593, 1963.

[WSV10] Henry Wolkowicz, Romesh Saigal, and Lieven Vandenberghe, editors. *Handbook of Semidefinite Programming*, volume 27 of *International Series in Operations Research & Management Science*. Kluwer Academic Publishers, Boston, MA, 2010.

[WWY12] J. Wang, P. Wonka, and J. Ye. Lasso screening rules via dual polytope projection. *arXiv preprint arXiv:1211.3966*, 2012.

[WZL+14] J. Wang, J. Zhou, J. Liu, P. Wonka, and J. Ye. A safe screening rule for sparse logistic regression. In *NIPS*, pages 1053–1061, 2014.

[XCM10] H. Xu, C. Caramanis, and S. Mannor. Robust regression and lasso. *IEEE Trans. Inf. Theory*, 56(7):3561–3574, 2010.

[Xu17] Yangyang Xu. Accelerated first-order primal-dual proximal methods for linearly constrained composite convex programming. *SIAM Journal on Optimization*, 27(3):1459–1484, 2017.

[XWR14] Z. J. Xiang, Y. Wang, and P. J. Ramadge. Screening tests for lasso problems. *arXiv preprint arXiv:1405.4897*, 2014.

[XXR11] Z. J. Xiang, H. Xu, and P. J. Ramadge. Learning sparse representations of high dimensional data on large scale dictionaries. In *NIPS*, pages 900–908, 2011.

[YFLC18] Alp Yurtsever, Olivier Fercoq, Francesco Locatello, and Volkan Cevher. A conditional gradient framework for composite convex minimization with applications to semidefinite programming. In *International Conference on Machine Learning*, 2018.

[YL06] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *J. Roy. Statist. Soc. Ser. B*, 68(1):49–67, 2006.

[ZX14] Yuchen Zhang and Lin Xiao. Stochastic primal-dual coordinate method for regularized empirical risk minimization. *arXiv preprint arXiv:1409.3257*, 2014.