université
PARIS-SACLAY

Hyperparameter selection for high dimensional sparse learning: application to neuro-imaging

Sélection d'hyperparamètres pour l'apprentissage parcimonieux en grande dimension : application à la neuroimagerie

Thèse de doctorat de l'Université Paris-Saclay

École doctorale n° 580 sciences et technologies de l'information et de la communication (STIC)
Spécialité de doctorat : mathématiques et informatique
Unité de recherche : université Paris-Saclay, Inria, Inria Saclay-Île-de-France, 91120, Palaiseau, France
Référent : faculté des sciences d'Orsay

Thèse présentée et soutenue à Paris-Saclay, le 28 septembre 2021, par

Quentin BERTRAND

Composition du jury :

Jalal FADILI Professeur, ENSICAEN	Rapporteur
Massimiliano PONTIL Directeur de recherche, IIT Genova	Rapporteur
Carola-Bibiane SCHÖNLIEB Professeure, Université de Cambridge	Examinatrice
Karim LOUNICI Professeur, École polytechnique	Examinateur
Peter OCHS Professeur, Université de Tübingen	Examinateur

Direction de la thèse :

Joseph SALMON Professeur, Université de Montpellier	Codirecteur de thèse
Alexandre GRAMFORT Directeur de recherche, Inria	Codirecteur de thèse

Thèse de doctorat
NNT : 2021UPASG054

Hyperparameter selection for high dimensional sparse learning: application to neuro-imaging

by Quentin Bertrand

PhD dissertation

Université Paris-Saclay
Sciences et Technologies de l'Information et de la Communication

October 2018 - September 2021
Inria, Palaiseau, France

PH.D. COMMITTEE

DIRECTORS:
Alexandre Gramfort, Inria, Palaiseau, France
Joseph Salmon, Université de Montpellier, Montpellier, France

REVIEWERS:
Jalal Fadili, CNRS, ENSICAEN, Caen, France
Massimiliano Pontil, IIT, Genoa, Italy

EXAMINERS:
Carola-Bibiane Schönlieb, University of Cambridge, Cambridge, United-Kingdom
Karim Lounici, École polytechnique, Palaiseau, France
Peter Ochs, University of Tübingen, Tübingen, Germany

Contents

Résumé

Comprendre les maladies neurologiques modernes telles que la maladie d'Alzheimer ou de Parkinson apparaît comme un défi majeur pour lequel les neuroscientifiques s'appuient largement sur l'enregistrement et le traitement avancés de l'activité cérébrale. En raison de leur caractère non invasif et de leur excellente résolution temporelle, la magnéto- et l'électroencéphalographie (M/EEG) sont devenues des outils incontournables pour observer l'activité cérébrale. La reconstruction des signaux cérébraux à partir des mesures M/EEG peut être considérée comme un problème inverse en grande dimension mal posé. Les estimateurs classiques des signaux cérébraux sont basés sur la résolution de problèmes d'optimisation composites. Ces estimateurs basés sur la parcimonie ne sont actuellement pas massivement utilisés en neurosciences, principalement en raison de leurs hyperparamètres de régularisation notoirement difficiles à régler. Un des objectif de cette thèse est de fournir une méthode simple, rapide et automatique pour calibrer les modèles linéaires parcimonieux.

Nous étudions d'abord certaines propriétés de la descente par coordonnée, qui est un algorithme de pointe pour résoudre les problèmes d'optimisation composites «lisse + non lisse séparable». En s'appuyant sur la théorie du lissage partiel, nous montrons que la descente par coordonnée permet l'identification du modèle et la convergence linéaire locale. Ces propriétés sont ensuite utilisées dans cette thèse pour l'optimisation des hyperparamètres. Nous proposons également un schéma d'extrapolation d'Anderson, `andersoncd`, pour accélérer efficacement la descente par coordonnée en pratique.

Nous explorons ensuite une approche statistique pour définir l'hyperparamètre de régularisation des problèmes de type Lasso. Dans ce cas il existe une formule exacte pour l'hyperparamètre de régularisation optimal pour la régression linéaire parcimonieuse. Malheureusement, elle dépend du niveau de bruit réel, inconnu en pratique. Pour éliminer cette dépendance, on peut recourir à des estimateurs pour lesquels le paramètre de régularisation ne dépend pas du niveau de bruit. Cependant, ces derniers nécessitent de résoudre de difficiles problèmes d'optimisation «non lisse + non lisse». Nous montrons que le lissage partiel préserve leurs propriétés statistiques, tout en faisant que le problème d'optimisation puisse être résolu avec des techniques efficaces de descente par coordonnée. Des expériences approfondies sur des données réelles de M/EEG montrent l'intérêt de ces estimateurs sur des tâches visuelles et auditives.

Enfin, nous étudions la sélection d'hyperparamètres sous l'angle de l'optimisation à deux niveaux. Elle englobe les techniques de sélection d'hyperparamètres les plus populaires dans l'apprentissage automatique et les problèmes inverses comme le critère «hold-out», la validation croisée, ainsi que certains proxies de l'estimateur du risque sans biais de Stein. Cette approche repose sur des problèmes d'optimisation à deux niveaux avec des problèmes internes non lisses, qui sont généralement résolus avec des méthodes

d'ordre zéro, telles que la recherche sur grille ou la recherche aléatoire. Dans cette thèse, nous présentons un algorithme efficace pour résoudre ces problèmes d'optimisation à deux niveaux en utilisant des méthodes du premier ordre. Cela permet de calibrer efficacement des modèles parcimonieux avec un grand nombre d'hyperparamètres.

Afin de promouvoir la diffusion scientifique et la reproductibilité, tous les algorithmes développés dans cette thèse sont disponibles en ligne, avec des exemples et une documentation exhaustive. De plus, `andersoncd` a été implémenté dans la bibliothèque python la plus populaire de traitement du signal cérébral, `MNE`, et est maintenant l'algorithme par défaut pour le calcul des estimateurs de signaux parcimonieux. Les algorithmes d'optimisation à deux niveaux devraient bientôt être intégrés au module d'optimisation `jaxopt` de la bibliothèque de différentiation automatique `jax`.

Abstract

Tackling modern neurological diseases such as Alzheimer or Parkinson appears as a major challenge for which neuroscientists extensively rely on advanced recording and processing of brain activity. Due to non-invasiveness and excellent time resolution, magneto- and electroencephalography (M/EEG) have emerged as tools of choice to monitor brain activity. Reconstructing brain signals from M/EEG measurements can be cast as a high dimensional ill-posed inverse problem. Usual estimators of brain signals involve challenging composite optimization problems. Because of their notoriously hard to tune regularization hyperparameters, sparsity-based estimators are currently not massively used in neuroscience. The goal of this thesis is to provide a simple, fast, and automatic way to calibrate sparse linear models.

We first study some properties of coordinate descent, which is a state-of-art algorithm to solve composite "smooth + nonsmooth separable" optimization problem. Relying on the partial smoothness framework we show that coordinate descent achieves model identification and local linear convergence. These properties are latter used in this thesis for hyperparameter optimization. We also propose an Anderson extrapolation scheme, `andersoncd`, to accelerate coordinate descent in practice.

We then explore a statistical approach to set the regularization parameter of Lasso-type problems. A closed-form formula can be derived for the optimal regularization hyperparameter of sparse penalized linear regressions. Unfortunately, it relies on the true noise level, unknown in practice. To remove this dependency, one can resort to estimators for which the regularization hyperparameter does not depend on the noise level. However, they require to solve challenging "nonsmooth + nonsmooth" optimization problems. We show that partial smoothing preserves their statistical properties, while making the optimization problem amenable to efficient coordinate descent. Extensive experiments on real M/EEG data show the interest of these estimators on visual and auditory tasks.

Finally we investigate hyperparameter selection through the lens of bilevel optimization. It encompasses most popular hyperparameter selection techniques in machine learning and inverse problems: hold-out, cross-validation and proxies of the Stein unbiased risk estimator. This approach relies on bilevel optimization problems with nonsmooth inner problems, that are usually solved with zeros-order methods, such as grid-search or random-search. In this thesis we present an efficient algorithm to solve these bilevel optimization problems using first-order methods. This enabled to efficiently calibrate sparse models with large number of hyperparameters.

In order to promote scientific dissemination and reproducibility all the algorithms developed in this thesis are available online with examples and extensive documentation. In addition, `andersoncd` has been implemented in the largest python brain signal pro-

cessing package, `MNE`, and is now the default solver for sparse signal estimators. Algorithms for bilevel optimization with nonsmooth inner optimization problems should soon be implemented in the library `jaxopt`, which extends the automatic differentiation library `jax`, and allows to differentiate solutions of optimization problems.

Notation

General

$\triangleq$	Equal by definition	
$[p]$	Set of integers from 1 to p included	
Id_n	Identity matrix in $\mathbb{R}^{n\times n}$	
$A_{i:}$	i^{th} row of matrix A	
$A_{:j}$	j^{th} column of matrix A	
$\mathrm{Tr}\, A$	Trace of $A \in \mathbb{R}^{d\times d}$	$\mathrm{Tr}\, A = \sum_{i=1}^{d} A_{ii}$
$A^\top$	Transpose of matrix A	
$\mathbb{S}_+^p$	Set of p by p symmetric positive semidefinite matrices	
$\mathbb{S}_{++}^p$	Set of p by p symmetric positive definite matrices	
$\|\cdot\|$	Euclidean norm on vectors and matrices	
$\langle\cdot,\cdot\rangle$	Euclidean inner product	
$\|\cdot\|_2$	Spectral norm on matrices	
$\|\cdot\|_\gamma$	Norm induced by the vector $\gamma \in \mathbb{R}^p$	$\|\beta\|_\gamma = \sqrt{\sum_{j=1}^{p} \gamma_j \beta_j^2}$
$\|\cdot\|_A$	Norm induced by the matrix $A \in \mathcal{S}_{++}^n$	$\|\beta\|_A = \sqrt{\beta^\top A \beta}$
$\|\cdot\|_{\mathscr{S},p}$	Schatten p-norm on matrices for $p \in [1, +\infty]$	
$\|\cdot\|_*$	Nuclear norm	
$\|\cdot\|_{2,1}$	Row-wise $\ell_{2,1}$-mixed norm on matrices	$\|A\|_{2,1} = \sum_{j=1}^{p} \|A_{j:}\|$
$\|\cdot\|_{1,2}$	Column-wise $\ell_{1,2}$-mixed norm on matrices	$\|A\|_{1,2} = \sum_{i=1}^{n} \|A_{:i}\|$
$\|\cdot\|_{2,\infty}$	Row-wise $\ell_{2,\infty}$-mixed norm on matrices	$\|A\|_{2,\infty} = \max_{j\in[p]} \|A_{j:}\|$
$u \odot v$	Coordinatewise multiplication of vectors u and v	
$u \odot A$	The row wise multiplication between a vector u and a matrix A	
$\rho(A)$	Spectral radius of the matrix A	
$\kappa(A)$	Condition number of the matrix A	
$(e_j)_{j=1}^p$	Canonical base of $\mathbb{R}^p$	
$\mathrm{supp}(\beta)$	Support of $\beta \in \mathbb{R}^p$	$\left\{j \in [p] \,:\, \beta_j \neq 0\right\}$
S^c	Complement of the set S	
$\mathcal{B}(x, \epsilon)$	Ball of center x and radius ϵ	
$\mathcal{J}\psi(x)$	Jacobian of the function ψ at x	

Convex analysis

$\text{prox}_g(x)$	Proximity operator of g at x	$\arg\min_{y\in\mathbb{R}^p} \frac{1}{2}\|x-y\|^2 + g(y)$
$\text{aff}(\mathcal{C})$	Affine hull of the convex set $\mathcal{C}$	
$\text{ri}(\mathcal{C})$	Relative interior of the convex set $\mathcal{C}$	
$\iota_{\mathcal{C}}(x)$	Indicator function of the set $\mathcal{C}$ at x	$\iota_{\mathcal{C}}(x) = \begin{cases} 0 \text{ if } x \in \mathcal{C} \\ +\infty \text{ otherwise} \end{cases}$.
$\text{dom}(f)$	Domain of the function f	$\{x \in \mathbb{R}^p : f(x) < +\infty\}$
$\partial f(x)$	Subdifferential of the function f at x	$\partial f(x) = \{s \in \mathbb{R}^p : f(y) \geq f(x) + \langle s, y-x\rangle, \forall y \in \text{dom}(f)\}$
$f^*(u)$	Fenchel conjugate of the function f at u	$\sup_x \langle u, x\rangle - f(x)$
$f_1 \,\square\, f_2(x)$	Infimal convolution of f_1 and f_2 at x	$\inf\{f_1(x-y) + f_2(y) : y \in \mathbb{R}^d\}$
L	Lipschitz contant of ∇f	
L_j	Lipschitz contant of the function $\nabla_j f$	$\|\nabla_j f(x + he_j) - \nabla_j f(x)\| \leq L_j\|h_j\|$, $\forall x \in \mathbb{R}^p$, $h \in \mathbb{R}$

For S_1 and $S_2 \in \mathcal{S}_+^n$, $S_1 \succeq S_2$ if $S_1 - S_2 \in \mathcal{S}_+^n$. When we write $S_1 \succeq S_2$ we implicitly assume that both matrices belong to $\mathcal{S}_+^n$.

A function is said to be *smooth* if it has Lipschitz gradients. Let f be a L-smooth function. Lipschitz constants of the functions $\nabla_j f$ are denoted by L_j; hence for all $x \in \mathbb{R}^p$, $h \in \mathbb{R}$:

$$|\nabla_j f(x + he_j) - \nabla_j f(x)| \leq L_j|h| \ .$$

A function $h : \mathbb{R} \to \mathbb{R} \cup \{+\infty\}$ is said to be proper if $\text{dom}(h) = \{x \in \mathbb{R} : h(x) < +\infty\} \neq \emptyset$), and closed if for any $\alpha \in \mathbb{R}$, the sublevel set $\{x \in \text{dom}(h) : h(x) \leq \alpha\}$ is a closed set.

1 Motivation and contributions

Contents

1.1 Model selection for the M/EEG inverse problem

Over the last century, neuroscience has led to significant advances in brain understanding. Breakthroughs include functional localization (Penfield and Rasmussen, 1950), better comprehension of diseases such as epilepsy (Penfield and Jasper, 1954), or precise description of the visual cortex (Hubel and Wiesel, 1962). Much remains to be understood and tackling modern brain diseases appears as a major challenge. To this aim, neuroscientific studies extensively rely on advanced recording and processing of brain activity.

1.1.1 Neuro-imaging

What is brain activity? The nervous system is composed of a complex network of billions of neurons. Each neuron is composed of a cell body (the soma), dendrites, and an axon. The variation of the ionic concentration at the soma's membrane produces an electrical potential. When this potential rapidly rises and falls, an *action potential* can be triggered, locally propagating potential variations along the axons to ten to thousands neighboring neurons. This leads to *excitatory postsynaptic potentials* in the dendrites (Baillet et al., 2001). The goal of neuro-imaging is to record brain activity, for example through the measure of these simultaneous and localized neural activations.

How to measure the brain activity? Neuro-imaging, or brain imaging, is defined as the direct or indirect imaging of the nervous system structure. Main neuro-imaging techniques include electrocorticography, electro- and magnetoencephalography and functional magnetic resonance imaging:

(a) ECoG. (b) EEG. (c) MEG.

Figure 1.1 – **Multiple neuro-imaging modalities.** In electrocorticography (left, Blausen 2014), a grid of sensors is placed onto the brain surface. For electroencephalography (middle[1]) and magnetoencephalography (right, image from the National Institute of Mental Health), sensors are placed at or close to the head surface and no surgery is needed.

- Electrocorticography (**ECoG**, Jasper and Penfield 1949) measures the electrical field produced by the active neurons, with electrodes directly placed in the head, on cortical surfaces (Figure 1.1a). ECoG leads to signals with very good spatial and temporal resolutions, approximately 1 mm and 1 ms, but is extremely invasive: a part of the skull must be removed in order to place electrodes on cortical surfaces.

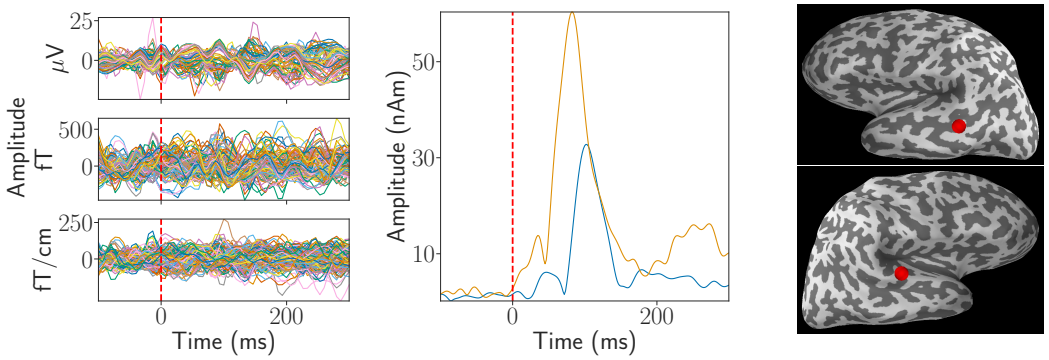
- Electro- and magnetoencephalography (**EEG** Berger 1929, and **MEG** Cohen 1968) measure the electric and magnetic fields at the head surface (Figures 1.1b and 1.1c). After an excitatory postsynaptic potential is triggered, all synchronized neurons produce electric and magnetic fields, with sufficiently large amplitudes to be measured by sensors at the head surface. Measurements are very noisy (Figure 1.2a) but have an excellent time resolution, around 1 ms. Spatial resolution of M/EEG is unclear and seems to be more related to the reconstruction problems difficulty than the techniques themselves. As opposed to ECoG, EEG and MEG are non invasive techniques: no brain surgery is required.

- Functional magnetic resonance imaging (**fMRI**) measures changes in brain blood flow: the latter is coupled to neuronal activity (Logothetis et al., 2001) and increases in active brain areas. This blood flow, called *haemodynamic response*, leads to higher concentrations in oxygenated hemoglobin, which can be detected using blood oxygen-level-dependent (BOLD) signals (Ogawa et al., 1990). As opposed to EEG and MEG measurements, fMRI is an indirect technique that measures blood flow as a proxy for brain activity. fMRI BOLD signals typically have a high spatial resolution, from 1 to 3 mm, but a poor temporal resolution, from 1 to 3 s.

With their outstanding temporal resolution, MEG and EEG are tools of choice to observe and understand brain activity. In the next paragraphs we detail M/EEG data specificities and present a mathematical modelling for neuro-imaging with M/EEG.

Sensors. M/EEG recordings typically involve around 300 sensors, each of which records the amplitude of the electric or magnetic field at a different location at the head surface or periphery. There are three types of sensors (see Figure 1.2a):

[1]Image is in the public domain: https://commons.wikimedia.org/wiki/File:EEG_cap.jpg.



(a) M/EEG sensors Y. (b) Brain sources B^*. (c) Brain sources location.

Figure 1.2 – **M/EEG data and brain signals we aim at reconstructing.** Amplitude of the electric field, magnetic field, and gradient of the magnetic field (left) at the head surface or periphery. Amplitude of the current in two areas (middle), and their location (right). Data comes from the `MNE` software (Gramfort et al., 2014).

- Electrodes, recording the amplitude of the electric field (in μV).
- Magnetometers, recording the amplitude of the magnetic field (in fT), which is eight orders of magnitude smaller than the Earth's magnetic field.
- Gradiometers, recording the amplitude of the gradient of the magnetic field (in fT/cm).

Figure 1.2a shows M/EEG measurements after auditory stimulation at time 0 (dashed red line). Each sensor provides a time series of 400 to 500 ms with a sampling rate of 600 Hz (leading to around 50 time points after resampling): M/EEG data recordings consist in a matrix Y of size $\underbrace{\text{number of sensors}}_{n \approx 300} \times \underbrace{\text{number of time points}}_{T \approx 50}$.

Sources. Simultaneous postsynaptic potentials can be modelled as an *equivalent dipole* at the macroscopic scale (Williamson et al., 2013). Source candidates are chosen on a discrete mesh of the cortical surface (Nunez and Silberstein, 2000). The goal is then to reconstruct the electric current intensity amplitude (Figure 1.2b) of the equivalent dipole in each point on the mesh (Figure 1.2c). The equivalent dipole direction is assumed to be given, and normal to the cortical surface (Lin et al., 2006a). To summarize, our goal is to recover a matrix B^* of size $\underbrace{\text{number of source candidates}}_{p \approx 10^4} \times \underbrace{\text{number of time points}}_{T \approx 50}$, where each row is a time series with T time points, corresponding to the amplitude of the electric current on the cortical surfaces along the cognitive experiment (Baillet et al., 2001).

Forward model. Physics of the problem leads to a direct relation between the neural activity we aim at reconstructing (Figures 1.2b and 1.2c) and the recorded M/EEG signals (Figure 1.2a). The biomagnetic *forward* model is well-posed (Plonsey and Heppner, 1967; Hämäläinen et al., 1993): if one knows the number of active sources, their amplitudes and localizations in the brain, one can determine the amplitudes of the observed M/EEG signals. We use the quasi-static approximation (Tripp, 1983; Heller and van Hulsteyn, 1992; Hämäläinen et al., 1993): propagation of the electric and magnetic fields is supposed to be instantaneous (no delayed potentials). This leads to a linear relation

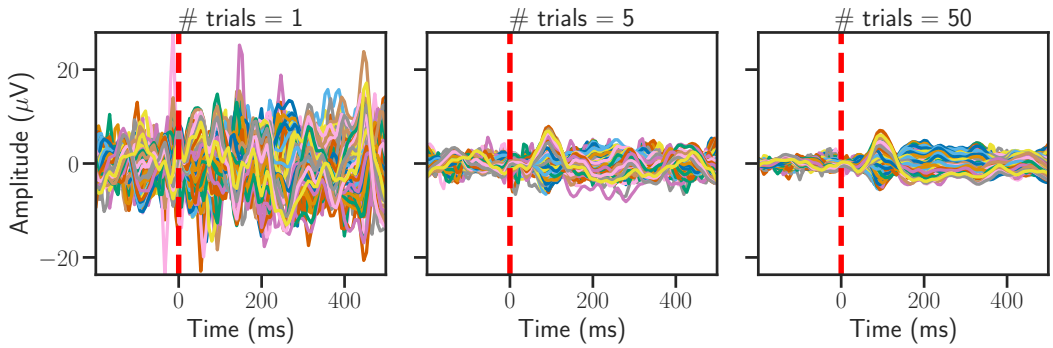


Figure 1.3 – **Influence of the number of trials.** Amplitude of the electric field measured at the surface of the head, as a function of time, for multiple numbers of averaged trials $Y^{(l)}$. With only one trial (left), one cannot see a brain response, after the auditory stimulation at time 0 (dashed red line). When the number of trials increases (middle and right), one can observe a brain response after the stimulation.

(Hämäläinen and Ilmoniemi, 1994) between the observed signals amplitudes $Y \in \mathbb{R}^{n \times p}$ and the electrical current amplitudes in the brain $\mathrm{B}^* \in \mathbb{R}^{p \times T}$, through a *design matrix* $X \in \mathbb{R}^{n \times p}$ (also called *forward operator*). In addition, signals are corrupted with additive noise $\mathrm{E} \in \mathbb{R}^{n \times T}$:

$$Y = X\mathrm{B}^* + \mathrm{E} \ . \tag{1.1}$$

This linear model is frequent in machine learning and inverse problems. In the next paragraph, we present some aspects of it specific to M/EEG data.

Signal characteristics. First, M/EEG recordings are very noisy: noise and signal amplitudes are of the same order of magnitude (Figure 1.3, left). In order to decrease the noise level, a common procedure is to repeat the same cognitive experiment multiple times. The brain response B^* is assumed to be the same for all trials. M/EEG data is thus composed of $r \in \mathbb{N}$ measurements $Y^{(l)} \in \mathbb{R}^{n \times T}$:

$$Y^{(l)} = X\mathrm{B}^* + \mathrm{E}^{(l)} \ , \tag{1.2}$$

with $\mathrm{E}^{(l)} \in \mathbb{R}^{n \times p}$, $l \in [r]$. Data is then averaged across the repetitions $Y^{(l)}$, called *trials* in M/EEG, to obtain a better signal-to-noise ratio (Figure 1.3). Another specificity of M/EEG data is that the additive noise is usually not Gaussian i.i.d., but correlated. In Chapter 5 we will take advantage of the repetitions structure to better handle the correlated Gaussian noise.

Equipped with this mathematical modelling of M/EEG data and brain sources, we now recall some usual techniques to reconstruct neural activity from M/EEG recordings.

1.1.2 Linear inverse problem

With M/EEG data we have access to the amplitudes of the electric and magnetic fields *at the surface* of the head, yet we seek to recover the intensity of the current *inside* the brain. There exists three main types of approaches to identify sources responsible for observed M/EEG signals: parametric methods, scanning methods, and imaging methods (Table 1.1).

Method	Parametric	Scanning Beamforming	Scanning MUSIC	Imaging
Predefined number of sources	yes	no	yes	no
Predefined grid of locations	no	yes	yes	yes

Table 1.1 – Imaging techniques requirements.

Parametric methods (*dipole fitting*, Scherg and Cramon 1985; Scherg 1990) assume that observed signals are produced by a small and fixed number of active sources. Then source locations and orientations are determined minimizing a criterion. This approach leads to challenging non-convex optimization problems, which can be solved using clustering methods, simulated annealing, or genetic algorithms (Uutela et al., 1998). In addition to complex algorithms, no clear stopping criteria, and sensitive initialization, the number of active sources can also be hard to estimate in advance.

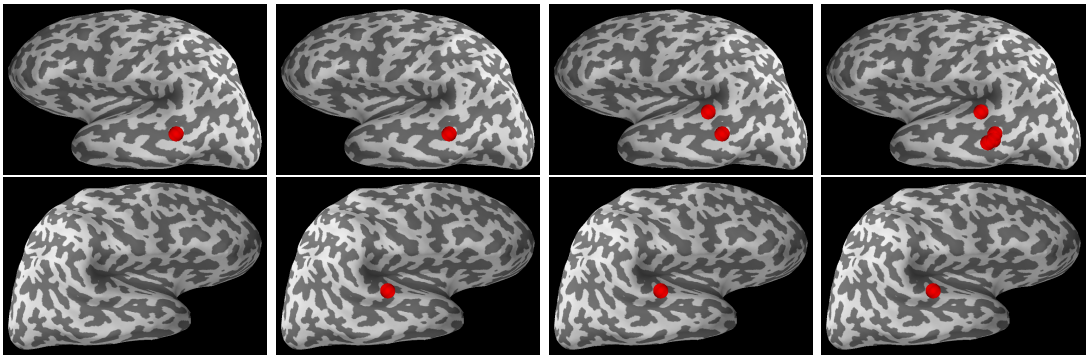
Scanning methods (*beamforming* Mosher et al. 1999 and MUSIC Mosher and Leahy 1998) evaluate the contribution of each source on a discrete grid, *scanning* the source space to determine optimal active source locations (Hillebrand et al., 2005; Sekihara et al., 2002). No interference is assumed between the sources, and their contributions are computed independently using correlations between each brain location and the observed signals. More refined methods combine beamforming with signal subspace estimation, leading to MUSIC approaches (Mosher and Leahy, 1999).

Imaging methods rely on a fixed predefined grid of sources. As opposed to scanning methods, they jointly estimate amplitudes of the currents in all potential brain source locations. This leads to a severely ill-posed inverse problem: the number of sensors (the number of samples from a statistical point of view, around 10^2) is orders of magnitude lower than the number of potential source locations (around 10^4). This leads to non-unique solutions, highly sensitive to noise corruption: to circumvent these problems, one can incorporate some prior knowledge on the desired sources to recover. This prior can be enforced through constrained optimization, regularized optimization, or using Bayesian statistics (Gelman et al., 2013). Most popular priors in M/EEG include ℓ_2 regularization (called MNE in the neuro-imaging community Hämäläinen and Ilmoniemi 1994), weighted ℓ_2 regularization (Lin et al., 2006b), ℓ_1 regularization (Tibshirani, 1996; Ou et al., 2009; Gramfort et al., 2012), and iterative reweighting techniques (Candès et al., 2008; Strohmeier et al., 2014).

In the rest of this thesis we focus on imaging methods with sparsity inducing priors, *e.g.,* ℓ_1 or $\ell_{2,1}$ penalties, and more generally with nonsmooth regularization. A wealth of such imaging techniques has been proposed in the literature, but unfortunately only very few are actually used in practice: they often lack of efficient and automatic ways to tune their hyperparameters. A notable example in M/EEG is the multitask Lasso (Argyriou et al., 2008; Obozinski et al., 2011):

$$\hat{\mathrm{B}}^{(\lambda)} \in \underset{\mathrm{B}\in\mathbb{R}^{p\times T}}{\arg\min} \frac{1}{2}\|Y - X\mathrm{B}\|^2 + \lambda \underbrace{\sum_{j=1}^{p} \|\mathrm{B}_{j:}\|}_{\triangleq \|\mathrm{B}\|_{2,1}} \quad . \tag{1.3}$$

Modern block coordinate descent algorithms (Tseng and Yun, 2009a; Wright, 2015; Massias et al., 2020b) efficiently solve Problem (1.3) and selection of the regularization hyperparameter now appears as the main obstacle for these estimators to be massively



(a) $\lambda = 0.85\lambda_{\max}$ (b) $\lambda = 0.82\lambda_{\max}$ (c) $\lambda = 0.80\lambda_{\max}$ (d) $\lambda = 0.75\lambda_{\max}$

Figure 1.4 – **Hyperparameter selection for real M/EEG data.** Brain source locations (in red) found in the left (top) and in the right (bottom) hemispheres, using multitask Lasso Problem (1.3), for multiple values of λ. Each column corresponds to one value of the hyperparameter λ. Which λ to pick? How to *automatically* select λ?

used by neuroscientists. Figure 1.4 represents estimated active brain locations found when solving Problem (1.3) for multiple values of the regularization parameter λ. Brain source locations are taken as the non zeros rows of the estimator $\hat{B}^{(\lambda)}$ defined in Problem (1.3), where λ is chosen as a fraction of $\lambda_{\max} \triangleq \|X^\top Y\|_{2,\infty}$ (the largest value of the parameter producing non-trivial solutions). Small variations in λ can lead to different numbers of active sources and brain locations. Ignoring the temporal aspect, in Chapter 6 we focus on the setting where the regularization parameter λ trades data fidelity against prior:

$$\hat{\beta}^{(\lambda)} \in \underset{\beta \in \mathbb{R}^p}{\arg\min}\, \Phi(\beta, \lambda) \triangleq f(\beta) + \lambda g(\beta) \ , \tag{1.4}$$

where f is a convex smooth function (the data fit), and g a convex, proper, lower semicontinuous and usually nonsmooth function (the regularizer).

Figure 1.4 highlights the importance of model selection for sparse linear inverse problems. Hence, in the next section we review some usual procedures to select the regularization hyperparameter λ for estimators based on Problem (1.4).

1.1.3 Hyperparameter selection for sparse models

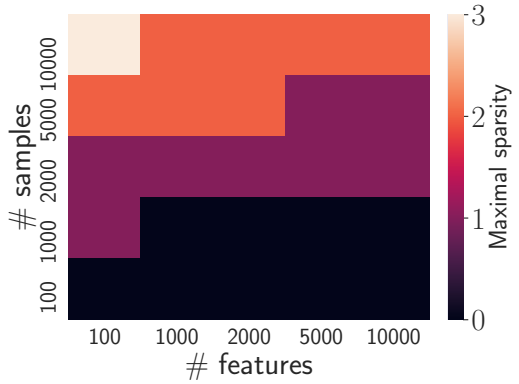
The hyperparameter selection problem can be tackled in various frameworks. Most of them can be fitted in three categories: *the statistical route*, *Bayesian statistics*, and *hyperparameter optimization*. In this section we assume to have access to a target vector $y \in \mathbb{R}^n$, and a design matrix $X \in \mathbb{R}^{n \times p}$.

The statistical route. The Lasso (Tibshirani, 1996):

$$\hat{\beta}^{(\lambda)} \in \underset{\beta \in \mathbb{R}^p}{\arg\min}\, \frac{1}{2}\|y - X\beta\|^2 + \lambda\|\beta\|_1 \ , \tag{1.5}$$

and its variations (Zou and Hastie, 2005; Koh et al., 2007; Obozinski et al., 2011; Simon et al., 2013), led to a broad literature for hyperparameter tuning (Lounici, 2008; Lounici et al., 2009; Bickel et al., 2009; Belloni et al., 2011). Under the exact sparse linear model assumption (Lounici, 2008, Ass. 1)

$$y = X\beta^* + \varepsilon \ , \text{ with } \varepsilon \sim \mathcal{N}(0, \sigma^{*2}\,\mathrm{Id}_n), \ \ \beta^* \in \mathbb{R}^p, \ \ \|\beta^*\|_0 \leq s, \text{ for some } s \in \mathbb{N} \ , \tag{1.6}$$



(a) Synthetic data.

Dataset name	Maximal sparsity
leukemia	0
breast cancer	0
colon cancer	0
madelon	0
colon cancer	0
covtype	0

(b) Real data.

Figure 1.5 – Maximal sparsity that can theoretically be recovered assuming mutual incoherence in Equation (1.7) is displayed for multiple values of number of samples and number of features (left), and for multiple real-life datasets (right). For the synthetic data, entries of the design matrix X are drawn i.i.d. following a Gaussian distribution of mean 0 and variance 1. Real datasets are taken from `libsvm` (Chang and Lin, 2011)[2].

and under the mutual incoherence hypothesis on the design matrix X (Lounici, 2008, Ass. 2)

$$\|X_{:j}\| = 1 \ , \text{ and } \max_{j' \neq j} |X_{:j'}^\top X_{:j}| \leq \tfrac{1}{7\alpha s}, \forall j \in [p], \text{ for some constant } \alpha > 1 \ , \tag{1.7}$$

one can show that if the regularization parameter λ is chosen as

$$\lambda = A\sqrt{\frac{\log p}{n}}\sigma^* \ , \text{ with } A \geq 2\sqrt{2} \ , \tag{1.8}$$

then the regression coefficients are well estimated, for some $C > 0$, with high probability

$$\|\hat{\beta} - \beta^*\|_\infty \leq C\sqrt{\frac{\log p}{n}}\sigma^* \ . \tag{1.9}$$

Under the additional hypothesis that the amplitude of the true regression coefficients β^* is "large enough", it is possible to show that their sign is recovered with high probability (Lounici, 2008, Thm. 2). The presented properties can be generalized to other estimators, with multiple data fitting terms and penalties (van de Geer, 2016, Chap. 6 and 7). However, the statistical route has multiple weaknesses:

(i) It requires the knowledge of the true noise level σ^*. This can be fixed using *pivotal* estimators (Belloni et al., 2011) as we will detail in Chapter 4. Note also than one can estimate the noise level σ^* in M/EEG recordings (Ledoit and Wolf, 2004; Engemann and Gramfort, 2015; Cai et al., 2021).

(ii) Although the mutual incoherence hypothesis in Equation (1.7) is not a necessary condition for support recovery, it is very strong and generally does not hold on real data (Figure 1.5).

[2] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/.

To summarize, the statistical route relies on restrictive hypotheses and quantities which are typically unknown in practice: Lasso users still have to resort to other techniques to select the hyperparameter λ.

Other techniques involving Bayesian statistics (Kaipio and Somersalo, 2006; Bolstad and Curran, 2016) do not rely on the mutual incoherence hypothesis and provide multiple ways to set the regularization parameter λ in the hierarchical Bayesian framework (Molina et al., 1999; Sato et al., 2004).

Hierarchical Bayesian modeling. In this approach the regularization parameter is incorporated to the model: a distribution $p(\lambda)$ has to be chosen on the hyperparameter λ. The joint probability distribution writes:

$$p(\beta, \lambda|y) \sim p(y|\beta, \lambda)p(\beta|\lambda)p(\lambda) \tag{1.10}$$

$$\sim \exp\Big(-f(\beta) - \lambda g(\beta) + \log p(\lambda)\Big) / C(\lambda) \ . \tag{1.11}$$

One can resort to multiple inference techniques to estimate the regression coefficients $\hat{\beta}$ and the regularization parameter $\hat{\lambda}$ (Table 1.2):

- One can jointly compute $\hat{\beta}$ and $\hat{\lambda}$ using a *full maximum-a-posteriori* approach (full-MAP, Calvetti and Somersalo 2008; Calvetti et al. 2009; Lucka 2012):

  $$(\hat{\beta}, \hat{\lambda}) \in \arg\max_{\beta,\lambda} p(\beta, \lambda|y) \ . \tag{1.12}$$

  Note that the constant $C(\lambda) = \int \exp(-\lambda g(\beta))\mathrm{d}\beta$ (which depends on λ) of the probability distribution in Equation (1.11) is unknown and can be hard to estimate. For some specific models, a closed-form formula can be derived (Pereyra et al., 2015), otherwise it is ignored (Bekhti et al., 2018), leading to alternate sampling schemes, where one successively samples from β and λ.

- One can marginalize with respect to β (Tipping, 2001; Wipf and Rao, 2004; Wipf and Nagarajan, 2009; Zhang and Rao, 2011):

  $$\hat{\lambda} \in \arg\max_{\lambda} p(\lambda|y) \tag{1.13}$$

  $$\hat{\lambda} \in \arg\max_{\lambda} \int p(\beta, \lambda|y)\mathrm{d}\beta \ , \tag{1.14}$$

  and then infer the regression coefficients $\hat{\beta}^{(\lambda)}$ with usual optimization tools. This inference is also referred to as λ *maximum-a-posteriori* (λ-MAP), *type-II likelihood* or *empirical Bayes* because the prior $p(\beta|\lambda)$ is learned from the data. Integrals in Equation (1.14) can be intractable: one can resort to *expectation-minimization* (Dempster et al., 1977), or *variational inference* (Jordan et al., 1999; MacKay, 2003; Friston et al., 2008) to compute them.

- One can marginalize with respect to λ (Figueiredo, 2001; Cotter et al., 2005; Seeger and Wipf, 2010; Pereyra et al., 2015):

  $$\hat{\beta} \in \arg\max_{\beta} p(\beta|y) \tag{1.15}$$

  $$\hat{\beta} \in \arg\max_{\beta} \int p(\beta, \lambda|y)\mathrm{d}\lambda \ . \tag{1.16}$$

  This approach is sometimes referred to as β *maximum-a-posteriori* (β-MAP, Wipf and Nagarajan 2009, Sec. 4).

Full-MAP	λ-MAP	β-MAP	MLE
$\hat{\beta}, \hat{\lambda} \in \underset{\beta,\lambda}{\arg\max}\, p(\beta, \lambda \vert y)$	$\hat{\lambda} \in \underset{\lambda}{\arg\max}\, p(\lambda \vert y)$	$\hat{\beta} \in \underset{\beta}{\arg\max}\, p(\beta \vert y)$	$\hat{\lambda} \in \underset{\lambda}{\arg\max}\, p(y \vert \lambda)$

Table 1.2 – Bayesian hyperparameter selection inferences.

- Previously presented Bayesian approaches require to specify a probability distribution on λ. Recent advances in sampling theory made it possible to remove this modeling: no probability distribution is specified on λ, it is learned from the data. The *maximum likelihood estimator* (MLE) writes:

$$\hat{\lambda} = \underset{\lambda}{\arg\max}\, p(y|\lambda) \tag{1.17}$$

$$= \underset{\lambda}{\arg\max} \int p(y|\beta)p(\beta|\lambda)\mathrm{d}\beta \ . \tag{1.18}$$

  Relying on state-of-the-art proximal Monte-Carlo Markov chain methods (Durmus and Moulines, 2016; Durmus et al., 2018; Durmus and Moulines, 2019) Vidal et al. (2020); Bortoli et al. (2020) made it possible to sample efficiently from $p(y|\lambda)$, through an alternated sampling scheme. Once λ has been computed, one can access $\hat{\beta}^{(\lambda)}$ by solving Problem (1.4) with usual convex optimization tools.

To conclude, Bayesian approaches are interpretable, flexible, and allow to incorporate prior knowledge in the regularization parameter λ with hierarchical approaches, or to directly use data to select the regularization parameter. The main disadvantages of these techniques are practical:

(i) Usual hyperpriors on λ are parametric and introduce new hyperparameters that have to be selected.

(ii) They lead to non-convex problems for which convergence toward a global minimum might depend on the initialization.

(iii) In addition to the lack of clear stopping criteria, convergence speed of samplers can be dramatically affected by other hyperparameters, such as stepsizes, or initializations (see for instance Vidal et al. 2020, Fig. 3.b).

Hyperparameter optimization. In machine learning the most popular approach for hyperparameter selection is *hyperparameter optimization* (Kohavi and John, 1995; Hutter et al., 2015; Feurer and Hutter, 2019): one selects the hyperparameter λ such that the regression coefficients $\hat{\beta}^{(\lambda)}$ minimize a given criterion $\mathcal{C} : \mathbb{R}^p \to \mathbb{R}$. Here $\mathcal{C}$ should ensure good generalization, or avoid overcomplex models. Common examples include the hold-out loss (Devroye and Wagner, 1979), the cross-validation loss (CV, Stone and Ramer 1965), the AIC (Akaike, 1974), BIC (Schwarz, 1978) or SURE (Stein, 1981) criteria. Formally, the hyperparameter optimization problem is a bilevel optimization problem (Bracken and McGill, 1973; Candler and Norton, 1977; Colson et al., 2007):

$$\begin{aligned} &\hat{\lambda} \in \underset{\lambda \in \mathbb{R}^r}{\arg\min} \left\{ \mathcal{L}(\lambda) \triangleq \mathcal{C}\left(\hat{\beta}^{(\lambda)}\right) \right\} \\ &\text{s.t. } \hat{\beta}^{(\lambda)} \in \underset{\beta \in \mathbb{R}^p}{\arg\min}\, \Phi(\beta, \lambda) \ . \end{aligned} \tag{1.19}$$

The most famous criterion may be the hold-out loss (Devroye and Wagner, 1979). Data (X, y) is split in two sets: the training set $(X^{\text{train}}, y^{\text{train}})$, on which the model in trained, and the validation set $(X^{\text{val}}, y^{\text{val}})$, on which the prediction quality of $\hat{\beta}^{(\lambda)}$ is evaluated. If $(X_{i:}, y_i)$ are i.i.d., $(X^{\text{val}}, y^{\text{val}})$ plays the role of "unseen" new data. For the Lasso it writes

$$\begin{aligned} &\hat{\lambda} \in \arg\min_{\lambda \in \mathbb{R}} \frac{1}{2} \|y^{\text{val}} - X^{\text{val}} \hat{\beta}^{(\lambda)}\|^2 \\ &\text{s.t. } \hat{\beta}^{(\lambda)} \in \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2} \|y^{\text{train}} - X^{\text{train}} \beta\|^2 + \lambda \|\beta\|_1 \ . \end{aligned} \tag{1.20}$$

Hold-out relies on one split of the data, which leads to one estimate of the risk. Cross-validation principle is to use multiple splits to create multiple estimators of the risk, and then average it. Multiple cross-validation procedures exist (Arlot and Celisse, 2010, Sec. 4.3), most popular include leave-one-out (Stone, 1974; Allen, 1974), leave-p-out (Shao, 1993; Zhang, 1993) or K-fold cross-validation (Geisser, 1974) which is the most common in machine learning. Data (X, y) is partitioned into $K \in \mathbb{N}^*$ hold-out datasets $(X^{\text{train}_k}, y^{\text{train}_k})$, $k \in [K]$. The regularization parameter λ is then chosen to minimize the averaged squared norm of the errors

$$\begin{aligned} &\hat{\lambda} \in \arg\min_{\lambda \in \mathbb{R}} \frac{1}{K} \sum_{k=1}^{K} \|y^{\text{val}_k} - X^{\text{val}_k} \hat{\beta}^{(\lambda,k)}\|_2^2 \\ &\text{s.t. } \hat{\beta}^{(\lambda,k)} \in \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2} \|y^{\text{train}_k} - X^{\text{train}_k} \beta\|_2^2 + \lambda \|\beta\|_1, \quad \forall k \in [K] \ . \end{aligned} \tag{1.21}$$

Cross-validation is well-suited when the "samples" $(X_{i:}, y_i)$ are assumed to be i.i.d., which is often not the case for inverse problems, and definitely not the case for the M/EEG inverse problem. In addition, if the goal is *model selection*, *i.e.,* recovering the exact support of the true coefficients β^*, then some cross-validation procedures were shown to be inconsistent: they select all active variables in β^*, but also select some additional irrelevant variables (Shao, 1993, 1997).

This is why other criteria, such as the Stein unbiased risk estimator (SURE, Stein 1981), are more popular in the inverse problem literature (Galatsanos, 1992; Donoho and Johnstone, 1995; Zhang and Desai, 1998; Blu and Luisier, 2007; Pesquet et al., 2009; Vaiter et al., 2013). It is an unbiased estimator of the mean squared error, which tends to penalize overcomplex models through the *degree of freedom* (dof) (Efron, 1986; Deledalle et al., 2014, Sec. 2.1). The SURE criterion writes

$$\text{SURE}(\beta) = \|y - X\beta\|^2 - \sigma^{*2} \operatorname{Tr}(X^\top X) + 2\text{dof}(\beta) \ . \tag{1.22}$$

Unfortunately the degree of freedom is not always available in closed-form or differentiable in β. To circumvent these problems, multiple SURE proxies have been developed: finite-difference SURE (Ye, 1998; Shen and Ye, 2002), Monte-Carlo SURE (Ramani et al., 2008), iterative differentiation Monte-Carlo SURE (Vonesch et al., 2008), or finite-differentiation Monte-Carlo SURE (Deledalle et al., 2014). For the Lasso, the

latter approximation yields the following problem:

$$\begin{aligned} \hat{\lambda} &\in \arg\min_{\lambda \in \mathbb{R}} \frac{1}{2}\|y - X\hat{\beta}^{(\lambda,1)}\|^2 + \frac{\sigma^{*2}}{\epsilon}\left\langle \hat{\beta}^{(\lambda,2)} - \hat{\beta}^{(\lambda,1)}, X^\top \delta \right\rangle \\ \text{s.t. } \hat{\beta}^{(\lambda,1)} &\in \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2}\|y - X\beta\|^2 + \lambda\|\beta\|_1 \\ \hat{\beta}^{(\lambda,2)} &\in \arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2}\|y + \epsilon\delta - X\beta\|^2 + \lambda\|\beta\|_1 \ , \end{aligned} \tag{1.23}$$

with $\epsilon > 0$ and $\delta \sim \mathcal{N}(0, \mathrm{Id}_n)$. As the statistical route, the SURE requires the knowledge of the true noise level σ^*. Finite-differentiation Monte-Carlo SURE also introduces a new hyperparameter ϵ, for which theoretical guidelines exist (Deledalle et al., 2014, Sec. 5.1).

Problems (1.20), (1.21) and (1.23) are bilevel optimization problems with nonsmooth inner optimization problems. These problems can be challenging to solve, especially as the dimension of the hyperparameter becomes larger. So far we have considered models with one hyperparameter λ, but more refined models rely on multiple hyperparameters, for instance the sparse-group Lasso (Simon et al., 2013), elastic-net (Zou and Hastie, 2005) or the weighted Lasso (Lasso with one regularization parameter per feature, Zou 2006). In Chapter 6 we propose efficient first-order techniques to solve Problem (1.19) with a potential large number of hyperparameters.

One of the main goal of this thesis is to provide a fast way to select the regularization parameter λ. Most hyperparameter selection techniques rely on efficient resolutions of Problem (1.4), solved with coordinate descent: we also use it intensively (Chapters 2, 3, 5 and 6). The next section proposes an introduction to coordinate descent.

1.2 Coordinate descent

Coordinate descent is a variant of gradient descent, which updates the iterates one coordinate at a time (Tseng and Yun, 2009b). Coordinate descent is mostly used in practice in its proximal variation, on composite "smooth + separable nonsmooth" optimization problems:

$$\min_{\beta \in \mathbb{R}^p} f(\beta) + \sum_{j=1}^{p} g_j(\beta_j) \ , \tag{1.24}$$

where f is a smooth convex function, and the functions g_j are convex, proper, lower semicontinuous and usually nonsmooth. Proximal coordinate descent has been applied to numerous machine learning problems (Shalev-Shwartz and Zhang, 2013a), in particular the Lasso (Tibshirani, 1996), the elastic net (Zou and Hastie, 2005) or the sparse logistic regression (Ng, 2004). It is currently the state-of-art default algorithm in preeminent machine learning packages such as `scikit-learn` (Pedregosa et al., 2011), `glmnet` (Friedman et al., 2009), `libsvm` (Fan et al., 2008) or `lightning` (Blondel and Pedregosa, 2016). Coordinate descent can also be applied on group-type penalties (Tseng and Yun, 2009a; Obozinski et al., 2011; Simon et al., 2013), and leads to good empirical performance with nonsmooth nonconvex penalties g_j (Breheny and Huang, 2011; Mazumder et al., 2011; Ge et al., 2019).

For exposition purposes we present the algorithms on an ordinary least squares problem (OLS), with a design matrix $X \in \mathbb{R}^{n \times p}$ and a target vector $y \in \mathbb{R}^n$:

$$\min_{\beta \in \mathbb{R}^p} f(\beta) \triangleq \frac{1}{2}\|y - X\beta\|^2 \ . \tag{1.25}$$

We suppose that $X^\top X \succ 0$, we define μ as the smallest eigenvalue of $X^\top X$, and L its largest, $L \triangleq \|X\|_2^2$. In Section 1.2.1 we give some intuitions on why coordinate descent achieves good performance, and explain implementation details. To give insights, we compare coordinate descent against gradient descent, and instantiate all algorithms on the simpler Problem (1.25). In Section 1.2.2 we recall some theoretical results on coordinate descent on quadratics. In particular we provide a proof of the linear convergence of cyclic coordinate descent for Problem (1.25), which is a simpler case of the proof of Theorem 2.16 from Chapter 2. We also remind under which conditions coordinate descent outperforms gradient descent. Section 1.2.3 presents the generalization of coordinate descent for composite optimization Problem (1.24). Readers can refer to Wright (2015); Shi et al. (2016) for comprehensive introductions on coordinate descent.

1.2.1 Introduction and intuitions

In this section we provide numerical intuitions on the success of coordinate descent. We first remind some properties of gradient descent.

Gradient descent. For Problem (1.25), gradient descent iterations with step size $1/L$ read, for $\beta \in \mathbb{R}^p$:

$$\beta \leftarrow \beta - \frac{1}{L} X^\top (X\beta - y) \ . \tag{1.26}$$

Proposition 1.1. *Gradient descent with step size $1/L$ converges linearly with rate $1 - \mu/L < 1$.*

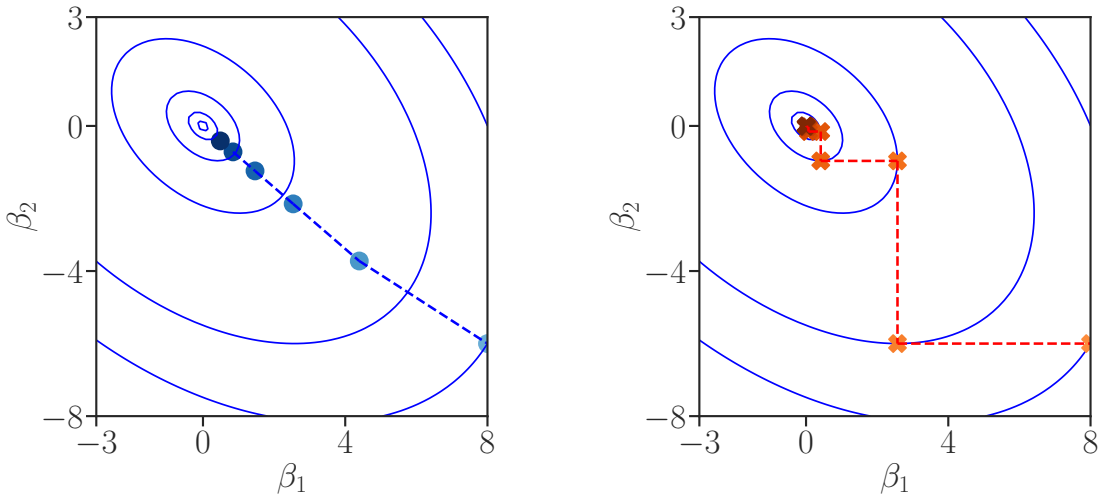
Proof One update of gradient descent writes:

$$\beta \leftarrow \underbrace{(\mathrm{Id}_p - \frac{1}{L} X^\top X)}_{\triangleq T^{\mathrm{GD}}} \beta - \frac{1}{L} X^\top y \ . \tag{1.27}$$

The spectral radius of T^{GD} has a closed-form: $\rho(T^{\mathrm{GD}}) = 1 - \mu/L < 1$, thus gradient descent linearly converges to the unique fixed point of $\beta \mapsto \beta - \frac{1}{L} X^\top (X\beta - y)$, which is also the minimizer of Problem (1.25) (Polyak, 1987, Thm. 1, Sec. 2.1.2). ■

In particular, the condition number L/μ of the Hessian $H \triangleq X^\top X$ of Problem (1.25) controls the convergence of the gradient descent algorithm. We will see that coordinate descent convergence rate is governed by other quantities (Proposition 1.10), the Lipschitz constants L_j of $\nabla_j f_{|\beta_j}$, defined as follows:

$$\text{for all } \beta \in \mathbb{R}^p,\ h \in \mathbb{R},\ |\nabla_j f(\beta + h e_j) - \nabla_j f(\beta)| \leq L_j |h| \ .$$



(a) Gradient descent.

(b) Coordinate descent.

Figure 1.6 – Five iterations of gradient descent (left in blue) and cyclic coordinate descent (right in red) to minimize the quadratic function $f(\beta_1, \beta_2) = 7\beta_1^2 + 6\beta_1\beta_2 + 8\beta_2^2$. Contour lines of the function f are in blue. Each algorithm starts at point $(8, -6)$ and converges toward the minimizer $(0, 0)$. The darker the point, the larger the iteration number. From a computational point of view, two updates of coordinate descent are equivalent to one update of gradient descent (see Algorithm 1.4).

Coordinate descent. The core idea of coordinate descent is to solve an optimization problem through the resolution of smaller dimension subproblems. These subproblems are usually easier and cheaper to solve. More formally, for a function $f : \mathbb{R}^p \to \mathbb{R}$, the idea is to minimize successively one dimensional functions $f_{|\beta_j} : \mathbb{R} \to \mathbb{R}$, updating only one coordinate at a time, while the others remain unchanged. There exist many variants of coordinate descent algorithms, the main branchings being:

- **The index selection.** Many procedures exist to select the coordinate to update, indices can be selected:
  - Cyclically in the set $[p] \triangleq \{1, \dots, p\}$ (Tseng and Yun, 2009a).
  - At random following a given distribution (Nesterov, 2012).
  - Greedily, to optimize a given criterion (Nutini et al., 2015): largest decrease of the objective function, or largest gradient norm (Gauss-Southwell rule, Southwell 1940).
- **The update rule.** The chosen coordinate can be updated in multiple manners:
  - Exact minimization of the function $f_{|\beta_j}$ (Tseng, 2001).
  - Coordinate gradient descent update (Tseng and Yun, 2009a).

Tables 1.3 and 1.4 summarize index selection and update rule variants. In this introduction we focus on cyclic coordinate descent with coordinate gradient descent update rule: for Problem (1.25), for the j^{th} coordinate, with step size $1/L_j$, it reads

$$\beta_j \leftarrow \beta_j - \frac{1}{L_j} X_{:j}^\top (X\beta - y) \ . \tag{1.28}$$

Index selection	Description
Cyclic	$j = (k \bmod p) + 1$
Random (sampling with replacement)	j is chosen randomly in $[p]$ following $\mathbb{P}(j = j') = 1/L_{j'}$
Gauss-Southwell	$j = \arg\max_{j'} \|\nabla_{j'} f(\beta)\|$

Table 1.3 – **Index selection.**

Update rule	Description
Exact	$\beta_j \leftarrow \arg\min_{\beta_j} f_{\|\beta_j}$
Coordinate gradient descent	$\beta_j \leftarrow \beta_j - \frac{1}{L_j} \nabla_j f(\beta)$

Table 1.4 – **Update rule.**

Five iterations of gradient descent and cyclic coordinate descent on a quadratic function are displayed in Figure 1.6. While gradient descent iterates move towards the minimum orthogonally to the level sets, coordinate descent iterates move successively along the axes β_1 and β_2. On this example the values of the coordinate specific Lipschitz constants of $\nabla_j f_{|\beta_j}$ are $L_1 = 7$, $L_2 = 8$, and the the global Lipschitz constant is larger: $L \approx 10.5$: coordinate descent uses larger coordinate specific stepsizes than gradient descent. Figure 1.8 will underline stepsizes importance. First we show that for Problem (1.25) L_j has a closed-form formula.

Proposition 1.2. *With $f(\beta) = \frac{1}{2}\|y - X\beta\|^2$, the best Lipschitz constant L_j of $\nabla f_{|\beta_j}$ is:*

$$L_j = \|X_{:j}\|^2 \ . \tag{1.29}$$

Proof The gradient of the j^{th} coordinate of f writes

$$\nabla_j f(\beta) = X_{:j}^\top (X\beta - y) \ .$$

Then the gradient of $\nabla_j f(\beta)$ restricted to β_j writes

$$\nabla^2_{j,j} f(\beta) = X_{:j}^\top X_{:j} = \|X_{:j}\|^2 \ .$$

Since $\nabla_j f_{|\beta_j}$ is differentiable with continuous derivatives, $\|X_{:j}\|^2$ is a valid Lipschitz constant for $\nabla_j f_{|\beta_j}$. ■

Examples, algorithmic and implementation details. Algorithms 1.1 to 1.3 are the instantiations of gradient descent, Kaczmarz algorithm (Kaczmarz 1937; Strohmer and Vershynin 2009, which can be seen as stochastic gradient descent SGD, Robbins and Monro 1951 with constant step size), and coordinate gradient descent on the least squares Problem (1.25).

From an algorithmic point of view, one naive update of coordinate descent (Algorithm 1.3) has a $\mathcal{O}(np)$ computation cost per iteration, as it requires computing $X\beta$. In the case of Problem (1.25) it is possible to make this update $\mathcal{O}(n)$ per iteration (Algorithm 1.4). This cheap update trick of coordinate descent generalizes to other sparse linear models, see Friedman et al. 2010 for details.

Whereas coordinate descent iterates on the columns $X_{:j}$ of X, stochastic gradient descent (or Kaczmarz) iterates over lines $X_{i:}$ of the design matrix X (Figure 1.7). From an implementation point of view, accessing efficiently the columns $X_{:j}$ of X requires the

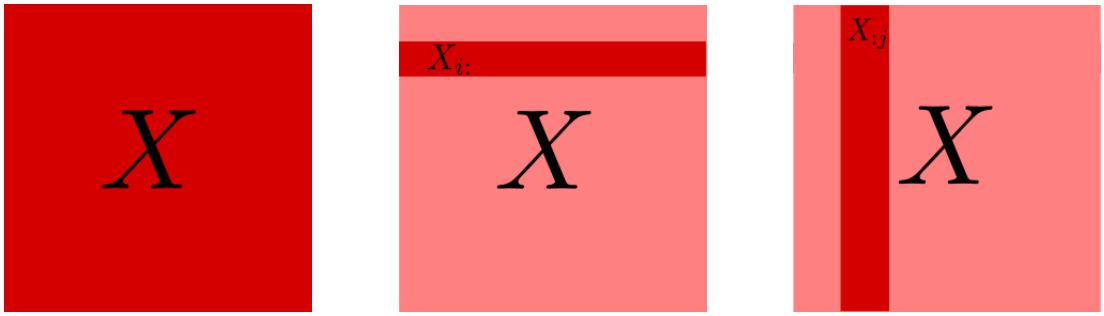


Figure 1.7 – **Matrix storage.** For coordinate descent (right), one needs to access efficiently successively each column $X_{:j}$ of $X \in \mathbb{R}^{n \times p}$. On the opposite, when applying Kaczmarz or SGD (middle), one needs to access efficient each line $X_{i:}$ of X.

design matrix X to be stored in *Fortran order*[3]. On the opposite accessing efficiently the lines $X_{i:}$ of X requires the design matrix X to be stored in *C order*. When dealing with sparse matrices, this means that the design matrix X should be stored is *compressed sparse column (CSC)*[4] format for coordinate descent, and *compressed sparse row (CSR)* for Kaczmarz.

Algorithm 1.1 Gradient descent

init : $\beta \in \mathbb{R}^p$
for $k = 0, 1, \dots,$ **do**
  $\beta \leftarrow \beta - \frac{1}{\|X\|_2^2} X^\top (X\beta - y)$ // $\mathcal{O}(np)$
return β

Algorithm 1.2 Kaczmarz

init : $\beta \in \mathbb{R}^p$
for $k = 0, 1, \dots,$ **do**
  Select a sample $i \in [n]$
  $\beta \leftarrow \beta - \frac{1}{\|X_{i:}\|^2} X_{i:}^\top (X_{i:}\beta - y_i)$ // $\mathcal{O}(p)$
return β

Algorithm 1.3 Naive CD

init : $\beta \in \mathbb{R}^p$
for $k = 0, 1, \dots,$ **do**
  Select a coordinate $j \in [p]$
  $\beta_j \leftarrow \beta_j - \frac{1}{\|X_{:j}\|^2} X_{:j}^\top (X\beta - y)$ // $\mathcal{O}(np)$
return β

Algorithm 1.4 Efficient CD

init : $\beta \in \mathbb{R}^p$, $r = y - X\beta$
for $k = 0, 1, \dots,$ **do**
  Select a coordinate $j \in [p]$
  $\beta_j^{\text{old}} \leftarrow \beta_j$ // $\mathcal{O}(1)$
  $\beta_j \leftarrow \beta_j + \frac{1}{\|X_{:j}\|^2} X_{:j}^\top r$ // $\mathcal{O}(n)$
  `// update the residuals r`
  $r \mathrel{-}= X_{:j}(\beta_j - \beta_j^{\text{old}})$ // $\mathcal{O}(n)$
return β

A quick review of coordinate descent. The index selection (Table 1.3) and the update rule (Table 1.4) branchings led to a plethora of results for coordinate descent (Fercoq and Richtárik, 2015, Tab. 1). Some landmark results for cyclic coordinate descent include Luo and Tseng (1992); Tseng (2001); Tseng and Yun (2009a); Razaviyayn et al. (2013) who have shown convergence results for cyclic coordinate descent for nonsmooth optimization problems. Then, Beck and Tetruashvili (2013) showed $1/k$ convergence rates for convex Lipschitz smooth functions and linear convergence rates in the smooth strongly convex case. Among the results on random coordinate descent

[3] see https://scipy-lectures.org/advanced/advanced_numpy/ for details.
[4] see https://scipy-lectures.org/advanced/scipy_sparse/index.html for details.

one can refer to Nesterov (2012) for the minimization of a smooth function f. It proved global non-asymptotic $1/k$ in expectation convergence rate in the case of a smooth and convex f. This work was later extended to composite optimization $f + \sum_j g_j$ for nonsmooth separable functions (Richtárik and Takáč, 2014; Fercoq and Richtárik, 2015). Refined convergence rates were also shown by Shalev-Shwartz and Tewari (2011); Shalev-Shwartz and Zhang (2013b).

Some branchings simplify for quadratic problems: exact minimization and coordinate gradient descent actually coincide.

Proposition 1.3. *On Problem (1.25) exact coordinate descent and coordinate gradient descent lead to the same update rule.*

Proof

$$\begin{aligned}
\arg\min_{\beta_j} \frac{1}{2}\|y - X\beta\|^2 &= \arg\min_{\beta_j} \frac{1}{2}\|y - \sum_{j'=1}^{p} \beta_{j'} X_{:j'}\|^2 \\
&= \arg\min_{\beta_j} \frac{1}{2}\|y - \sum_{j' \neq j} \beta_{j'} X_{:j'} - \beta_j X_{:j}\|^2 \\
&= \arg\min_{\beta_j} \frac{1}{2}\|\beta_j X_{:j}\|^2 - \langle y - \sum_{j' \neq j} \beta_{j'} X_{:j'}, \beta_j X_{:j} \rangle \\
&= \arg\min_{\beta_j} \frac{1}{2}\beta_j^2 \|X_{:j}\|^2 - \langle X_{:j}^\top (y - \sum_{j' \neq j} \beta_{j'} X_{:j'}), \beta_j \rangle \\
&= \arg\min_{\beta_j} \frac{1}{2}\Big(\beta_j - \frac{1}{\|X_{:j}\|^2} X_{:j}^\top (y - \sum_{j' \neq j} \beta_{j'} X_{:j'})\Big)^2 \\
&= \frac{1}{\|X_{:j}\|^2} X_{:j}^\top (y - X\beta + \beta_j X_{:j}) \\
&= \beta_j - \frac{1}{\|X_{:j}\|^2} X_{:j}^\top (X\beta - y) \ . \qquad (1.30)
\end{aligned}$$

■

As shown in Algorithms 1.1 and 1.4, one update of gradient descent costs $\mathcal{O}(np)$, whereas on update of coordinate descent costs $\mathcal{O}(n)$: in order to do fair comparisons, one must compare one update of gradient descent, and p updates of coordinate descent.

Definition 1.4 (Epoch)**.** *An epoch is p updates of coordinate descent. For the cyclic index selection it corresponds to exactly one update of each coordinate.*

Intuition on the efficiency of coordinate descent (Figure 1.8). Since $\|X_{:j}\| = \|Xe_j\| \leq \sup_{u \in \mathbb{R}^p | \|u\| \leq 1} \|Xu\| = \|X\|_2$, the coordinate specific step size γ_j, $j \in [p]$, is larger than the step size γ of gradient descent: $\gamma_j = 1/\|X_{:j}\|^2 \geq 1/\|X\|_2^2 = \gamma$. To confirm the influence of the step size in coordinate descent success let us do a simple experiment: Figure 1.8 compares the performance of cyclic coordinate descent with multiple values of step size. Gradient descent is compared against cyclic coordinate descent with step sizes $\gamma_j = \delta/L_j + (1-\delta)/L$, for multiple values of δ. When $\delta = 1$ one recovers the usual coordinate descent with step size $\gamma_j = 1/L_j$, when $\delta = 0$, one obtains coordinate descent with step size $\gamma_j = 1/L$. One can see that the performance of

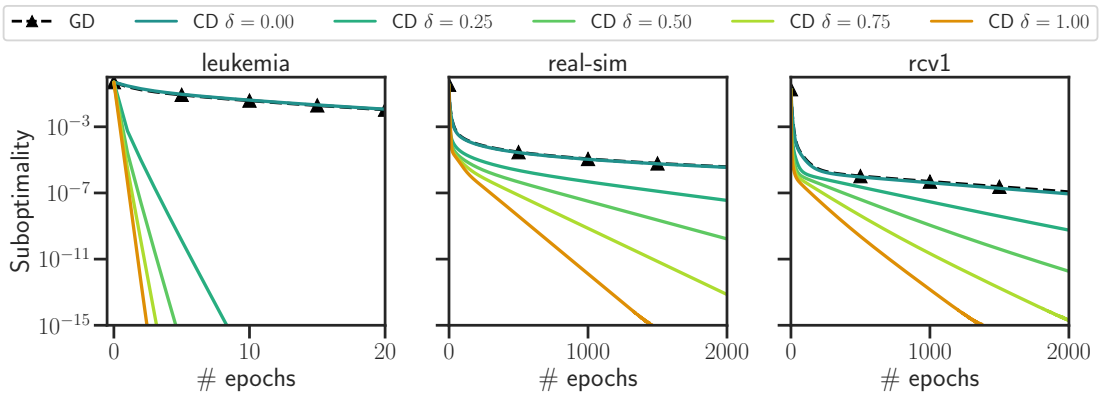


Figure 1.8 – **Influence of stepsizes for cyclic coordinate descent, OLS.** Suboptimality $f(\beta^{(k)}) - f(\hat{\beta})$ as a function of the number of epochs to solve a least squares problem on multiple datasets from `libsvm`: *leukemia*, *real-sim* and *rcv1*. Gradient descent is compared against coordinate descent with step sizes $\gamma_j = \delta/L_j + (1-\delta)/L$, for multiple values of δ.

coordinate descent with step size $1/L$ is similar to the performance of gradient descent. The larger the step size, up to $1/L_j$, the faster the convergence.

Figure 1.8 suggests that cyclic coordinate descent converges linearly on the least squares Problem (1.25), and surpasses gradient descent on real-world datasets. Natural questions are: is it possible to show linear convergence for coordinate descent? If yes, is it possible to show the convergence rate of coordinate descent is better than the convergence rate of gradient descent?

1.2.2 Theoretical results on least squares

First, we recall cyclic coordinate descent (Algorithm 1.4) converges.

Proposition 1.5 (Convergence, Bertsekas 2015, Prop. 6.5.1)**.** *Cyclic coordinate descent converges to the minimizer of Problem (1.25).*

In addition, one can show linear convergence for cyclic coordinate descent.

Proposition 1.6 (Linear convergence of cyclic CD)**.** *Cyclic coordinate descent converges linearly on Problem (1.25).*

Proof The proof will be in two parts:

- First we will show that one epoch of cyclic coordinate descent leads to a linear iteration (as for one iteration of gradient descent, Proposition 1.1).
- Then we show that the iteration matrix of cyclic coordinate descent has its spectral radius strictly smaller than 1, implying that cyclic coordinate descent converges linearly.

Lemma 1.7 (Linear iteration)**.** *On Problem (1.25), one epoch of cyclic coordinate descent can be seen as a linear iteration:*

$$\beta \leftarrow T^{\mathrm{CD}}\beta + b^{\mathrm{CD}} \quad , \tag{1.31}$$

$$T^{\mathrm{CD}} = (\mathrm{Id}_p - \gamma_p e_p e_p^\top X^\top X) \dots (\mathrm{Id}_p - \gamma_1 e_1 e_1^\top X^\top X) \in \mathbb{R}^{p \times p} \ , \tag{1.32}$$

with $\gamma_j = 1/\|X_{:j}\|^2$*, and* $b^{\mathrm{CD}} \in \mathbb{R}^p$*.*

Proof The update of the coordinate j writes:

$$\begin{aligned}
\beta_j &\leftarrow \beta_j - \gamma_j X_{:j}^\top (X\beta - y) \\
\beta &\leftarrow \beta - \gamma_j X_{:j}^\top (X\beta - y) e_j \\
\beta &\leftarrow \beta - \gamma_j e_j X_{:j}^\top X\beta - \gamma_j X_{:j}^\top y e_j \\
\beta &\leftarrow \beta - \gamma_j e_j e_j^\top X^\top X\beta - \gamma_j X_{:j}^\top y e_j \\
\beta &\leftarrow (\mathrm{Id}_p - \gamma_j e_j e_j^\top X^\top X)\beta - \gamma_j X_{:j}^\top y e_j \ .
\end{aligned} \tag{1.33}$$

Updating the coordinate from $j = 1$ to $j = p$, this leads to:

$$\beta \leftarrow \underbrace{(\mathrm{Id}_p - \gamma_p e_p e_p^\top X^\top X) \dots (\mathrm{Id}_p - \gamma_1 e_1 e_1^\top X^\top X)}_{\triangleq T^{\mathrm{CD}}} \beta + b^{\mathrm{CD}} \ , \tag{1.34}$$

for some $b^{\mathrm{CD}} \in \mathbb{R}^p$. ■

We have shown that one epoch of cyclic coordinate descent could be seen as a linear iteration with iteration matrix T^{CD}. In the next lemma we that the modulus of its larger eigenvalue is strictly smaller than 1.

Lemma 1.8. *The spectral radius of the iteration matrix of cyclic coordinate descent* T^{CD} *is strictly smaller than* 1*:*

$$\rho(T^{\mathrm{CD}}) < 1 \ . \tag{1.35}$$

Proof With $H = X^\top X \succ 0$,

$$\begin{aligned}
\rho(T^{\mathrm{CD}}) &= \rho((\mathrm{Id}_p - \gamma_p e_p e_p^\top H) \dots (\mathrm{Id}_p - \gamma_1 e_1 e_1^\top H)) \\
&= \rho(H^{-1/2}(\mathrm{Id}_p - \gamma_p H^{1/2} e_p e_p^\top H^{1/2}) \dots (\mathrm{Id}_p - \gamma_1 H^{1/2} e_1 e_1^\top H^{1/2}) H^{1/2}) \\
&= \rho((\mathrm{Id}_p - \gamma_p H^{1/2} e_p e_p^\top H^{1/2}) \dots (\mathrm{Id}_p - \gamma_1 H^{1/2} e_1 e_1^\top H^{1/2})) \ .
\end{aligned}$$

For $j \in [p]$, let $T^{(j)} \triangleq (\mathrm{Id}_p - \gamma_j H^{1/2} e_j e_j^\top H^{1/2})$, $T^{(j)}$ is the orthogonal projection onto $\mathrm{Span}(H^{1/2} e_j)^\perp$, we thus have $\|T^{(j)}\|_2 \leq 1$, and

$$\rho(T^{(p)} \times \dots \times T^{(1)}) = \|T^{(p)} \times \dots \times T^{(1)}\|_2 \leq \|T^{(p)}\|_2 \times \dots \times \|T^{(1)}\|_2 \leq 1 \ . \tag{1.36}$$

Suppose there exists a (potentially complex) eigenvalue $\delta \in \mathbb{C}$ and an eigenvector $\beta \in \mathbb{C}^p \neq 0$ of $T^{(p)} \times \dots \times T^{(1)}$ such that $|\delta| = 1$, we have $\|T^{(p)} \times \dots \times T^{(1)}\beta\|_2 = \|\beta\|_2$.

Since $\|T^{(1)}\| \leq 1$, we have $\|T^{(1)}\beta\| = 1$, leading to $\beta \in \mathrm{Span}(H^{1/2} e_1)^\perp$ and $T^{(1)}\beta = \beta$. Recursively we have that $\beta \in \mathrm{Span}(H^{1/2} e_1, \dots, H^{1/2} e_p)^\perp = \{0\}$, which is a contradiction.

To conclude we have $\rho(T^{(p)} \times \dots \times T^{(1)}) \leq 1$ and $T^{(p)} \times \dots \times T^{(1)}$ has no eigenvalue of modulus 1, thus,

$$\rho(T^{\mathrm{CD}}) = \rho(T^{(p)} \times \dots \times T^{(1)}) < 1 \ . \tag{1.37}$$

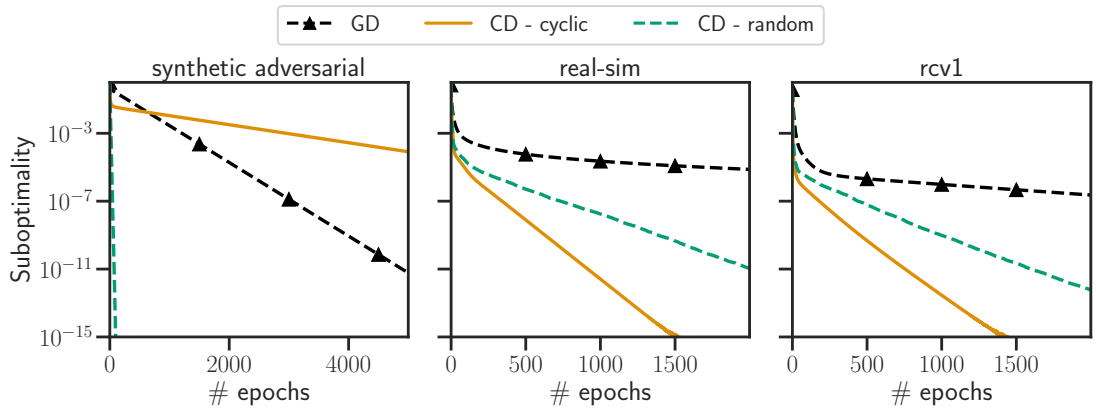


Figure 1.9 – **Cyclic against random coordinate descent.** Suboptimality as a function of the number of epochs to solve a least squares problems. Gradient descent is compared against random and cyclic coordinate descent.

Using Polyak (1987, Theorem 1, Section 2.1.2), cyclic coordinate descent converges linearly. ■

We just showed the linear convergence of cyclic coordinate descent on the simple Problem (1.25). The reasoning above is extended in Chapter 2 to show the local linear convergence of cyclic coordinate descent on Problem (1.24). ■

Does coordinate descent outperforms gradient descent? In other words, for the same cost per update, how do the worst-case convergence rates of gradient and coordinate descent compare? Unfortunately, the answer will depend on the index selection procedure. While being used as a default mode for nonsmooth optimization problems in all preeminent machine learning packages, cyclic coordinate has a poor worst-case convergence rate (Beck and Tetruashvili, 2013). Indeed, it is possible to construct adversarial examples to deteriorate the performance of coordinate descent.

Example 1.9 (Sun and Ye 2019)**.** *For $c \in \left]0, 1\right[$, we define the quadratic problem:*

$$\min_{\beta \in \mathbb{R}^p} \frac{1}{2} \beta^\top \begin{pmatrix} 1 & c & \dots & c \\ c & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & c \\ c & \dots & c & 1 \end{pmatrix} \beta \ . \tag{1.38}$$

On this objective, as c approaches 1, cyclic coordinate descent is $\mathcal{O}(p^2)$ times slower than randomized coordinate descent. Figure 1.9, left, shows the suboptimality as a function of the number of epochs on Example 1.9, with $c = 0.8$ and $p = 100$. Regression coefficients β are initialized with random Gaussian i.i.d. entries $\beta \sim \mathcal{N}(0, 1)$. On this specific example, coordinate descent with cyclic index selection performs worse than gradient descent. However, coordinate descent with random index selection performs much better than gradient descent. Theoretically, it is possible to show that coordinate descent with random index selection perfoms better than gradient descent in expectation.

Proposition 1.10 (Convergence rates on quadratics, Sun and Ye 2019)**.** *For L-smooth μ-strongly convex quadratic functions f, gradient descent (GD), random (RCD) and cyclic (CCD) coordinate descent converge with the following per epochs complexity rates:*

$$f\left(\beta_{\mathrm{GD}}^{(k+1)}\right) - f\left(\hat{\beta}\right) \leq \exp\left(-\frac{\mu}{L}\right)\left(f\left(\beta_{\mathrm{GD}}^{(k)}\right) - f\left(\hat{\beta}\right)\right) \quad , \tag{1.39}$$

$$\mathbb{E}f\left(\beta_{\mathrm{RCD}}^{(k+1)}\right) - f\left(\hat{\beta}\right) \leq \exp\left(-\frac{p\mu}{\sum_{j=1}^{p} L_j}\right)\left(\mathbb{E}f\left(\beta_{\mathrm{RCD}}^{(k)}\right) - f\left(\hat{\beta}\right)\right) \quad , \tag{1.40}$$

$$f(\beta_{\mathrm{CCD}}^{(k+1)}) - f(\hat{\beta}) \leq \exp\left(-\frac{\mu}{p\sum_{j=1}^{p} L_j}\right)\left(f(\beta_{\mathrm{CCD}}^{(k)}) - f(\hat{\beta})\right) \quad , \tag{1.41}$$

and these rates are tight.

Remark 1.11 (Proposition 1.10)**.** *Since $\sum_{j=1}^{p} L_j/p \leq L \leq \sum_{j=1}^{p} L_j$, we have:*

$$\underbrace{\exp\left(-\frac{p\mu}{\sum_{j=1}^{p} L_j}\right)}_{\mathrm{RCD}} \leq \underbrace{\exp\left(-\frac{\mu}{L}\right)}_{\mathrm{GD}} \leq \underbrace{\exp\left(-\frac{\mu}{p\sum_{j=1}^{p} L_j}\right)}_{\mathrm{CCD}} \quad . \tag{1.42}$$

The worst-case convergence rate of random coordinate descent is better than the one of gradient descent, which is itself better than the one of cyclic coordinate descent. Note that `scikit-learn` *and* `glmnet` *implement the cyclic index selection procedure: on real-life datasets cyclic coordinate descent seems to perform better (Figure 1.9, middle and right). This phenomenon is understood in some very specific cases, when the Hessian is very structured (Gurbuzbalaban et al., 2017), but remains largely unexplained in general.*

1.2.3 Beyond quadratics

Coordinate descent for ordinary least squares (Algorithm 1.4) can be generalized for nonquadratic smooth optimization problems (Nesterov, 2012; Beck and Tetruashvili, 2013) and for composite Problem (1.24) using its proximal variation (PCD, Richtárik and Takáč 2014, Algorithm 1.5). Note that for sparse generalized linear models (GLMs), cheap coordinate descent updates are possible. If for all $\beta \in \mathbb{R}^p$, $f(\beta) = F(X\beta)$, for some convex smooth function F, then an update of coordinate descent can be made $\mathcal{O}(n)$ (Algorithm 1.6).

Algorithm 1.5 Naive PCD

init : $\beta \in \mathbb{R}^p$, $L_j > 0$
for $k = 0, 1, \ldots,$ **do**
  Select a coordinate $j \in [p]$
  $\beta_j \leftarrow \mathrm{prox}_{\frac{g_j}{L_j}}(\beta_j - \frac{1}{L_j}\nabla_j f(\beta))$ // $\mathcal{O}(np)$
return β

Algorithm 1.6 Efficient PCD for GLMs

init : $\beta \in \mathbb{R}^p$, $r = X\beta$, $L_j > 0$
for $k = 0, 1, \ldots,$ **do**
  Select a coordinate $j \in [p]$
  $\beta_j^{\mathrm{old}} \leftarrow \beta_j$ // $\mathcal{O}(1)$
  $\beta_j \leftarrow \mathrm{prox}_{\frac{g_j}{L_j}}(\beta_j - \frac{1}{L_j}X_{:j}^\top \nabla F(X\beta))$ // $\mathcal{O}(n)$
  // update efficiently $r = X\beta$
  $r \mathrel{+}= X_{:j}(\beta_j - \beta_j^{\mathrm{old}})$ // $\mathcal{O}(n)$
return β

1.3 Contributions

This thesis is organized as follows: in Part I we investigate some theoretical and empirical properties of coordinate descent to solve "smooth + nonsmooth separable" optimization problems. In Part II we explore the hyperparameter selection from a statistical point of view: we show that for some estimators, the regularization parameter is independent from the noise level. We also provide applications to the brain source localization problem on real M/EEG data. In Part III we present multiple ways of applying first-order methods to tackle bilevel optimization problems with nonsmooth inner optimization problems: these methods are applied to hyperparameter optimization.

Note that each chapter is self-sufficient and can be read independently. In this thesis we strongly emphasize on numerical contributions as well as reproducibility: each part comes with extensively documented and tested open source code.

More precisely in Part I:

- Chapter 2 introduces two properties of coordinate descent: under some mild assumptions on the composite nonsmooth optimization Problem (1.24), it is shown finite time *support identification* and *local linear convergence*. For example, zeros coefficients of Lasso solutions $\hat{\beta}$ are exactly attained after some iteration K: with $\mathcal{S} = \{j \in [p] : \hat{\beta}_j \neq 0\}$, $\beta^{(k)}$ the iterates of coordinate descent, there exists $K \in \mathbb{N}$, such that for all $k \geq K$, for all $j \in \mathcal{S}^c$, $\beta_j^{(k)} = \hat{\beta}_j = 0$. After support identification, there exists a regime where the iterates $\beta^{(k)}$ converge linearly toward $\hat{\beta}$. This properties are extensively used for the rest of the thesis, in particular in Chapter 3 for acceleration, and in Chapter 6 for hyperparameter optimization.

- In Chapter 3, we investigate Anderson acceleration for coordinate descent. First we illustrate that *à la Nesterov* acceleration for coordinate descent can lead to poor convergence in practice: this contrasts with its optimal theoretically accelerated convergence rate. Then we show numerically that usual proofs of Anderson accelerated rates cannot be applied for coordinate descent, and we provide a way to theoretically accelerate coordinate descent on quadratic optimization problems. Finally, we illustrate extensively the performance of Anderson accelerated coordinate descent on a large number of optimization problems, datasets, and experimental settings. The approach proposed in this chapter can be combined with working sets techniques, and generalized to numerous data fitting terms and penalties implemented in a modular open source package `andersoncd`: https://github.com/mathurinm/andersoncd. The proposed algorithm is also now implemented[5] as the default solver in the preeminent brain signal processing package `MNE`.

In Part II:

- Chapter 4: in this work we study estimators for which the regularization parameter is independent of the noise level. These estimators rely on challenging "nonsmooth + nonsmooth" optimization problems. One practical way of computing such estimators is to smooth the data fitting term: the larger the smoothing term, the easier the optimization, but the more modified the initial estimator. Provided

[5] https://github.com/mne-tools/mne-python/commit/080e7a879d325ad8f0c11fe28a8ff9f5983dfd2f

that the smoothing parameter is not too large, we show that partial smoothing preserves the support recovery and pivotal properties. In addition we precisely quantify how large the smoothing parameter can be, leading to practical guidelines to calibrate this hyperparameter.

- In Chapter 5 we propose an application of pivotal estimators to the M/EEG source localization problem. We introduce an estimator based on an optimization problem with a smoothed trace norm as a data fitting term and an $\ell_{2,1}$ norm as a regularizer. In addition to be independent to the noise level, the proposed estimator can take advantage of the M/EEG data multiple repetitions, and is designed to take into account Gaussian correlated noise. Last but not least, thanks to partial smoothing, one can efficiently apply block coordinate descent algorithms to solve the underlying optimization problem. Extensive experiments on real M/EEG data show the interest of the this estimator on visual and auditory tasks. Code and documentation can be found online: `https://github.com/QB3/CLaR`. The proposed algorithm should soon be implemented in the `jaxopt` library (Blondel et al., 2021), which extends the automatic differentiation library `jax` (Bradbury et al., 2018), and allows to differentiate through solutions of optimization problems.

In Part III:

- In Chapter 6 we study the most popular hyperparameter selection approach in machine learning, encompassing hold-out and cross-validation: hyperparameter optimization. Our approach models this step as a challenging bilevel optimization with nonsmooth inner problems. Usual solvers usually rely on zeros order algorithms, whose complexity scales exponentially with the number of hyperparameters. To circumvent this problem, we propose an efficient implicit differentiation algorithm to compute the *hypergradient*. The bilevel optimization problem can then be solved using first-order methods. The proposed approach can be applied to set the regularization parameter for a wide range of estimators such as the Lasso, the elastic net, or the SVM: efficient implementation and extensive documentation can be found here: `https://github.com/QB3/sparse-ho`.

1.4 Publications

The work presented in this thesis led to the following publications:

- **Q. Bertrand**, M. Massias, A. Gramfort, and J. Salmon. Handling correlated and repeated measurements with the smoothed multivariate square-root Lasso. In *NeurIPS*, 2019

- M. Massias, **Q. Bertrand**, A. Gramfort, and J. Salmon. Support recovery and sup-norm convergence rates for sparse pivotal estimation. In *AISTATS*, 2020a

- **Q. Bertrand**, Q. Klopfenstein, M. Blondel, S. Vaiter, A. Gramfort, and J. Salmon. Implicit differentiation of Lasso-type models for hyperparameter optimization. *ICML*, 2020

- **Q. Bertrand** and M. Massias. Anderson acceleration of coordinate descent. In *AISTATS*, 2021

- Q. Klopfenstein, **Q. Bertrand**, A. Gramfort, J. Salmon, and S. Vaiter. Model identification and local linear convergence of coordinate descent. *arXiv preprint arXiv:2010.11825*, 2020

- **Q. Bertrand**, Q. Klopfenstein, M. Massias, M. Blondel, S. Vaiter, A. Gramfort, and J. Salmon. Implicit differentiation for fast hyperparameter selection in non-smooth convex learning. *arXiv preprint arXiv:2105.01637*, 2021

Part I

Some properties of coordinate descent

2

Model identification and local linear convergence

Contents

For composite nonsmooth optimization problems, which are *regular enough*, proximal gradient descent achieves model identification after a finite number of iterations. For instance, for the Lasso, this implies that the iterates of proximal gradient descent identify the non-zeros coefficients after a finite number of steps. Identification properties often rely on the framework of *partial smoothness*. In this work we show simple sufficient conditions to be a partial smooth function when the nonsmooth penalty is separable. In this simplified framework, we show cyclic coordinate descent achieves model identification in finite time, which yields explicit local linear convergence rates. Theses two properties are paramount for other works in this thesis, in particular to design accelerated algorithms for coordinate descent (Chapter 3) and to perform gradient-based hyperparameter optimization with nonsmooth inner problems (Chapter 6).

This chapter is based on the following work

- Q. Klopfenstein, **Q. Bertrand**, A. Gramfort, J. Salmon, and S. Vaiter. Model identification and local linear convergence of coordinate descent. *arXiv preprint arXiv:2010.11825*, 2020

2.1 Introduction

Over the last two decades, coordinate descent (CD) algorithms have become a powerful tool to solve large scale optimization problems (Friedman et al., 2007, 2010). Many applications coming from machine learning or compressed sensing have lead to optimization problems that can be solved efficiently via CD algorithms: the Lasso (Tibshirani, 1996; Chen et al., 1998), the elastic net (Zou and Hastie, 2005) or support-vector machine (Boser et al., 1992). All the previously cited estimators are based on an op-

timization problem which can be written:

$$\hat{\beta} \in \arg\min_{\beta \in \mathbb{R}^p} \{\Phi(\beta) \triangleq f(\beta) + \underbrace{\sum_{j=1}^{p} g_j(\beta_j)}_{\triangleq g(\beta)}\} \ , \tag{2.1}$$

with f a convex smooth (*i.e.,* with a Lipschitz gradient) function and g_j proper closed and convex functions. In the past twenty years, the popularity of CD algorithms has greatly increased due to the well suited structure of the new optimization problems mentioned above (*i.e.,* separability of the nonsmooth term), as well as the possible parallelization of the algorithms (Fercoq and Richtárik, 2015; Richtárik and Takáč, 2016; Salzo and Villa, 2021).

The key idea behind CD (Algorithm 2.1) is to solve small and simple subproblems iteratively until convergence. More formally, for a function $\Phi : \mathbb{R}^p \mapsto \mathbb{R}$, the idea is to minimize successively one dimensional functions $\Phi_{|\beta_j} : \mathbb{R} \mapsto \mathbb{R}$, updating only one coordinate at a time, while the others remain unchanged. There exists many variants of CD algorithms, the main branching being:

- **The index selection.** There are different ways to choose the index of the updated coordinate at each iteration. The main variants can be divided in three categories, **cyclic** CD (Tseng and Yun, 2009a) when the indices are chosen in the set $[p] \triangleq \{1, \dots, p\}$ cyclically. **Random** CD (Nesterov, 2012), where the indices are chosen following a given random distribution. Finally, **greedy** CD (Nutini et al., 2015) *picks* an index, optimizing a given criterion: largest decrease of the objective function, or largest gradient norm (Gauss-Southwell rule), for instance.

- **The update rule.** There also exists several possible schemes for the coordinate update: exact minimization, coordinate gradient descent or prox-linear update (see Shi et al. 2016, Sec. 2.2 for details).

In this work, we will focus on the most popular combination: **cyclic** CD with **prox-linear** update rule (Algorithm 2.1), implemented in popular packages such as `glmnet` (Friedman et al., 2007) or `sklearn` (Pedregosa et al., 2011).

Among the methods of coordinate selection, **random** CD has been extensively studied, especially by Nesterov (2012) for the minimization of a smooth function f. It was the first paper proving global non-asymptotic $1/k$ convergence rate in the case of a smooth and convex f. This work was later extended to composite optimization $f + \sum_j g_j$ for nonsmooth separable functions (Richtárik and Takáč, 2014; Fercoq and Richtárik, 2015). Refined convergence rates were also shown by Shalev-Shwartz and Tewari (2011); Shalev-Shwartz and Zhang (2013b). These convergence results have then been extended to coordinate descent with equality constraints (Necoara and Patrascu, 2014) that induce non-separability as found in the SVM dual problem in the presence of the bias term. Different distributions have been considered for the index selection such as uniform distribution (Fercoq and Richtárik, 2015; Nesterov, 2012; Shalev-Shwartz and Tewari, 2011; Shalev-Shwartz and Zhang, 2013b), importance sampling (Leventhal and Lewis, 2010; Zhang, 2004) and arbitrary sampling (Necoara and Patrascu, 2014; Qu and Richtárik, 2016a,b).

On the opposite, theory on **cyclic** coordinate descent is more fuzzy, the analysis in the cyclic case being more difficult. First, Luo and Tseng (1992); Tseng (2001); Tseng and Yun (2009a); Razaviyayn et al. (2013) have shown convergence results for (block) CD algorithms for nonsmooth optimization problems (without rates[1]). Then, Beck and Tetruashvili (2013) showed $1/k$ convergence rates for Lipschitz convex functions and linear convergence rates in the strongly convex case. Saha and Tewari (2013) proved $1/k$ convergence rates for composite optimization $f + \|.\|_1$ under "isotonicity" condition. Sun and Hong (2015); Hong et al. (2017) have extended the latter results and showed $1/k$ convergence rates with improved constants for composite optimization $f + \sum_j g_j$. Li et al. (2017) have extended the work of Beck and Tetruashvili (2013) to the nonsmooth case and refined their convergence rates in the smooth case. Finally, as far as we know, the work by Xu and Yin (2017) is the first one tackling the problem of local linear convergence. They have proved local linear convergence under the very general Kurdyka-Lojasiewicz hypothesis, relaxing convexity assumptions. Following the line of work by Liang et al. (2014), we use a more restrictive framework, that allows to achieve finer results: model identification as well as improved local convergence results.

2.2 Model identification

Nonsmooth optimization problems coming from machine learning such as the Lasso or the support-vector machine (SVM) generally generate solutions lying onto a low-complexity model (see Definition 2.7 for details). For the Lasso, for example, a solution $\hat{\beta}$ has typically only a few non-zeros coefficients: it lies on the model set $T_{\hat{\beta}} = \{u \in \mathbb{R}^p : \text{supp}(u) \subseteq \text{supp}(\hat{\beta})\}$, where $\text{supp}(\beta)$ is the support of β, *i.e.,* the set of indices corresponding to the non-zero coefficients. A question of interest in the literature is: does the algorithm achieve model identification after a finite number of iterations? Formally, does it exist $K > 0$ such that for all $k > K$, $\beta^{(k)} \in T_{\hat{\beta}}$? For the Lasso the question boils down to "does it exist $K > 0$ such that for all $k > K$, $\text{supp}(\beta^{(k)}) \subseteq \text{supp}(\hat{\beta})$"? This finite time identification property is paramount for features selection (Tibshirani, 1996), but also for potential acceleration methods (Massias et al., 2018b) of the CD algorithm, as well as model calibration (see Chapter 6).

Finite model identification was first proved in Bertsekas (1976) for the projected gradient method with non-negative constraints. In this case, after a finite number of steps the sparsity pattern of the iterates is the same as the sparsity pattern of the solution. It means that for k large enough, $\beta_j^{(k)} = 0$ for all j such that $\hat{\beta}_j = 0$. Then, many other results of finite model identification have been shown in different settings and for various algorithms. For the projected gradient descent algorithm, identification was proved for polyhedral constraints (Burke and Moré, 1988), for general convex constraints (Wright, 1993), and even non-convex constraints (Hare and Lewis, 2004). More recently, identification was proved for proximal gradient algorithm (Lions and Mercier, 1979; Combettes and Wajs, 2005), for the ℓ_1 regularized problem (Hare, 2011). Liang et al. (2014, 2017); Vaiter et al. (2018) have shown model identification and local linear convergence for proximal gradient descent. These results have then been extended to other popular machine learning algorithms such as SAGA, SVRG (Poon et al., 2018) and ADMM (Poon and Liang, 2019), see Iutzeler and Malick (2020) for a review. To our knowledge, apart from the line of work of Nutini et al. (2017); Nutini (2018); Nutini et al. (2019), which we discuss thoroughly in Remark 2.14, CD has not been extensively

[1]Note that some local rates are shown in Tseng and Yun (2009a) but under some strong hypothesis.

studied with a similar generality. Some identification results have been shown for CD, but only on specific models (She and Schmidt, 2017; Massias et al., 2020b) or variants of CD (Wright, 2012), in general, under restrictive hypothesis.

Coordinate descent. We denote $0 < \gamma_j \leq 1/L_j$ the local step size and $\gamma = (\gamma_1, \dots, \gamma_p)^\top$. To prove model identification we need to "keep track" of the iterates: following the notation from Beck and Tetruashvili (2013) coordinate descent can be written:

Algorithm 2.1 Proximal coordinate descent

input : $\gamma_1, \dots, \gamma_p \in \mathbb{R}_+, n_{\text{iter}} \in \mathbb{N}, \beta^{(0)} \in \mathbb{R}^p$
for $k = 0, \dots, n_{\text{iter}}$ **do**
  $\beta^{(0,k)} \leftarrow \beta^{(k)}$
  for $j = 1, \dots, p$ **do** // index selection
    $\beta^{(j,k)} \leftarrow \beta^{(j-1,k)}$
    $\beta_j^{(j,k)} \leftarrow \text{prox}_{\gamma_j g_j}\left(\beta_j^{(j-1,k)} - \gamma_j \nabla_j f(\beta^{(j-1,k)})\right)$ // update rule
  $\beta^{(k+1)} \leftarrow \beta^{(p,k)}$
return $\beta^{n_{\text{iter}}+1}$

We consider the optimization problem defined in Equation (2.1) with the following assumptions:

Assumption 2.1 (Smoothness)**.** *f is a convex and differentiable function, with a Lipschitz gradient.*

Assumption 2.2 (Proper, closed, convex)**.** *For any $j \in [p], g_j$ is proper, closed and convex.*

Assumption 2.3 (Existence)**.** *The problem admits at least one solution:*

$$\arg\min_{\beta \in \mathbb{R}^p} \Phi(\beta) \neq \emptyset \ . \tag{2.2}$$

Assumption 2.4 (Non degeneracy)**.** *The problem is non-degenerate: for any $\hat{\beta} \in \arg\min_{x \in \mathbb{R}^p} \Phi(x)$*

$$-\nabla f(\hat{\beta}) \in \text{ri}\left(\partial g(\hat{\beta})\right) \ . \tag{2.3}$$

Remark 2.5. *Assumption 2.4 can be seen as a generalization of qualification constraints (Hare and Lewis, 2007, Sec. 1), and is usual in the machine learning literature (Zhao and Yu, 2006; Bach, 2008). Note that this assumption can be considered too conservative, and one can try to relax it using the mirror-stratifiable framework (Fadili et al., 2018, 2019).*

The contributions of this chapter are the following, first,

- When the nonsmooth penalty g is separable, we provide simple sufficient conditions for the optimized function to be partly smooth.

Then, under mild assumptions on the g_j functions, for the **cyclic** proximal coordinate descent algorithm:

- We prove finite time model identification (Theorem 2.13).
- We provide local linear convergence rates (Theorem 2.16).
- We illustrate our results on multiple real datasets and estimators (Section 2.4) showing that our theoretical rates match the empirical ones.

2.2.1 Partial smoothness for separable functions

The class of partly smooth functions was first defined in Lewis (2002). It encompasses a large number of known nonsmooth machine learning optimization penalties, such as the ℓ_1-norm or box constraints to only name a few, see Vaiter et al. (2018, Section 2.1) for details. Interestingly, this framework enables powerful theoretical tools on model identification such as Hare and Lewis (2004, Thm. 5.3). Loosely speaking, a partly smooth function behaves smoothly as it lies on the related model and sharply if we move normal to that model. Formally, we recall the definition of partly smooth functions restricted to the case of proper, lower semicontinuous and convex functions.

Definition 2.6 (Partial smoothness)**.** *Let $g : \mathbb{R}^p \to \mathbb{R}$ be a proper closed convex function. g is said to be partly smooth at β relative to a set $\mathcal{M} \subseteq \mathbb{R}^n$ if there exists a neighborhood $\mathcal{U}$ of β such that*

- ***(Smoothness)*** *$\mathcal{M} \cap \mathcal{U}$ is a $\mathcal{C}^2$-manifold and g restricted to $\mathcal{M} \cap \mathcal{U}$ is $\mathcal{C}^2$,*
- ***(Sharpness)*** *The tangent space of $\mathcal{M}$ at β is the model tangent space T_β where $T_\beta = \mathrm{Lin}(\partial g(\beta))^\perp$,*
- ***(Continuity)*** *The set valued mapping ∂g is continuous at β relative to $\mathcal{M}$.*

One of the key assumptions to solve Equation (2.1) using CD is that the nonsmooth function g is separable. In this setting, the set $\mathcal{M}$ appearing in Definition 2.6 is related to the notion of support that we now define.

Definition 2.7 (Generalized support, Nutini et al. 2019, Def. 1)**.** *For a vector $x \in \mathbb{R}^p$, its* generalized support $\mathcal{S}_\beta \subseteq [p]$ *is the set of indices $j \in [p]$ such that g_j is differentiable at β_j:*

$$\mathcal{S}_\beta \triangleq \{j \in [p] : \partial g_j(\beta_j) \text{ is a singleton}\} \ .$$

An iterative algorithm is said to achieve ***finite support identification*** *if its iterates $\beta^{(k)}$ converge to $\hat{\beta} \in \arg\min_{x \in \mathbb{R}^p} \Phi(x)$, and there exists $K \geq 0$ such that for all $j \notin \mathcal{S}_{\hat{\beta}}$, for all $k \geq K, \beta_j^{(k)} = \hat{\beta}_j$.*

This notion can be unified with the definition of model subspace from Vaiter et al. (2015, Sec. 3.1):

Definition 2.8 (Model subspace, Vaiter et al. 2015)**.** *We denote the model subspace at x:*

$$T_\beta = \{u \in \mathbb{R}^p : \forall j \in \mathcal{S}_\beta^c, u_j = 0\} \ , \tag{2.4}$$

where $\mathcal{S}_\beta$ is the generalized support of β (see Definition 2.7).

Example 2.9 (The ℓ_1 norm)**.** *The function $g(\beta) = \sum_{j=1}^{p} |\beta_j|$ is certainly the most popular nonsmooth convex regularizer promoting sparsity. Indeed, the ℓ_1 norm generates structured solution with model subspace (Vaiter et al., 2018). We have that $\mathcal{S}_\beta = \{j \in [p] : \beta_j \neq 0\}$ since $|\cdot|$ is differentiable everywhere but not at 0, and the model subspace reads:*

$$T_\beta = \{u \in \mathbb{R}^p : supp(u) \subseteq supp(\beta)\} \ . \tag{2.5}$$

The ℓ_1-norm is partly smooth at β relative to the set $\mathcal{M}_\beta = T_\beta$.

Example 2.10 (The box constraints indicator function $\iota_{[0,C]}$)**.** *This indicator function appears for instance in box constrained optimization problems such as the dual problem of the SVM. Let $\mathcal{I}_\beta^0 = \Big\{j \in [p] : \beta_j = 0\Big\}$ and $\mathcal{I}_\beta^C = \Big\{j \in [p] : \beta_j = C\Big\}$, then*

$$T_\beta = \{u \in \mathbb{R}^p : \mathcal{I}_\beta^0 \subseteq \mathcal{I}_u^0 \text{ and } \mathcal{I}_\beta^C \subseteq \mathcal{I}_u^0\}.$$

For the SVM, model identification boils down to finding the active set of the box constrained quadratic optimization problem after a finite number of iterations. The box indicator function of the interval $[0, C]$ is partly smooth at β relative to the set $\mathcal{M}_\beta = \beta + T_\beta$.

For separable functions, the next lemma gives an explicit link between the generalized support (Definition 2.7) and the framework of partly smooth functions (Hare and Lewis, 2004).

Lemma 2.11. *Let $\hat{\beta} \in \mathrm{dom}(g)$. If for every $j \in \mathcal{S}_{\hat{\beta}}$, g_j is locally C^2 around $\hat{\beta}_j$, then g is partly smooth at $\hat{\beta}$ relative to $\hat{\beta} + T_{\hat{\beta}}$.*

Proof We need to prove the three properties of the partial smoothness (Definition 2.6):

- *Smoothness.* Let us write $\mathcal{M}_{\hat{\beta}} = \hat{\beta} + T_{\hat{\beta}}$ the affine space directed by the model subspace and pointed by $\hat{\beta}$. In particular, it is a C^2-manifold.

  For every $j \in \mathcal{S}_{\hat{\beta}}$, g_j is locally C^2 around $\hat{\beta}_j$, hence there exists a neighborhood U_j of $\hat{\beta}_j$ such that the restriction of g_j to U is twice continuously differentiable. For $j \in \mathcal{S}_{\hat{\beta}}^c$, let's write $U_j = \mathbb{R}$. Take $U = \bigotimes_{j\in[p]} U_j$. This a neighborhood of $\hat{\beta}$ (it is open, and contains $\hat{\beta}$). Consider the restriction $g_{|\mathcal{M}_{\hat{\beta}}}$ of g to $\mathcal{M}_{\hat{\beta}}$. It is C^2 at each point of U since each coordinates (for $j \in \mathcal{S}_{\hat{\beta}}$) are C^2 around U_j.

- *Sharpness.* Since g is completely separable, we have that

  $$\partial g(\hat{\beta}) = \partial g_1(\hat{\beta}_1) \times \ldots \times \partial g_p(\hat{\beta}_p) \ . \tag{2.6}$$

  Note that $\partial g_j(\hat{\beta}_j)$ is a set valued mapping which is equal to the singleton $\{\nabla_j g(\hat{\beta}_j)\}$ if g_j is differentiable at $\hat{\beta}_j$ or it is equal to an interval. The model tangent space $T_{\hat{\beta}}$ of g at $\hat{\beta}$ is given by

  $$T_{\hat{\beta}} = \mathrm{span}(\partial g(\hat{\beta}))^\perp \quad \text{where} \quad \mathrm{span}(\partial g(\hat{\beta})) = \mathrm{aff}(\partial g(\hat{\beta})) - e_{\hat{\beta}} \ , \tag{2.7}$$

with

$$e_{\hat{\beta}} = \underset{e \in \mathrm{aff}(\partial g(\hat{\beta}))}{\arg\min} \|e\| \ , \tag{2.8}$$

called the model vector.

In the particular case of separable functions, we have that

$$\begin{aligned} \mathrm{aff}\left(\partial g(\hat{\beta})\right) &= \mathrm{aff}\left(\partial g_1(\hat{\beta}_1) \times \ldots \times \partial g_p(\hat{\beta}_p)\right) \\ &= \mathrm{aff}\left(\partial g_1(\hat{\beta}_1)\right) \times \ldots \times \mathrm{aff}\left(\partial g_p(\hat{\beta}_p)\right) \ . \end{aligned}$$

In this case,

$$\mathrm{aff}\left(\partial g_j(\hat{\beta}_j)\right) = \begin{cases} \{\nabla_j g(\hat{\beta}_j)\} & \text{if } j \in \mathcal{S}_{\hat{\beta}} \\ \mathbb{R} & \text{otherwise} \end{cases} \tag{2.9}$$

and

$$e_{\hat{\beta}_j} = \begin{cases} \nabla_j g(\hat{\beta}_j) & \text{if } j \in \mathcal{S}_{\hat{\beta}} \\ 0 & \text{otherwise} \ . \end{cases} \tag{2.10}$$

Thus we have that

$$\mathrm{span}\left(\partial g(\hat{\beta})\right) = \mathrm{aff}\left(\partial g(\hat{\beta})\right) - e_{\hat{\beta}} = \{x \in \mathbb{R}^p : \forall j' \in \mathcal{S}_{\hat{\beta}}, \beta_{j'} = 0\} \ .$$

Then

$$T_{\hat{\beta}} = \mathrm{span}\left(\partial g(\hat{\beta})\right)^{\perp} = \{x \in \mathbb{R}^p : \forall j' \in \mathcal{S}^c_{\hat{\beta}}, \beta_{j'} = 0\} \ . \tag{2.11}$$

- *Continuity.* We are going to prove that ∂g is inner semicontinuous at $\hat{\beta}$ relative to $\mathcal{M}_{\hat{\beta}}$, *i.e.,* that for any sequence $(\beta^{(k)})$ of elements of $\mathcal{M}_{\hat{\beta}}$ converging to $\hat{\beta}$ and any $\bar{\eta} \in \partial g(\hat{\beta})$, there exists a sequence of subgradients $\eta^{(k)} \in \partial g(\beta^{(k)})$ converging to $\bar{\eta}$.

  Let $\beta^{(k)}$ be a sequence of elements of $\mathcal{M}_{\hat{\beta}}$ converging to $\hat{\beta}$, or equivalently, let $t^{(k)}$ be a sequence of elements of $T_{\hat{\beta}}$ converging to 0, and let $\bar{\eta} \in \partial g(\hat{\beta})$.

  For $j \in \mathcal{S}_{\hat{\beta}}$, we choose $\eta_j^{(k)} \triangleq g_j'(\hat{\beta}_j + t_j^{(k)})$, using the smoothness property we have $\eta_j^{(k)} \triangleq \bar{\eta}_j$. For all $j \in \mathcal{S}^c_{\hat{\beta}}$ $\beta_j^{(k)} = \hat{\beta}_j$ we choose $\eta_j^{(k)} \triangleq \bar{\eta}_j$, since $\beta^{(k)} \in \mathcal{M}_{\hat{\beta}}$, we have $\eta_j^{(k)} \in \partial g(\beta^{(k)})$.

  We have that $\eta^{(k)} \in \partial g(\beta^k)$ and $\eta^{(k)}$ converges towards $\bar{\eta}$ since g_j' is C^1 around $\hat{\beta}_j$ for $j \in \mathcal{S}_{\hat{\beta}}$, hence, $g_j'(\hat{\beta}_j + t_j^{(k)})$ converges to $g_j'(\hat{\beta}_j) = \bar{\eta}_j$. Thus, it proves that g is partly smooth at $\hat{\beta}$ relative to $\hat{\beta} + T_{\hat{\beta}}$.

■

2.2.2 Identification

In this section we show the generalized support identification result of cyclic coordinate descent. To ensure model identification, we need the following (mild) assumption:

Assumption 2.12 (Locally $\mathcal{C}^2$)**.** *For all $j \in \mathcal{S}_{\hat{\beta}}$, g_j is locally $\mathcal{C}^2$ around $\hat{\beta}_j$, and f is locally $\mathcal{C}^2$ around $\hat{\beta}$.*

It is satisfied for the Lasso and the dual SVM problem mentioned above, but also for sparse logistic regression and elastic net. The following theorem shows that the CD (Algorithm 2.1) has the model identification property with local constant step size $0 < \gamma_j \leq 1/L_j$:

Theorem 2.13 (Model identification of CD)**.** *Consider a solution $\hat{\beta} \in \arg\min_{\beta \in \mathbb{R}^p} \Phi(\beta)$ and $\mathcal{S} = \mathcal{S}_{\hat{\beta}}$. Suppose*

1. *Assumptions 2.1 to 2.4 and 2.12 hold,*
2. *The sequence $(\beta^{(k)})_{k \geq 0}$ generated by Algorithm 2.1 converges to $\hat{\beta}$.*

Then, Algorithm 2.1 identifies the model after a finite number of iterations, which means that there exists $K > 0$ such that for all $k \geq K$, $\beta^{(k)}_{\mathcal{S}^c} = \hat{\beta}_{\mathcal{S}^c}$.

This result implies that for k large enough, $\beta^{(k)}$ shares the support of $\hat{\beta}$ (potentially smaller).

Proof Thanks to Lemma 2.11, we have that g is *partly smooth* (Lewis, 2002) at $\hat{\beta}$ relative to the affine space $\hat{\beta} + T_{\hat{\beta}}$.

We now prove that for the CD Algorithm 2.1: $\operatorname{dist}\left(\partial\Phi(\beta^{(k)}), 0\right) \to 0$, when $k \to \infty$.

As written in Algorithm 2.1, one update of coordinate descent reads:

$$\begin{aligned} \frac{1}{\gamma_j}\beta_j^{(j-1,k)} - \nabla_j f\left(\beta^{(j-1,k)}\right) - \frac{1}{\gamma_j}\beta_j^{(j,k)} &\in \partial g_j\left(\beta_j^{(j,k)}\right) \\ \frac{1}{\gamma_j}\beta_j^{(k)} - \nabla_j f\left(\beta^{(j-1,k)}\right) - \frac{1}{\gamma_j}\beta_j^{(k+1)} &\in \partial g_j\left(\beta_j^{(k+1)}\right) \quad . \end{aligned}$$

Since g is separable with non-empty subdifferential, the coordinatewise subdifferential of g is equal to the subdifferential of g, we then have

$$\frac{1}{\gamma} \odot \beta^{(k)} - \left(\nabla_j f\left(\beta^{(j-1,k)}\right)\right)_{j \in [p]} - \frac{1}{\gamma} \odot \beta^{(k+1)} \in \partial g(\beta^{(k+1)}) \ , \tag{2.12}$$

which leads to

$$\frac{1}{\gamma} \odot \beta^{(k)} - \left(\nabla_j f\left(\beta^{(j-1,k)}\right)\right)_{j \in [p]} - \frac{1}{\gamma} \odot \beta^{(k+1)} + \nabla f(\beta^{(k+1)}) \in \partial\Phi(\beta^{(k+1)}) \ . \tag{2.13}$$

To prove support identification using Hare and Lewis (2004, Thm. 5.3), we need to bound the distance between $\partial\Phi(\beta^{(k+1)})$ and 0, using Equation (2.13):

$$\begin{aligned}
\operatorname{dist}\left(\partial\Phi(\beta^{(k+1)}), 0\right)^2 &\leq \sum_{j=1}^{p} \left| \frac{\beta_j^{(k)}}{\gamma_j} - \nabla_j f(\beta^{(j-1,k)}) - \frac{\beta_j^{(k+1)}}{\gamma_j} + \nabla_j f(\beta^{(k+1)}) \right|^2 \\
&\leq \|\beta^{(k)} - \beta^{(k+1)}\|_{\gamma^{-1}}^2 + \sum_{j=1}^{p} \left| \nabla_j f\left(\beta^{(j-1,k)}\right) - \nabla_j f\left(\beta^{(k+1)}\right) \right|^2 \\
&\leq \|\beta^{(k)} - \beta^{(k+1)}\|_{\gamma^{-1}}^2 + L^2 \sum_{j=1}^{p} \|\beta^{(j-1,k)} - \beta^{(k+1)}\|^2 \\
&\leq \underbrace{\|\beta^{(k)} - \beta^{(k+1)}\|_{\gamma^{-1}}^2 + L^2 \sum_{j=1}^{p} \sum_{j'\geq j}^{p} \left| \beta_{j'}^{(k)} - \beta_{j'}^{(k+1)} \right|^2}_{\to 0 \text{ when } k\to\infty} .
\end{aligned}$$

We thus have:

- $\operatorname{dist}\left(\partial\Phi(\beta^{(k+1)}), 0\right) \to 0$
- $\Phi(\beta^{(k)}) \to \Phi(\hat{\beta})$ because Φ is prox-regular (since it is convex, see Poliquin and Rockafellar 1996a) and subdifferentially continuous.

Then the conditions to apply Hare and Lewis (2004, Th. 5.3) are met and hence we have model identification after a finite number of iterations. ■

Remark 2.14 (Link with existing literature)**.** *Regarding generic model identification results for CD Algorithm 2.1, Nutini et al. (2017, Lemma 3) shows model identification of coordinate descent, but relies on (Nutini et al., 2019, Lemma 1), assuming the strong convexity of the smooth function f. This is a strong hypothesis in the sparse setting, usually not realistic. Nutini (2018, Sec. 6.2.2, Thm. 9) shows model identification of coordinate descent with step size $1/L$, without the strong convexity of f. We proposed a similar results for CD with step size $1/L_j$, relying a different proof technique: the proof of Nutini (2018, Sec. 6.2.2, Thm. 9) is based on an explicit manipulation of the expression of the minimum distance to the boundary of the subdifferential, whereas our proof is based on a geometrical statement showing that any separable function regular enough enjoys a natural control on this distance, thanks to partial smoothness theory.*

2.3 Local linear convergence

In this section, we prove the local linear convergence of the CD Algorithm 2.1. After model identification, there exists a regime where the convergence towards a solution of Equation (2.1) is linear. Local linear convergence was already proved in various settings such as for ISTA and FISTA algorithms (*i.e.,* with an ℓ_1 penalty, Tao et al. 2016) and then for the general Forward-Backward algorithm (Liang et al., 2014).

Local linear convergence requires an additional assumption: *restricted injectivity*. It is classical for this type of analysis as it can be found in Liang et al. (2017) and Poon and Liang (2019).

Assumption 2.15. *(Restricted injectivity) For a solution* $\hat{\beta} \in \arg\min_{\beta \in \mathbb{R}^p} \Phi(\beta)$*, the restricted Hessian to its generalized support* $\mathcal{S} = \mathcal{S}_{\hat{\beta}}$ *is definite positive,* i.e.,

$$\nabla^2_{\mathcal{S},\mathcal{S}} f(\hat{\beta}) \succ 0 \ . \tag{2.14}$$

In order to study local linear convergence, we consider the fixed-point iteration of a complete epoch (an epoch is a complete pass over all the coordinates). Thanks to model identification (Theorem 2.13), we are able to prove that once the generalized support is correctly identified, there exists a regime where CD algorithm converges linearly towards the solution.

Theorem 2.16 (Local linear convergence)**.** *Consider a solution* $\hat{\beta} \in \arg\min_{\beta \in \mathbb{R}^p} \Phi(\beta)$ *and* $\mathcal{S} = \mathcal{S}_{\hat{\beta}}$*. Suppose*

1. *Assumptions 2.1 to 2.4, 2.12 and 2.15 hold.*
2. *The sequence* $(\beta^{(k)})_{k \geq 0}$ *generated by Algorithm 2.1 converges to* $\hat{\beta}$*.*
3. *The model has been identified* i.e., *there exists* $K \geq 0$ *such as for all* $k \geq K$

$$\beta^{(k)}_{\mathcal{S}^c} = \hat{\beta}_{\mathcal{S}^c} \ .$$

Then $(\beta^{(k)})_{k \geq K}$ *converges locally linearly towards* $\hat{\beta}$*. More precisely, once the model has been identified, there exists* $\psi : \mathbb{R}^{|\mathcal{S}|} \to \mathbb{R}^{|\mathcal{S}|}$ *(defined in Equation (2.18)) such that:*

$$\beta^{(k+1)}_{\mathcal{S}} = \psi(\beta^{(k)}_{\mathcal{S}}) \ ,$$

and for any $\nu \in [\rho(\mathcal{J}\psi(\hat{\beta}_{\mathcal{S}})), 1[$*, there exists* $K > 0$ *and* $C > 0$ *such that for all* $k \geq K$*,*

$$\|\beta^{(k)}_{\mathcal{S}} - \hat{\beta}_{\mathcal{S}}\| \leq C\nu^{(k-K)} \|\beta^{(K)}_{\mathcal{S}} - \hat{\beta}_{\mathcal{S}}\| \ .$$

Proof To simplify the notations we write $\mathcal{S} \triangleq \mathcal{S}_{\hat{\beta}}$, and its elements as follows: $\mathcal{S} = \{j_1, \ldots, j_{|\mathcal{S}|}\}$. We also define $\pi : \mathbb{R}^{|\mathcal{S}|} \to \mathbb{R}^p$ for all $\beta_{\mathcal{S}} \in \mathbb{R}^{|\mathcal{S}|}$ and all $j \in \mathcal{S}$ by

$$\pi(\beta_{\mathcal{S}})_j = \begin{cases} \beta_j & \text{if } j \in \mathcal{S} \\ \hat{\beta}_j & \text{if } j \in \mathcal{S}^c \ , \end{cases} \tag{2.15}$$

and for all $s \in [|\mathcal{S}|]$, $\mathcal{P}^{(s)} : \mathbb{R}^{|\mathcal{S}|} \to \mathbb{R}^{|\mathcal{S}|}$ is defined for all $u \in \mathbb{R}^{|\mathcal{S}|}$ and all $s' \in [|S|]$ by

$$\left(\mathcal{P}^{(s)}(u)\right)_{s'} = \begin{cases} u_{s'} & \text{if } s \neq s' \\ \operatorname{prox}_{\gamma_{j_s} g_{j_s}}\left(u_s - \gamma_{j_s} \nabla_{j_s} f(\pi(u))\right) & \text{if } s = s' \ . \end{cases} \tag{2.16}$$

Once the model is identified (Theorem 2.13), we have that there exists $K \geq 0$ such that for all $k \geq K$

$$\beta^{(k)}_{\mathcal{S}^c} = \hat{\beta}_{\mathcal{S}^c} \quad \text{and} \tag{2.17}$$

$$\beta^{(k+1)}_{\mathcal{S}} = \psi(\beta^{(k)}_{\mathcal{S}}) \triangleq \mathcal{P}^{(|\mathcal{S}|)} \circ \ldots \circ \mathcal{P}^{(1)}(\beta^{(k)}_{\mathcal{S}}) \ . \tag{2.18}$$

The proof of Theorem 2.16 follows the same steps as the ones of Proposition 1.6 on least squares in Chapter 1:

- First we show that the fixed-point operator ψ is differentiable at $\hat{\beta}_{\mathcal{S}}$ (Lemma 2.17).
- Then we show that the Jacobian spectral radius of ψ is strictly bounded by one (Lemma 2.18 (e)). Proof of Lemma 2.18 (e) relies on Lemmas 2.18 (a) to 2.18 (d).
- Finally we conclude to local linear convergence by applying Polyak (1987, Theorem 1, Section 2.1.2).

The following lemma shows that ψ is differentiable at the optimum.

Lemma 2.17 (Differentiability fixed-point operator)**.** *ψ is differentiable at $\hat{\beta}_{\mathcal{S}}$ with Jacobian:*

$$\mathcal{J}\psi(\hat{\beta}_{\mathcal{S}}) = M^{-1/2} \underbrace{(\mathrm{Id} - B^{(|\mathcal{S}|)}) \dots (\mathrm{Id} - B^{(1)})}_{\triangleq A} M^{1/2} \ , \tag{2.19}$$

$$M \triangleq \nabla^2_{\mathcal{S},\mathcal{S}} f(\hat{\beta}) + \mathrm{diag}\,(u) \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|} \ , \tag{2.20}$$

$u \in \mathbb{R}^{|\mathcal{S}|}$ is defined for all $s \in [|\mathcal{S}|]$ by

$$u_s = \begin{cases} \frac{1}{\gamma_{j_s} \partial_x \operatorname{prox}_{\gamma_{j_s} g_{j_s}}(\hat{z}_{j_s})} - \frac{1}{\gamma_{j_s}} & \text{if } \operatorname{prox}_{\gamma_{j_s} g_{j_s}}(\hat{z}_{j_s}) \neq 0 \\ 0 & \text{otherwise,} \end{cases} \tag{2.21}$$

$$B^{(s)} = M^{1/2}_{:s} \gamma_{j_s} \partial_x \operatorname{prox}_{\gamma_{j_s} g_{j_s}} \left(\hat{z}_{j_s} \right) M^{1/2\top}_{:s} \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}, \tag{2.22}$$

and $\hat{z} = \hat{\beta} - \gamma \odot \nabla f(\hat{\beta}) \in \mathbb{R}^p$.

Proof (Lemma 2.17). From Assumption 2.12, we know there exists a neighborhood of $\hat{\beta}_j$ denoted $\mathcal{U}$ such that, for $j \in \mathcal{S}$, the restriction of g_j to $\mathcal{U}$ is $\mathcal{C}^2$ on $\mathcal{U}$. In particular, it means that $\hat{\beta}_j$ is a differentiable point of g_j and given a pair $(u, v) \in \mathcal{U} \times \mathbb{R}^p$ such that

$$u = \operatorname{prox}_{\gamma_j g_j}(v) \in \mathcal{U} \ , \tag{2.23}$$

we have $\frac{1}{\gamma_j}(v - u) \in \partial g_j(u)$ becomes

$$\frac{1}{\gamma_j}(v - u) = g_j'(u) \Leftrightarrow v = u + \gamma_j g_j'(u) \Leftrightarrow v = (\mathrm{Id} + g_j')(u) \ . \tag{2.24}$$

Let $H(u) = (\mathrm{Id} + g_j')(u)$, since g_j is twice differentiable at u, we have that

$$H'(u) = 1 + \gamma_j g_j''(u) \ . \tag{2.25}$$

Thus, $H' : \mathcal{U} \mapsto \mathbb{R}$ is continuous and then $H : \mathcal{U} \mapsto \mathbb{R}$ is continuously differentiable. Hence $F(v, u) \triangleq v - H(u)$ is $\mathcal{C}^1$ and $F(v, u) = 0$. By convexity of g, we have $g_j''(u) \geq 0$ and

$$\frac{\partial F}{\partial u}(v, u) = -H'(u) = -1 - \gamma_j g''(u) \neq 0 \ . \tag{2.26}$$

Using the implicit functions theorem, we have that there exists an open interval $\mathcal{V} \subseteq \mathbb{R}$ with $v \in \mathcal{V}$ and a function $h : \mathcal{V} \mapsto \mathbb{R}$ which is $\mathcal{C}^1$ such as $u = h(v)$.

Using (2.23) we thus have with the choice $u = \hat{\beta}_j$, $v = \hat{\beta}_j - \gamma_j \nabla_j f(\hat{\beta})$ that the map h coincides with $\text{prox}_{\gamma_j g_j}$ on $\mathcal{V}$ and is differentiable at $v = \hat{\beta}_j - \gamma_j \nabla_j f(\hat{\beta}) \in \mathcal{V}$. It follows that $\mathcal{P}^{(s)}$ is differentiable at $\hat{\beta}_{\mathcal{S}}$.

For all $s \in [|\mathcal{S}|]$, $\mathcal{P}^{(s)}$ is differentiable at $\hat{\beta}_{\mathcal{S}}$. In addition, $\mathcal{P}^{(s)}(\hat{\beta}_{\mathcal{S}}) = \hat{\beta}_{\mathcal{S}}$, thus

$$\psi \triangleq \mathcal{P}^{(|\mathcal{S}|)} \circ \ldots \circ \mathcal{P}^{(1)} \ , \tag{2.27}$$

is also differentiable at $\hat{\beta}_{\mathcal{S}}$. To compute, the Jacobian of $\mathcal{P}^{(s)}$ at $\hat{\beta}_{\mathcal{S}}$, let us first notice that

$$\mathcal{J}\mathcal{P}^{(s)}(\hat{\beta}_{\mathcal{S}})^\top = \left(\ e_1 \ \middle| \ \ldots \ \middle| \ e_{s-1} \ \middle| \ v_s \ \middle| \ e_{s+1} \ \middle| \ \ldots \ \middle| \ e_{|\mathcal{S}|} \ \right) \ , \tag{2.28}$$

where $v_s = \partial_x \text{prox}_{\gamma_{j_s} g_{j_s}} \left(\hat{z}_{j_s} \right) \left(e_{j_s} - \gamma_{j_s} \nabla^2_{j_s,:} f(\hat{\beta}) \right)$ and $\hat{z}_j = \hat{\beta}_j - \gamma_j \nabla_j f(\hat{\beta})$. This matrix can be rewritten as

$$\begin{aligned}
\mathcal{J}\mathcal{P}^{(s)}(\hat{\beta}_{\mathcal{S}}) &= \text{Id}_{|S|} - e_s e_s^\top + \partial_x \text{prox}_{\gamma_{j_s} g_{j_s}} \left(\hat{z}_{j_s} \right) \left(e_s e_s^\top - \gamma_{j_s} e_s e_s^\top \nabla^2 f(\hat{\beta}) \right) \\
&= \text{Id}_{|S|} - e_s e_s^\top \gamma_{j_s} \partial_x \text{prox}_{\gamma_{j_s} g_{j_s}} \left(\hat{z}_{j_s} \right) \left(\text{diag}(u) + \nabla^2 f(\hat{\beta}) \right) \\
&= \text{Id}_{|S|} - e_s e_s^\top \gamma_{j_s} \partial_x \text{prox}_{\gamma_{j_s} g_{j_s}} \left(\hat{z}_{j_s} \right) M \\
&= M^{-1/2} \left(\text{Id}_{|S|} - M^{1/2} e_s e_s^\top \gamma_{j_s} \partial_x \text{prox}_{\gamma_{j_s} g_{j_s}} \left(\hat{z}_{j_s} \right) M^{1/2} \right) M^{1/2} \\
&= M^{-1/2} \left(\text{Id}_{|S|} - B^{(s)} \right) M^{1/2} \ ,
\end{aligned} \tag{2.29}$$

where

$$M \triangleq \nabla^2_{\mathcal{S},\mathcal{S}} f(\hat{\beta}) + \text{diag}\,(u) \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|} \ , \tag{2.30}$$

and $u \in \mathbb{R}^{|\mathcal{S}|}$ is defined for all $s \in [|\mathcal{S}|]$ by

$$u_s = \begin{cases} \frac{1}{\gamma_{j_s} \partial_x \text{prox}_{\gamma_{j_s} g_{j_s}}(\hat{z}_{j_s})} - \frac{1}{\gamma_{j_s}} & \text{if } \text{prox}_{\gamma_{j_s} g_{j_s}}(\hat{z}_{j_s}) \neq 0 \\ 0 & \text{otherwise,} \end{cases} \tag{2.31}$$

and

$$B^{(s)} \triangleq M^{1/2}_{:s} \gamma_{j_s} \partial_x \text{prox}_{\gamma_{j_s} g_{j_s}} \left(\hat{z}_{j_s} \right) M^{1/2 \top}_{:s} \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|}. \tag{2.32}$$

The chain rule leads to

$$\begin{aligned}
\mathcal{J}\psi(\hat{\beta}_{\mathcal{S}}) &= \mathcal{J}\mathcal{P}^{(|\mathcal{S}|)}(\hat{\beta}_{\mathcal{S}}) \mathcal{J}\mathcal{P}^{(|\mathcal{S}|-1)}(\hat{\beta}_{\mathcal{S}}) \ldots \mathcal{J}\mathcal{P}^{(1)}(\hat{\beta}_{\mathcal{S}}) \\
&= M^{-1/2} \underbrace{(\text{Id} - B^{(|\mathcal{S}|)})(\text{Id} - B^{(|\mathcal{S}|-1)}) \ldots (\text{Id} - B^{(1)})}_{\triangleq A} M^{1/2} \ .
\end{aligned}$$

■

The next series of lemmas (Lemmas 2.18 (a) to 2.18 (d)) will be useful to bound the spectral radius of the Jacobian of the fixed-point operator ψ (Lemma 2.18 (e)).

Lemma 2.18. *a) The matrix M defined in (2.20) is symmetric definite positive.*

b) For all $s \in [|\mathcal{S}|]$, the spectral radius of the matrix $B^{(s)}$ defined in (2.22) is bounded by 1, i.e., $\|B^{(s)}\|_2 \leq 1$.

c) For all $s \in [|\mathcal{S}|]$, $B^{(s)}/\|B^{(s)}\|$ is an orthogonal projector onto $\mathrm{Span}(M_{:s}^{1/2})$.

d) For all $s \in [|\mathcal{S}|]$ and for all $u \in \mathbb{R}^S$, if $\|(\mathrm{Id} - B^{(s)})u\| = \|u\|$ then $u \in \mathrm{Span}(M_{:s}^{1/2})^\perp$.

e) The spectral radius of the Jacobian $\mathcal{J}\psi(\hat{\beta}_\mathcal{S})$ of the fixed-point operator ψ is bounded by 1

$$\rho(\mathcal{J}\psi(\hat{\beta}_\mathcal{S})) < 1 \ . \tag{2.33}$$

Proof (Lemma 2.18 (a)). Using the non-expansivity of the proximal operator, and the property $\partial_x \operatorname{prox}_{\gamma_j g_j}(\hat{z}_j) > 0$ for $j \in \mathcal{S}$, $\mathrm{diag}(u)$ is a symmetric semidefinite matrix, so M is a sum of a symmetric definite positive matrix and a symmetric semidefinite matrix, hence M is symmetric definite positive. ■

Proof (Lemma 2.18 (b)). $B^{(s)}$ is a rank one matrix which is the product of $\gamma_{j_s}\partial_x \operatorname{prox}_{\gamma_{j_s} g_j}(\hat{z}_j) M_{:s}^{1/2}$ and $M_{:s}^{1/2\top}$, its non-zeros eigenvalue is thus given by

$$\begin{aligned}
\|B^{(s)}\|_2 &= \left| M_{:s}^{1/2\top} \gamma_{j_s} \partial_x \operatorname{prox}_{\gamma_{j_s} g_{j_s}}(\hat{z}_j)_s M_{:s}^{1/2} \right| \\
&= \left| \gamma_{j_s} \partial_x \operatorname{prox}_{\gamma_{j_s} g_{j_s}}(\hat{z}_{j_s}) M_{s,s} \right| \\
&= \left| \gamma_{j_s} \partial_x \operatorname{prox}_{\gamma_{j_s} g_{j_s}}(\hat{z}_{j_s}) \left(\underbrace{\nabla^2_{j,j} f(\hat{\beta})}_{0\leq} + \underbrace{\left(\frac{1}{\gamma_{j_s} \partial_x \operatorname{prox}_{\gamma_{j_s} g_{j_s}}(\hat{z}_{j_s})} - \frac{1}{\gamma_{j_s}} \right)}_{0\leq} \right) \right| .
\end{aligned} \tag{2.34}$$

By positivity of the two terms,

$$\begin{aligned}
\|B^{(s)}\|_2 &= \gamma_{j_s} \partial_x \operatorname{prox}_{\gamma_{j_s} g_{j_s}}(\hat{z}_{j_s}) \underbrace{\nabla^2_{j,j} f(\hat{\beta})}_{\leq L_j \leq \frac{1}{\gamma_{j_s}}} + \left(1 - \partial_x \operatorname{prox}_{\gamma_{j_s} g_{j_s}}(\hat{z}_{j_s}) \right) \\
&\leq \partial_x \operatorname{prox}_{\gamma_{j_s} g_{j_s}}(\hat{z}_{j_s}) + \left(1 - \partial_x \operatorname{prox}_{\gamma_{j_s} g_{j_s}}(\hat{z}_{j_s}) \right) \\
&\leq 1 \ .
\end{aligned} \tag{2.35}$$

■

Proof (Lemma 2.18 (c)). It is clear that $B^{(s)}/\|B^{(s)}\|$ is symmetric. We now prove that it is idempotent, *i.e.*, $(B^{(s)}/\|B^{(s)}\|)^2 = B^{(s)}/\|B^{(s)}\|$.

$$\begin{aligned}
(B^{(s)}/\|B^{(s)}\|)^2 &= (\gamma_j \partial_x \operatorname{prox}_{\gamma_j g_j}(\hat{z}_j))^2 M_{:s}^{1/2} M_{:s}^{1/2\top} M_{:s}^{1/2} M_{:s}^{1/2\top} / \|B^{(s)}\|^2 \\
&= (\gamma_j \partial_x \operatorname{prox}_{\gamma_j g_j}(\hat{z}_j)) \|B^{(s)}\| M_{:s}^{1/2} M_{:s}^{1/2\top} / \|B^{(s)}\|^2 \\
&= B^{(s)}/\|B^{(s)}\| \ .
\end{aligned}$$

Hence, $B^{(s)}/\|B^{(s)}\|$ is an orthogonal projector. ■

Proof (Lemma 2.18 (d)).

$$\begin{aligned}
\mathrm{Id} - B^{(s)} &= \mathrm{Id} - \|B^{(s)}\| \frac{B^{(s)}}{\|B^{(s)}\|} \\
&= (1 - \|B^{(s)}\|)\,\mathrm{Id} + \|B^{(s)}\|_2\,\mathrm{Id} - \|B^{(s)}\|_2 \frac{B^{(s)}}{\|B^{(s)}\|_2} \\
&= (1 - \|B^{(s)}\|)\,\mathrm{Id} + \|B^{(s)}\| \underbrace{\left(\mathrm{Id} - \frac{B^{(s)}}{\|B^{(s)}\|_2}\right)}_{\text{projection onto } M_{:s}^{1/2\perp}} . \qquad (2.36)
\end{aligned}$$

Let $u \notin \mathrm{Span}(M_{:s}^{1/2})^\perp$, then there exists $\alpha \neq 0$, $u_{M_{:s}^{1/2\perp}} \in \mathrm{Span}(M_{:s}^{1/2})^\perp$ such that

$$u = \alpha M_{:s} + u_{M_{:s}^{1/2\perp}} . \qquad (2.37)$$

Combining Equations (2.36) and (2.37) leads to:

$$\begin{aligned}
(\mathrm{Id} - B^{(s)})u &= (1 - \|B^{(s)}\|_2)u + \|B^{(s)}\|_2 u_{M_{:s}^{1/2\perp}} \\
\|(\mathrm{Id} - B^{(s)})u\| &\leq \underbrace{|1 - \|B^{(s)}\|_2|}_{=(1-\|B^{(s)}\|_2)} \|u\| + \|B^{(s)}\|_2 \underbrace{\|u_{M_{:s}^{1/2\perp}}\|}_{<\|u\|} \\
&< \|u\| .
\end{aligned}$$

■

Proof (Lemma 2.18 (e)). Let $u \in \mathbb{R}^s$ such that

$$\|(\mathrm{Id}_{|\mathcal{S}|} - B^{(|\mathcal{S}|)}) \dots (\mathrm{Id}_{|\mathcal{S}|} - B^{(1)})u\| = \|u\| . \qquad (2.38)$$

Since

$$\|(\mathrm{Id}_{|\mathcal{S}|} - B^{(|\mathcal{S}|)} \dots (\mathrm{Id}_{|\mathcal{S}|} - B^{(1)})\|_2 \leq \underbrace{\|(\mathrm{Id}_{|\mathcal{S}|} - B^{(|\mathcal{S}|)})\|_2}_{\leq 1} \times \dots \times \underbrace{\|(\mathrm{Id}_{|\mathcal{S}|} - B^{(1)})\|_2}_{\leq 1} ,$$

we thus have for all $j \in \mathcal{S}$, $\|(\mathrm{Id}_{|\mathcal{S}|} - B^{(s)})u\| = \|u\|$. One can thus successively apply Lemma 2.18 (d) which leads to:

$$u \in \bigcap_{s \in [|\mathcal{S}|]} \mathrm{Span}\, M_{:s}^{1/2\perp} \Leftrightarrow u \in \mathrm{Span}\left(M_{:1}^{1/2}, \dots, M_{:|\mathcal{S}|}^{1/2}\right)^\perp .$$

Moreover $M^{1/2}$ has full rank (see Lemma 2.18 (a)), thus $u = 0$ and

$$\|(\mathrm{Id}_{|\mathcal{S}|} - B^{(|\mathcal{S}|)}) \dots (\mathrm{Id}_{|\mathcal{S}|} - B^{(1)})\|_2 < 1 .$$

From Lemma 2.18 (e), $\|A\|_2 < 1$. Moreover A and $\mathcal{J}\psi(\hat{\beta}_{\mathcal{S}})$ are similar matrices (Equation (2.19)), then $\rho(\mathcal{J}\psi(\hat{\beta}_{\mathcal{S}})) = \rho(A) < 1$. ■

Then all conditions (Lemmas 2.18 (e) and 2.17) are met to apply Polyak (1987, Theorem 1, Section 2.1.2) which proves the local linear convergence. ■

2.3.1 Differentiability outside the generalized support

For the sake of completeness, we also show that $\text{prox}_{\gamma_j g_j}$ is differentiable on the complement of the generalized support at $\hat{\beta}_j - \nabla_j f(\hat{\beta})$. This lemma will be useful in Chapter 6 to derive an implicit differentiation formula.

Lemma 2.19. *Consider a solution* $\hat{\beta} \in \arg\min_{\beta \in \mathbb{R}^p} \Phi(\beta)$ *and* $\mathcal{S} = \mathcal{S}_{\hat{\beta}}$. *Suppose Assumptions 2.1 to 2.4, 2.12 and 2.15 hold.*

Then, for all $j \in \mathcal{S}^c$, $\text{prox}_{\gamma_j g_j}$ *is constant around* $\hat{\beta}_j - \nabla_j f(\hat{\beta})$. *Moreover, the map* $\beta \mapsto \text{prox}_{\gamma_j g_j}(\beta_j - \nabla_j f(\beta))$ *is differentiable at* $\hat{\beta}$ *with gradient* 0.

Proof Let $\partial g_j(\hat{\beta}_j) = [a; b]$ and let $\hat{z}_j = \hat{\beta}_j - \nabla_j f(\hat{\beta})$, then combining the fixed point equation and Assumption 2.4 leads to:

$$\frac{1}{\gamma_j}(\hat{z}_j - \hat{\beta}_j) \in \text{ri}\left(\partial g_j(\hat{\beta}_j)\right) =]a; b[\ . \tag{2.39}$$

Thus,

$$\hat{z}_j \in]\gamma_j a + \hat{\beta}_j; \gamma_j b + \hat{\beta}_j[\ . \tag{2.40}$$

For all $v \in]\gamma_j a + \hat{\beta}_j; \gamma_j b + \hat{\beta}_j[$, we have $\frac{1}{\gamma_j}(v - \hat{\beta}_j) \in]a; b[= \text{ri}\left(\partial g_j(\hat{\beta}_j)\right)$, *i.e.,* $\text{prox}_{\gamma_j g_j}(v) = \hat{\beta}_j$. As f is $\mathcal{C}^2$ in $\hat{\beta}$, we have that $\beta \mapsto \text{prox}_{\gamma_j g_j}(\beta_j - \nabla_j f(\beta))$ is differentiable at $\hat{\beta}$ with gradient being 0. ■

2.4 Experiments

We now illustrate Theorems 2.13 and 2.16 on multiple datasets and estimators: the Lasso, the logistic regression and the SVM. In this section, we consider a design matrix $X \in \mathbb{R}^{n \times p}$ and a target $y \in \mathbb{R}^n$ for regression (Lasso) and $y \in \{-1, 1\}^n$ for classification (logistic regression and support-vector machine). We used classical datasets from `libsvm` (Chang and Lin, 2011) summarized in Table 2.1.

In Figures 2.1 to 2.3 the distance of the iterates to the optimum, $\|\beta^{(k)} - \hat{\beta}\|$ as a function of the number of iterations k is plotted as a solid blue line. The vertical red dashed line represents the iteration $\hat{k}$ where the model has been identified by CD (Algorithm 2.1) illustrating Theorem 2.13. The yellow dashed line represents the theoretical linear rate from Theorem 2.16. Theorem 2.16 gives the slope of the dashed yellow line, the (arbitrary) origin point of the theoretical rate line is chosen such that blue and yellow lines coincide at identification time, *i.e.,* all lines intersect at this point. More precisely, if $\hat{k}$ denotes the iteration where model identification happens, the equation of the dashed yellow line is:

$$h(k) = \|\beta^{(\hat{k})} - \hat{\beta}\| \times \rho(\mathcal{J}\psi(\hat{\beta}_{\mathcal{S}}))^{(k-\hat{k})} \ . \tag{2.41}$$

Once a solution $\hat{\beta}$ has been computed, one can calculate $\mathcal{J}\psi(\hat{\beta}_{\mathcal{S}})$ and its spectral radius for each estimator.

For the experiments we used three different estimators that we detail here.

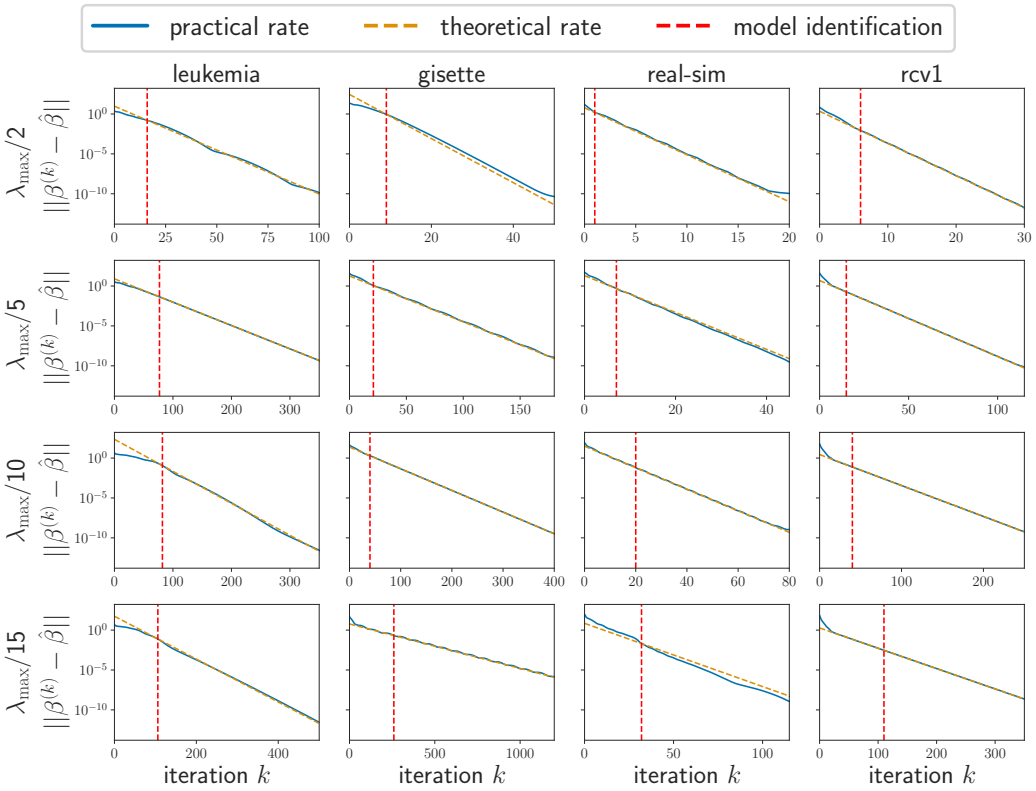


Figure 2.1 – **Lasso, linear convergence.** Distance to optimum, $\|\beta^{(k)} - \hat{\beta}\|$, as a function of the number of iterations k, on 4 different datasets: *leukemia*, *gisette*, *rcv1*, and *real-sim*.

Table 2.1 – Characteristics of the datasets.

Datasets	#samples n	#features p	density
leukemia	38	7129	1
gisette	6000	4955	1
rcv1	20 242	19 959	3.6×10^{-3}
real-sim	72 309	20 958	2.4×10^{-3}
20news	5184	155 148	1.9×10^{-3}

Lasso. (Tibshirani, 1996) The most famous estimator based on a nonsmooth optimization problem may be the Lasso. For a design matrix $X \in \mathbb{R}^{n \times p}$ and a target $y \in \mathbb{R}^n$ it writes:

$$\arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2n} \|X\beta - y\|^2 + \lambda \|\beta\|_1 \ . \tag{2.42}$$

The CD update for the Lasso is given by

$$\beta_j \leftarrow \mathrm{ST}_{\gamma_j \lambda} \left(\beta_j - \gamma_j X_{:j}^\top (y - X\beta) \right) \ , \tag{2.43}$$

where $\mathrm{ST}_\lambda(\beta) = \mathrm{sign}(\beta) \cdot \max(|\beta| - \lambda, 0)$. The solution of Equation (2.42) is obtained using Algorithm 2.1 with constant stepsizes $1/\gamma_j = \frac{\|X_{:j}\|^2}{n}$.

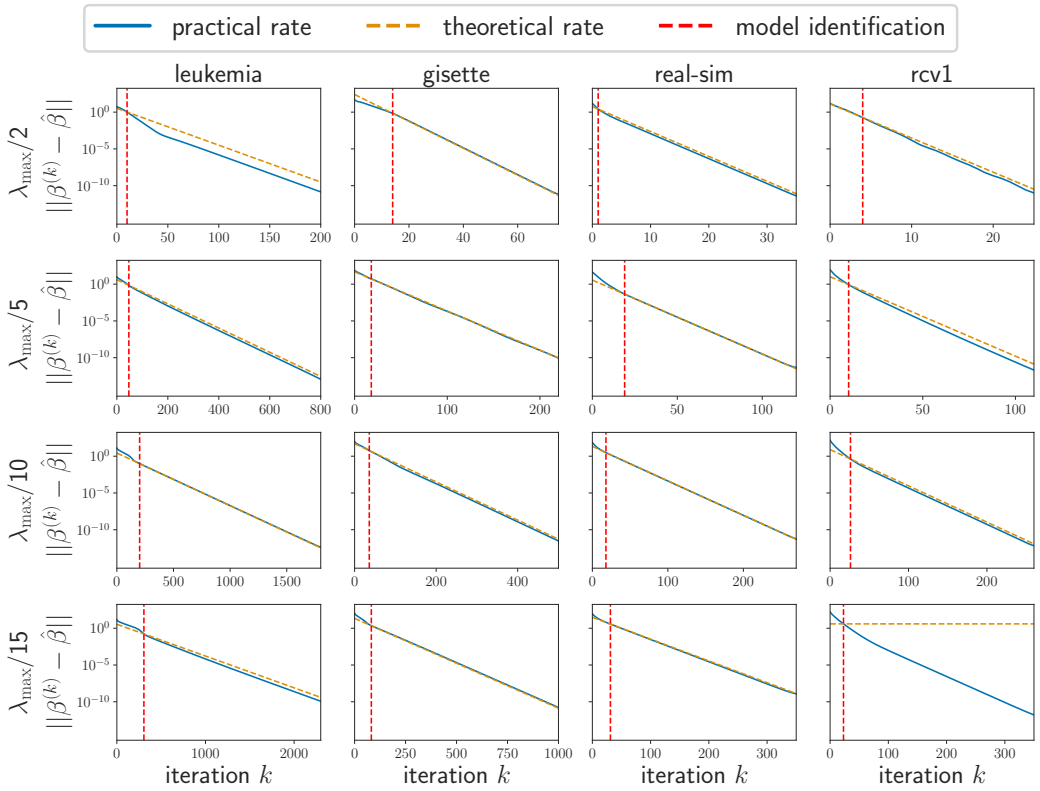


Figure 2.2 – **Sparse logistic regression, linear convergence.** Distance to optimum, $\|\beta^{(k)} - \hat{\beta}\|$, as a function of the number of iterations k, on 4 different datasets: *leukemia*, *gisette*, *rcv1*, and *real-sim*.

Sparse logistic regression. The sparse logistic regression is an estimator for classification tasks. It is the solution of the following optimization problem, for a design matrix $X \in \mathbb{R}^{n \times p}$ and a target variable $y \in \{-1, 1\}^n$, with $\sigma(z) \triangleq \frac{1}{1+e^{-z}}$:

$$\arg\min_{\beta \in \mathbb{R}^p} -\frac{1}{n} \sum_{i=1}^{n} \log \sigma(y_i \beta^\top X_{i:}) + \lambda \|\beta\|_1 \ . \tag{2.44}$$

The CD update for the sparse logistic regression is

$$\beta_j \leftarrow \mathrm{ST}_{\gamma_j \lambda} \left(\beta_j - \gamma_j X_{:j}^\top (y \odot (\sigma(y \odot X\beta) - 1)) \right) \ . \tag{2.45}$$

The constant stepsizes for the CD algorithm to solve Equation (2.44) are given by $1/\gamma_j = \frac{\|X_{:j}\|^2}{4n}$.

Support-vector machine. (Boser et al., 1992) The support-vector machine (SVM) primal optimization problem is, for a design matrix $X \in \mathbb{R}^{n \times p}$ and a target variable $y \in \{-1, 1\}^n$:

$$\arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2} \|\beta\|^2 + C \sum_{i=1}^{n} \max\left(1 - y_i X_{i:} \beta, 0\right) \ . \tag{2.46}$$

The SVM can be solved using the following dual optimization problem:

$$\arg\min_{w \in \mathbb{R}^n} \frac{1}{2} w^\top (y \odot X)(y \odot X)^\top w - \sum_{i=1}^{n} w_i + \sum_{i=1}^{n} \iota_{0 \le w_i \le C} \ . \tag{2.47}$$

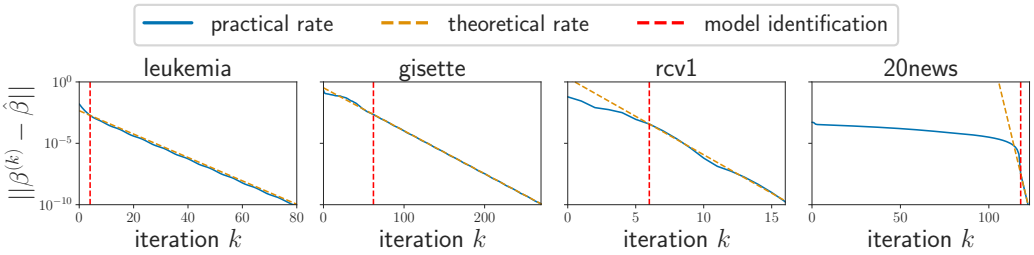


Figure 2.3 – **Support vector machine, linear convergence.** Distance to optimum, $\|\beta^{(k)} - \hat{\beta}\|$, as a function of the number of iterations k, on 4 different datasets: *leukemia*, *gisette*, *rcv1* and *20news*.

Table 2.2 – C values for SVM.

dataset	leukemia	gisette	rcv1	20news
C value	10	$1.5\ 10^{-2}$	$1.5\ 10^{-2}$	$5\ 10^{-1}$

The CD update for the SVM reads:

$$w_i \leftarrow \mathcal{P}_{[0,C]}\left(w_i - \gamma_i((y \odot X)_{i:}^\top (y \odot Xw) - 1)\right) \ , \tag{2.48}$$

where $\mathcal{P}_{[0,C]}(x) = \min(\max(0, x), C)$. The stepsizes of the CD algorithm to solve Equation (2.47) are given by $1/\gamma_i = \|(y \odot X)_{i:}\|^2$. The values of the regularization parameter C for each dataset from Figure 2.3 are given in Table 2.2.

Comments on Figures 2.1 to 2.3. Finite time model identification and local linear convergence are illustrated on the Lasso, the sparse logistic regression and the SVM in Figures 2.1 to 2.3. As predicted by Theorem 2.13, the relative model is identified after a finite number of iterations. For the Lasso (Figure 2.1) and the sparse logistic regression (Figure 2.2), we observe that as the regularization parameter gets smaller, the number of iterations needed by the CD algorithm to identify the model increases. To our knowledge, this is a classical empirical observation, that is not backed up by theoretical results. After identification, the convergence towards a solution is linear as predicted by Theorem 2.16. The theoretical local speed of convergence provided by Theorem 2.16 seems like a sharp estimation of the true speed of convergence as illustrated by the three figures.

Note that on Figures 2.1 to 2.3 high values of λ (or small values of C) were required for the restricted injectivity Assumption 2.15 to hold. Indeed, despite its lack of theoretical foundation, it is empirically observed that, in general, the larger the value of λ, the smaller the cardinal of the generalized support: $|\mathcal{S}|$. It makes the restricted injectivity Assumption 2.15: $\nabla^2_{\mathcal{S},\mathcal{S}} f(\hat{\beta}) \succ 0$ easier to be satisfied. For instance, for $\lambda = \lambda_{\max}/20$, the restricted injectivity Assumption 2.15 was not verified for a lot of datasets for the Lasso and the sparse logistic regression (Figures 2.1 and 2.2). In the same vein, values of C for the SVM had to be chosen small enough, in order to make $|\mathcal{S}|$ not too large (Figure 2.3).

Note that finite time model identification is crucial to ensure local linear convergence, see for instance *20news* dataset on Figure 2.3. However there exists very few quantitative theoretical results for the convergence speed of the model identification. Nutini et al. (2019); Sun et al. (2019) tried to obtain some rates on the identification, quantifying "how much the problem is qualified", *i.e.,* how much Assumption 2.4 is satisfied. But

these theoretical results do not seem to explain fully the experimental results of the CD: in particular the identification speed of the model compared to other algorithms.

Limits. We would like to point out the limit of our analysis illustrated for the case of $\lambda = \lambda_{\max}/15$ for the sparse logistic regression and the *rcv1* dataset in Figure 2.2. In this case, the solution may no longer be unique. The support gets larger and Assumption 2.15 is no longer met. In this case, the largest eigenvalue of $\mathcal{J}\psi(\hat{\beta}_S)$ is exactly one, which leads to the constant rate observed in Figure 2.2. Despite the largest eigenvalue being exactly 1, a regime of locally linear convergence toward a (potentially non unique) minimizer is still observed. Linear convergence of non-strongly convex functions starts to be more and more understood (Necoara et al., 2019). Figure 2.2 with $\lambda = \lambda_{\max}/15$ for *rcv1* suggests extensions of Necoara et al. (2019) could be possible in the nonsmooth case.

2.5 Conclusion

In conclusion, we show finite time model identification for coordinate descent Algorithm 2.1 (Theorem 2.13). Thanks to this identification property we were able to show local linear rates of convergence (Theorem 2.16). These two theoretical results were illustrated on popular estimators (Lasso, sparse logistic regression and SVM dual) and popular machine learning datasets (Section 2.4).

A first natural extension of this chapter would be to investigate block coordinate minimization: Theorem 2.13 could be extended for blocks under general partial smoothness assumption (Hare and Lewis, 2004). However, it seems that Theorem 2.16 would require a more careful analysis. A second extension could be to show linear convergence without the restricted injectivity (Assumption 2.15), paving the way for a generalization of Necoara et al. (2019) as suggested by Figure 6.4.

3

Anderson acceleration

Contents

Acceleration of first order methods is mainly obtained via inertia à la Nesterov, or via nonlinear extrapolation. The latter has known a recent surge of interest, with successful applications to gradient and proximal gradient techniques. On multiple machine learning problems, coordinate descent achieves performance significantly superior to full-gradient methods. Speeding up coordinate descent in practice is notoriously hard: inertially accelerated versions of coordinate descent are theoretically accelerated, but might not always lead to practical speed-ups. Capitalizing on the identification (Theorem 2.13) and local linear convergence (Theorem 2.16) from Chapter 2, we propose an accelerated version of coordinate descent using Anderson extrapolation.

This chapter is based on the following work, accepted at AISTATS 2021:

- **Q. Bertrand** and M. Massias. Anderson acceleration of coordinate descent. In *AISTATS*, 2021

3.1 Introduction

Gradient descent is the workhorse of modern convex optimization (Nesterov, 2004; Beck, 2017). For composite problems, proximal gradient descent retains the nice properties enjoyed by the latter. In both techniques, inertial acceleration achieves accelerated convergence rates (Nesterov, 1983; Beck and Teboulle, 2009).

Coordinate descent is a variant of gradient descent, which updates the iterates one coordinate at a time (Tseng and Yun, 2009b; Friedman et al., 2010). Proximal coordinate descent has been applied to numerous machine learning problems (Shalev-Shwartz and Zhang, 2013a; Wright, 2015; Shi et al., 2016), in particular the Lasso (Tibshirani, 1996),

elastic net (Zou and Hastie, 2005) or sparse logistic regression (Ng, 2004). It is used in preeminent packages such as scikit-learn (Pedregosa et al., 2011), glmnet (Friedman et al., 2009), libsvm (Fan et al., 2008) or lightning (Blondel and Pedregosa, 2016). On the theoretical side, inertial accelerated versions of coordinate descent (Nesterov, 2012; Lin et al., 2014; Fercoq and Richtárik, 2015) achieve accelerated rates. Note that usual lower bounds (Nesterov, 2004, Sec. 2.1.2) are derived for algorithms with iterates lying in the span of previous gradients, which is not the case for coordinate descent. However there also exists similar lower bounds for cyclic coordinate descent (Sun and Ye, 2019).

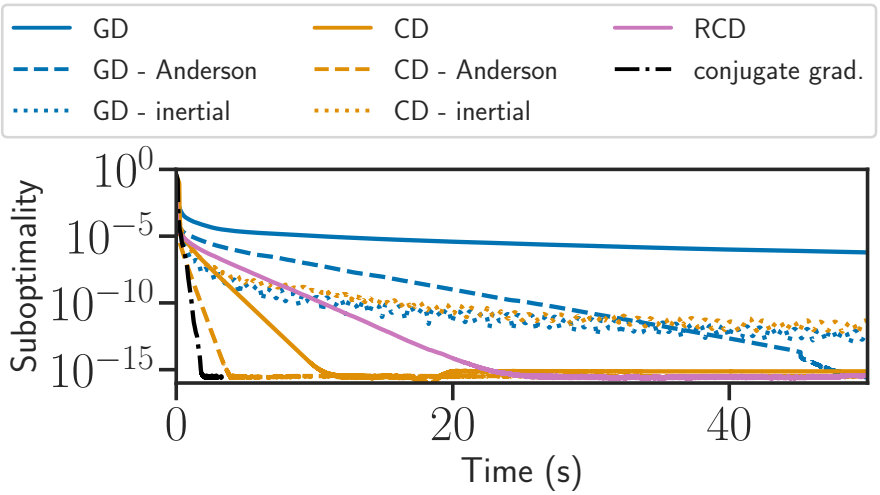


Figure 3.1 – Suboptimality along time for a quadratic problem on the 5000 first features of the *rcv1* dataset. GD: gradient descent, CD: cyclic coordinate descent, RCD: randomized coordinate descent.

To obtain accelerated rates, Anderson extrapolation (Anderson, 1965) is an alternative to inertia: it provides acceleration by exploiting the iterates' structure. This procedure has been known for a long time, under various names and variants (Wynn, 1962; Eddy, 1979; Smith et al., 1987), see Sidi (2017); Brezinski et al. (2018) for reviews. Anderson acceleration enjoys accelerated rates on quadratics (Golub and Varga, 1961), but theoretical guarantees in the nonquadratic case are weaker (Scieur et al., 2016). Interestingly, numerical performance still shows significant improvements on nonquadratic objectives. Anderson acceleration has been adapted to various algorithms such as Douglas-Rachford (Fu et al., 2019), ADMM (Poon and Liang, 2019) or proximal gradient descent (Zhang et al., 2018; Mai and Johansson, 2019; Poon and Liang, 2020). Among main benefits, the practical version of Anderson acceleration is memory efficient, easy to implement, line search free, has a low cost per iteration and does not require knowledge of the strong convexity constant. Finally, it introduces a single additional parameter, which often does not require tuning (see Section 3.3.1).

In this work:

- We propose an Anderson acceleration scheme for cyclic coordinate descent, which, as visible on Figure 3.1, outperforms inertial and extrapolated gradient descent, as well as inertial and randomized coordinate descent.
- The acceleration is obtained even though the iteration matrix is not symmetric, a notable problem in the analysis of Anderson extrapolation.

Algorithm 3.1 Offline Anderson extrapolation

init: $\beta^{(0)} \in \mathbb{R}^p$
for $k = 1, \dots$ **do**
  `// regular linear iteration`
  $\beta^{(k)} = T\beta^{(k-1)} + b$
  $U = [\beta^{(1)} - \beta^{(0)}, \dots, \beta^{(k)} - \beta^{(k-1)}]$
  `// solve a linear system of size k`
  $c = (U^\top U)^{-1}\mathbf{1}_k \in \mathbb{R}^k$
  $c \mathrel{/}= \mathbf{1}_k^\top c$
  `// does not affect` $\beta^{(k)}$
  $\beta^{(k)}_{\text{e-off}} = \sum_{i=1}^k c_i \beta^{(i)}$
return $\beta^{(k)}_{\text{e-off}}$

Algorithm 3.2 Online Anderson extrapolation

init: $\beta^{(0)} \in \mathbb{R}^p, K \in \mathbb{N}$
for $k = 1, \dots$ **do**
  `// regular iteration`
  $\beta^{(k)} = T\beta^{(k-1)} + b$
  if $k = 0 \mod K$ **then**
    $U = [\beta^{(k-K+1)} - \beta^{(k-K)}, \dots,$
     $\beta^{(k)} - \beta^{(k-1)}]$
    `// solve a linear system of size K`
    $c = (U^\top U)^{-1}\mathbf{1}_K \in \mathbb{R}^K$
    $c \mathrel{/}= \mathbf{1}_K^\top c$
    $\beta^{(k)}_{\text{e-on}} = \sum_{i=1}^K c_i \beta^{(k-K+i)}$
    `// base sequence changes`
    $\beta^{(k)} = \beta^{(k)}_{\text{e-on}}$
return $\beta^{(k)}$

- We empirically highlight that the proposed acceleration technique can generalize in the non-quadratic case (Algorithm 3.3) and significantly improve proximal coordinate descent algorithms (Section 3.3), which are state-of-the-art first order methods on the considered problems.

Notation. The vector of size K with all one entries is written $\mathbf{1}_K$.

3.2 Anderson extrapolation

3.2.1 Background

Anderson extrapolation is designed to accelerate the convergence of sequences based on fixed point linear iterations, that is:

$$\beta^{(k+1)} = T\beta^{(k)} + b \ , \tag{3.1}$$

where the *iteration matrix* $T \in \mathbb{R}^{p\times p}$ has spectral radius $\rho(T) < 1$. There exist two variants: offline and online, which we recall briefly.

Offline extrapolation (Algorithm 3.1), at iteration k, looks for a fixed point as an affine combination of the k first iterates: $\beta^{(k)}_{\text{e-off}} = \sum_1^k c_i^{(k)} \beta^{(i-1)}$, and solves for the coefficients $c^{(k)} \in \mathbb{R}^k$ as follows:

$$\begin{aligned} c^{(k)} &= \underset{\sum_1^k c_i = 1}{\arg\min} \|\textstyle\sum_1^k c_i \beta^{(i-1)} - T\sum_1^k c_i \beta^{(i-1)} - b\|^2 \\ &= \underset{\sum_1^k c_i = 1}{\arg\min} \|\textstyle\sum_1^k c_i \left(\beta^{(i)} - \beta^{(i-1)}\right)\|^2 \\ &= (U^\top U)^{-1}\mathbf{1}_k / \mathbf{1}_k^\top (U^\top U)^{-1}\mathbf{1}_k \ , \end{aligned} \tag{3.2}$$

where $U = [\beta^{(1)} - \beta^{(0)}, \dots, \beta^{(k)} - \beta^{(k-1)}] \in \mathbb{R}^{p\times k}$ (and hence the objective rewrites $\left\|Uc\right\|^2$). In practice, since $\beta^{(k)}$ is available when $c^{(k)}$ is computed, one uses $\beta^{(k)}_{\text{e}} = \sum_1^k c_i^{(k)} \beta^{(i)}$ instead of $\sum_1^k c_i^{(k)} \beta^{(i-1)}$. The motivation for introducing the coefficients

$c^{(k)}$ is discussed in more depth after Prop. 6 in Massias et al. (2020b), and details about the closed-form solution can be found in Scieur et al. (2016, Lem. 2.4). In offline acceleration, more and more base iterates are used to produce the extrapolated point, but the extrapolation sequence does not affect the base sequence. This may not scale well since it requires solving larger and larger linear systems.

A more practical variant is the *online* version (Algorithm 3.2) considered in this chapter. The number of points to be extrapolated is fixed to K; $\beta^{(1)}, \dots, \beta^{(K)}$ are computed normally with the fixed point iterations, but $\beta_{\text{e}}^{(K)}$ is computed by extrapolating the iterates from $\beta^{(1)}$ to $\beta^{(K)}$, and $\beta^{(k)}$ is taken equal to $\beta_{\text{e}}^{(K)}$. K normal iterates are then computed from $\beta^{(K+1)}$ to $\beta^{(2K)}$ then extrapolation is performed on these last K iterates, etc.

As we recall below, results on Anderson acceleration mainly concern fixed-point iterations with symmetric iteration matrices T, and results concerning non-symmetric iteration matrices are weaker (Bollapragada et al., 2018). Poon and Liang (2020, Thm 6.4) do not assume that T is symmetric, but only diagonalizable, which is still a strong requirement.

Proposition 3.1 (Symmetric T, Scieur 2019)**.** *Let the iteration matrix T be symmetric semi-definite positive, with spectral radius $\rho = \rho(T) < 1$. Let $\hat{\beta}$ be the limit of the sequence $(\beta^{(k)})$. Let $\zeta = (1-\sqrt{1-\rho})/(1+\sqrt{1-\rho})$. Then the iterates of offline Anderson acceleration satisfy, with $B = (\mathrm{Id} - T)^2$,*

$$\|\beta_{\text{e-off}}^{(k)} - \hat{\beta}\|_B \leq \frac{2\zeta^{k-1}}{1+\zeta^{2(k-1)}} \|\beta^{(0)} - \hat{\beta}\|_B \ , \tag{3.3}$$

and those of online extrapolation satisfy:

$$\|\beta_{\text{e-on}}^{(k)} - \hat{\beta}\|_B \leq \left(\frac{2\zeta^{K-1}}{1+\zeta^{2(K-1)}} \right)^{k/K} \|\beta^{(0)} - \hat{\beta}\|_B \ . \tag{3.4}$$

Scieur et al. (2016) showed that the offline version in Proposition 3.1 matches the accelerated rate of the conjugate gradient (Hestenes and Stiefel, 1952). As it states, gradient descent can be accelerated by Anderson extrapolation on quadratics.

Application to quadratics The canonical application of Anderson extrapolation is gradient descent on quadratics. Consider a quadratic problem, with $b \in \mathbb{R}^p$, $H \in \mathbb{S}_{++}^p$ such that $H \succ 0$, L denotes the largest eigenvalue of H, $L \triangleq \|H\|_2$:

$$\hat{\beta} = \operatorname*{arg\,min}_{\beta \in \mathbb{R}^p} \frac{1}{2} \beta^\top H \beta + \langle b, \beta \rangle \ . \tag{3.5}$$

A typical instance is overdetermined least squares with full-column rank design matrix $X \in \mathbb{R}^{n \times p}$, and observations $y \in \mathbb{R}^n$, such that $H = X^\top X$ and $b = -X^\top y$. On Problem (3.5) gradient descent with step size $1/L$ reads:

$$\beta^{(k+1)} = \underbrace{\left(\mathrm{Id}_p - \tfrac{1}{L} H\right)}_{T^{\text{GD}} \in \mathbb{S}_+^p} \beta^{(k)} + \underbrace{(-b/L)}_{b^{\text{GD}}} \ . \tag{3.6}$$

Because they have this linear structure, iterates of gradient descent can benefit from Anderson acceleration, observing that the fixed point of $\beta \mapsto T^{\text{GD}}\beta + b^{\text{GD}}$ solves (3.5), with $T^{\text{GD}} \in \mathbb{S}_+^p$. Anderson acceleration of gradient descent has therefore been well-studied beyond the scope of machine learning (Pulay, 1980; Eyert, 1996). However, on

many machine learning problems, coordinate descent achieves far superior performance, and it is interesting to determine whether or not it can also benefit from Anderson extrapolation.

3.2.2 Linear iterations of coordinate descent

To apply Anderson acceleration to coordinate descent, we need to show that its iterates satisfy linear iterations as in (3.6). An epoch of cyclic coordinate descent for Problem (3.5) consists in updating the vector x one coordinate at a time, sequentially, i.e. for $j = 1, \dots, p$:

$$\beta_j \leftarrow \beta_j - \frac{1}{H_{jj}}(H_{j:}\beta + b_j) \ , \tag{3.7}$$

which can be rewritten, for $j = 1, \dots, p$:

$$\beta \leftarrow \left(\mathrm{Id}_p - \frac{e_j e_j^\top}{H_{jj}} H \right) \beta - \frac{b_j}{H_{jj}} e_j \ . \tag{3.8}$$

Thus, for primal iterates, one full pass (updating coordinates from 1 to p) leads to a linear iteration (as in Lemma 1.7 from Chapter 1), for some $b^{\mathrm{CD}} \in \mathbb{R}^p$:

$$\beta^{(k+1)} = T^{\mathrm{CD}} \beta^{(k)} + b^{\mathrm{CD}} \ , \tag{3.9}$$

with

$$T^{\mathrm{CD}} \triangleq \left(\mathrm{Id}_p - \frac{e_p e_p^\top}{H_{pp}} H \right) \dots \left(\mathrm{Id}_p - \frac{e_1 e_1^\top}{H_{11}} H \right) \ . \tag{3.10}$$

Note that in the case of coordinate descent we write $\beta^{(k)}$ for the iterates after one pass of coordinate descent on all features, and not after each update (3.7). The iterates of cyclic coordinate descent therefore also have a fixed-point structure, but contrary to gradient descent, their iteration matrix T^{CD} is not symmetric, which we address in Section 3.2.3. Note that random coordinate descent does not exhibit such a fixed-point structure.

3.2.3 Anderson extrapolation for nonsymmetric iteration matrices

Even on quadratics, Anderson acceleration with non-symmetric iteration matrices is less developed, and the only results concerning its theoretical acceleration are recent and weaker than in the symmetric case.

Proposition 3.2 (Bollapragada et al. 2018, Thm 2.2)**.** *When T is not symmetric, and $\rho(T) < 1$,*

$$\|\beta^{(k)}_{\mathrm{e-off}} - T\beta^{(k)}_{\mathrm{e-off}} - b\| \leq \|\mathrm{Id} - \rho(T - \mathrm{Id})\|_2 \|P^*(T)(\beta^{(1)} - \beta^{(0)})\| \ ,$$

where the unavailable polynomial P^* minimizes $\|P(T)(\beta^{(1)} - \beta^{(0)})\|$ amongst all polynomials P of degree exactly $k-1$ whose coefficients sum to 1.

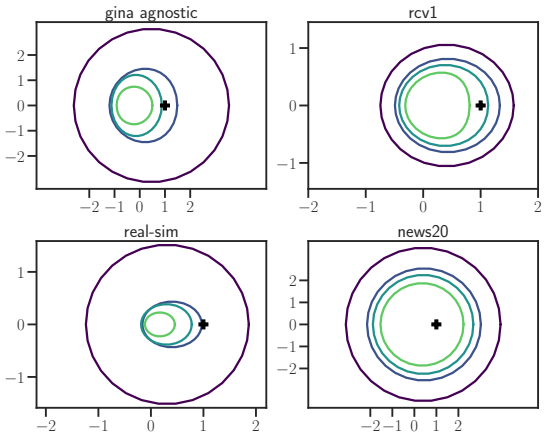


Figure 3.2 – Numerical range of T^q as q varies; T is the iteration matrix of Ridge regression problems with conditioning $\kappa = 10^3$, on 4 datasets. The black cross marks the $(1, 0)$ point, which should lie outside the range for the theoretical bound to be useful.

The quality of the bound (in particular, its eventual convergence to 0) crucially depends on $\|P(T)\|$. Using the Crouzeix conjecture (Crouzeix, 2004), Bollapragada et al. (2018) managed to bound $\|P(T)\|$, with P a polynomial:

$$\|P(T)\| \leq c \max_{z \in W(T)} |P(z)| \ , \tag{3.11}$$

with $c \geq 2$ (Crouzeix, 2007; Crouzeix and Palencia, 2017), and $W(T)$ the numerical range:

$$W(T) \triangleq \{\beta^* T \beta : \|\beta\|_2 = 1, \beta \in \mathbb{C}^p\} \ . \tag{3.12}$$

Since there is no general formula for this bound, Bollapragada et al. (2018) used numerical bounds on $W(T^q)$ to ensure convergence. Figure 3.2 displays the numerical range $W(T^q)$ in the complex plane for $q \in \{1, 128, 256, 512\}$. In order to be able to apply the theoretical result from Bollapragada et al. (2018), one must chose q such that the point $(1, 0)$ is not contained in $W(T^q)$, and extrapolate $\beta^{(0)}, \beta^{(q)}, \beta^{(2q)}, \ldots$ One can see on Figure 3.2 that large values of q are needed, unusable in practice: $q = 512$ is greater than the number of iterations needed to converge on some problems. Moreover, Anderson acceleration seems to provide speed up on coordinate descent even with $q = 1$ as we perform, which highlights the need for refined bounds for Anderson acceleration on nonsymmetric matrices.

We propose two means to fix this lack of theoretical results: to modify the algorithm in order to have a more amenable iteration matrix (Section 3.2.4), or to perform a simple cost function decrease check (Section 3.2.5).

3.2.4 Pseudo-symmetrization of T

A first idea to make coordinate descent theoretically amenable to extrapolation is to perform updates of coefficients from indices 1 to p, followed by a reversed pass from p to 1. This leads to an iteration matrix which is not symmetric either but friendlier: it writes

$$\begin{aligned} T^{\text{CD-sym}} &\triangleq \left(\mathrm{Id}_p - \frac{e_1 e_1^\top}{H_{11}} H\right) \times \cdots \times \left(\mathrm{Id}_p - \frac{e_p e_p^\top}{H_{pp}} H\right) \left(\mathrm{Id}_p - \frac{e_p e_p^\top}{H_{pp}} H\right) \times \cdots \times \left(\mathrm{Id}_p - \frac{e_1 e_1^\top}{H_{11}} H\right) \\ &= H^{-1/2} S H^{1/2} \ , \end{aligned} \tag{3.13}$$

with

$$S = \left(\mathrm{Id}_p - H^{1/2}\frac{e_1 e_1^\top}{H_{11}}H^{\frac{1}{2}}\right) \times \cdots \times \left(\mathrm{Id}_p - H^{\frac{1}{2}}\frac{e_p e_p^\top}{H_{pp}}H^{\frac{1}{2}}\right) \times \left(\mathrm{Id}_p - H^{\frac{1}{2}}\frac{e_p e_p^\top}{H_{pp}}H^{\frac{1}{2}}\right) \times \cdots \times \left(\mathrm{Id}_p - H^{\frac{1}{2}}\frac{e_1 e_1^\top}{H_{11}}H^{\frac{1}{2}}\right) . \tag{3.14}$$

S is symmetric, thus, S and T (which has the same eigenvalues as S), are diagonalisable with real eigenvalues. We call these iterations pseudo-symmetric, and show that this structure allows to preserve the guarantees of Anderson extrapolation.

Proposition 3.3 (Pseudosym. $T = H^{-1/2}SH^{1/2}$)**.** *Let T be the iteration matrix of pseudo-symmetric coordinate descent: $T = H^{-1/2}SH^{1/2}$, with S the symmetric positive semidefinite matrix of (3.13). Let $\hat{\beta}$ be the limit of the sequence $(\beta^{(k)})$. Let $\zeta = (1 - \sqrt{1-\rho})/(1+\sqrt{1-\rho})$. Then $\rho = \rho(T) = \rho(S) < 1$ and the iterates of offline Anderson acceleration satisfy, with $B = (T - \mathrm{Id})^\top(T - \mathrm{Id}) \succ 0$:*

$$\|\beta^{(k)}_{\text{e-off}} - \hat{\beta}\|_B \leq \sqrt{\kappa(H)}\frac{2\zeta^{k-1}}{1+\zeta^{2(k-1)}}\|\beta^{(0)} - \hat{\beta}\|_B \ , \tag{3.15}$$

and thus those of online extrapolation satisfy:

$$\|\beta^{(k)}_{\text{e-on}} - \hat{\beta}\|_B \leq \left(\sqrt{\kappa(H)}\frac{2\zeta^{K-1}}{1+\zeta^{2(K-1)}}\right)^{k/K}\|\beta^{(0)} - \hat{\beta}\|_B \ . \tag{3.16}$$

Proof (Proposition 3.3). First we link the quantity computed in Equation (3.2) to the extrapolated quantity $\sum_{i=1}^k c_i\beta^{(i-1)}$ (Lemma 3.4 (a)), then we bound $\|\beta^{(k)}_{\text{e-on}} - \hat{\beta}\|_B$ (Lemma 3.4 (b)).

Lemma 3.4. *a) For all $c \in \mathbb{R}^k$ such that $\sum_{i=1}^k c_i = 1$:*

$$\sum_{i=1}^k c_i(\beta^{(i)} - \beta^{(i-1)}) = (T - \mathrm{Id})\left(\sum_{i=1}^k c_i\beta^{(i-1)} - \hat{\beta}\right) \ . \tag{3.17}$$

b) For all $c \in \mathbb{R}^k$ such that $\sum_{i=1}^k c_i = 1$,

$$\|(T - \mathrm{Id})(\beta^{(k)}_{\text{e-off}} - \hat{\beta})\| \leq \sqrt{\kappa(H)}\left\|\sum_{i=0}^{k-1} c_i S^i\right\|\|(T - \mathrm{Id})(\beta^{(0)} - \hat{\beta})\| \ . \tag{3.18}$$

Here we link the quantity computed in Equation (3.2) to the extrapolated quantity $\sum_{i=1}^k c_i\beta^{(i-1)}$.

Proof (Lemma 3.4 (a)). Since $\beta^{(i)} = T\beta^{(i-1)} + (\hat{\beta} - T\hat{\beta})$,

$$\begin{aligned} c_i(\beta^{(i)} - \beta^{(i-1)}) &= c_i(T\beta^{(i-1)} + \hat{\beta} - T\hat{\beta} - \beta^{(i-1)}) \\ &= (T - \mathrm{Id})c_i(\beta^{(i-1)} - \hat{\beta}) \ . \end{aligned} \tag{3.19}$$

Hence, since $\sum_1^k c_i = 1$,

$$\sum_{i=1}^k c_i(\beta^{(i)} - \beta^{(i-1)}) = (T - \mathrm{Id})\left(\sum_{i=1}^k c_i\beta^{(i-1)} - \hat{\beta}\right) \ . \tag{3.20}$$

■

Here we bound $\|\beta_{\text{e}-\text{on}}^{(k)} - \hat{\beta}\|_B$.

Proof (Lemma 3.4 (b)). In this proof, we denote by c^* the solution of (3.2). We use the fact that for all $c \in \mathbb{R}^k$ such that $\sum_{i=1}^k c_i = 1$,

$$\|\sum_{i=1}^k c_i^*(\beta^{(i)} - \beta^{(i-1)})\| = \min_{\substack{c\in\mathbb{R}^k \\ \sum_i c_i=1}} \|\sum_{i=1}^k c_i(\beta^{(i)} - \beta^{(i-1)})\| \leq \|\sum_{i=1}^k c_i(\beta^{(i)} - \beta^{(i-1)})\| \ . \tag{3.21}$$

Then we use twice Lemma 3.4 (a) for the left-hand and right-hand side of Equation (3.21). Using Lemma 3.4 (a) with the c_i^* minimizing Equation (3.2) we have for all $c_i \in \mathbb{R}$ such that $\sum_{i=1}^k c_i = 1$:

$$\begin{aligned}
\|(T - \mathrm{Id})(\beta_{\text{e}} - \hat{\beta})\| &= \|\sum_{i=1}^k c_i^*(\beta^{(i)} - \beta^{(i-1)})\| \\
&\leq \|\sum_{i=1}^k c_i(\beta^{(i)} - \beta^{(i-1)})\| \\
&= \|(T - \mathrm{Id})\sum_{i=1}^k c_i(\beta^{(i-1)} - \hat{\beta})\| \\
&= \|(T - \mathrm{Id})\sum_{i=1}^k c_i T^{i-1}(\beta^{(0)} - \hat{\beta})\| \\
&= \|\sum_{i=1}^k c_i T^{i-1}(T - \mathrm{Id})(\beta^{(0)} - \hat{\beta})\| \\
&= \|\sum_{i=1}^k c_i T^{i-1}\| \times \|(T - \mathrm{Id})(\beta^{(0)} - \hat{\beta})\| \\
&\leq \|H^{-1/2}\sum_{i=1}^k c_i S^{i-1} H^{1/2}\| \times \|(T - \mathrm{Id})(\beta^{(0)} - \hat{\beta})\| \\
&\leq \sqrt{\kappa(H)}\|\sum_{i=1}^k c_i S^{i-1}\| \times \|(T - \mathrm{Id})(\beta^{(0)} - \hat{\beta})\| \ .
\end{aligned} \tag{3.22}$$

■

Finally, to conclude the proof of Proposition 3.3, we apply Lemma 3.4 (b) by choosing c_i equal to the Chebyshev weights c_i^{Cb}. Using the proof of Barré et al. (2020, Prop. B. 2), we have, with $\zeta = \frac{1-\sqrt{1-\rho(T)}}{1+\sqrt{1-\rho(T)}}$:

$$\|\sum_{i=1}^k c_i^{\text{Cb}} S^{i-1}\| \leq \tfrac{2\zeta^{k-1}}{1+\zeta^{2(k-1)}} \ . \tag{3.23}$$

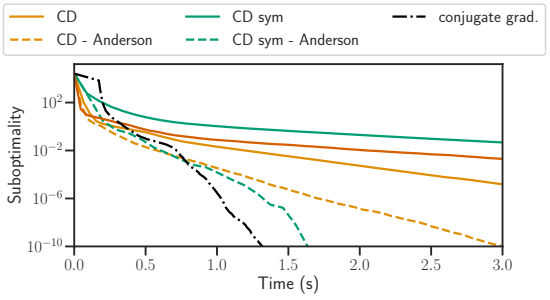


Figure 3.3 – **OLS, *rcv1*.** Suboptimality as a function of time on the 5000 first columns of the dataset *rcv1*.

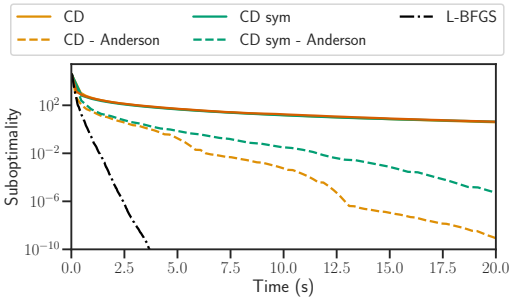


Figure 3.4 – **ℓ_2-regularized logistic regression, *real-sim*.** Suboptimality as a function of time on the 2000 first features of the *real-sim* dataset, Tikhonov strength set so that $\kappa = 10^5$.

Combined with Lemma 3.4 (b) this concludes the proof:

$$\|(T - \mathrm{Id})(\beta_e - \hat{\beta})\| \leq \sqrt{\kappa(H)}\|\sum_{i=1}^{k} c_i S^{i-1}\|\|(T - \mathrm{Id})(\beta^{(0)} - \hat{\beta})\| \tag{3.24}$$

$$\leq \sqrt{\kappa(H)}\frac{2\zeta^{k-1}}{1+\zeta^{2(k-1)}}\|(T - \mathrm{Id})(\beta^{(0)} - \hat{\beta})\| \ . \tag{3.25}$$

Proposition 3.3 shows accelerated convergence rates for the offline Anderson acceleration, but a $\sqrt{\kappa(H)}$ appears in the rate of the online Anderson acceleration, meaning that K must be large enough that ζ^K mitigates this effect. This factor however seems like a theoretical artefact of the proof, since we observed significant speed up of the online Anderson acceleration, even with bad conditioning of H (see Figure 3.3). ■

Figure 3.3 illustrates the convergence speed of cyclic and pseudo-symmetric coordinate descent on the *rcv1* dataset. Anderson acceleration provides speed up for both versions. Interestingly, on this quadratic problem, the non extrapolated pseudo-symmetric iterations perform poorly, worse than cyclic coordinate descent. However, the performance is reversed for their extrapolated counterparts: the pseudo-symmetrized version is better than the cyclic one (which has a nonsymmetric iteration matrix). Finally, Anderson extrapolation on the pseudo-symmetrized version even reaches the conjugate gradient performance.

3.2.5 Generalization to nonquadratic and proposed algorithm

After devising and illustrating an Anderson extrapolated coordinate descent procedure for a simple quadratic objective, our goal is to apply Anderson acceleration on problems where coordinate descent achieve state-of-the-art results, *i.e.*, of the form:

$$\min_{\beta \in \mathbb{R}^p} \Phi(\beta) = f(\beta) + \lambda g(\beta) \triangleq F(X\beta) + \lambda \sum_{j=1}^{p} g_j(\beta_j) \ ,$$

where $F : \mathbb{R}^n \to \mathbb{R}$ is convex, smooth and g_j's are proper, closed and convex functions As examples, we allow $g = 0$, $g = \|\beta\|_1$, $g = \frac{1}{2}\|\beta\|_2^2$, $g = \|\beta\|_1 + \frac{\rho}{2\lambda}\|x\|^2$.

Algorithm 3.3 Online Anderson proximal gradient descent (proposed)

init: $\beta^{(0)} \in \mathbb{R}^p$
for $k = 1, \dots$ **do**
  $\beta = \beta^{(k-1)}$
  for $j = 1, \dots p$ **do**
    $\beta_j^{\text{old}} = \beta_j$
    $\beta_j = \text{prox}_{\frac{\lambda}{L_j} g_j}(\beta_j - X_{:j}^\top \nabla F(X\beta)/L_j)$
    $X\beta \mathrel{+}= (\beta_j - \beta_j^{\text{old}}) X_{:j}$
  $\beta^{(k)} = \beta$ // regular iter. $\mathcal{O}(np)$
  if $k = 0 \mod K$ **then** // extrapol., $\mathcal{O}(K^3 + pK^2)$
    $U = [\beta^{(k-K+1)} - \beta^{(k-K)}, \dots, \beta^{(k)} - \beta^{(k-1)}]$
    $c = (U^\top U)^{-1} \mathbf{1}_K / \mathbf{1}_K^\top (U^\top U)^{-1} \mathbf{1}_K \in \mathbb{R}^K$
    $\beta_e = \sum_{i=1}^K c_i \beta^{(k-K+i)}$
    if $F(X\beta_e) + \lambda g(\beta_e) \leq F(\beta^{(k)}) + \lambda g(\beta^{(k)})$ **then**
      $\beta^{(k)} = \beta_e$
return $\beta^{(k)}$

In the nonquadratic case, updates of proximal coordinate descent do not lead to a linear iteration. In this case, T is not a matrix, but a nonlinear operator. However, as stated in Proposition 3.5, asymptotically, the fixed-point operator T is linear.

Proposition 3.5. *Consider a solution* $\hat{\beta} \in \arg\min_{\beta \in \mathbb{R}^p} \Phi(\beta)$. *Suppose*

1. *Assumptions 2.1 to 2.4, 2.12 and 2.15 hold.*
2. *The sequence* $(\beta^{(k)})_{k \geq 0}$ *generated by Algorithm 2.1 converges to* $\hat{\beta}$.

Then there exists $K \in \mathbb{N}$, $\psi : \mathbb{R}^{|\mathcal{S}|} \to \mathbb{R}^{|\mathcal{S}|}$ *(defined in Equation (2.18)) such that, for all* $k \in \mathbb{N}, k \geq K$*:*

$$\beta_j^{(k)} = \hat{\beta}_j \text{ , for all } j \in \mathcal{S}^c, \tag{3.26}$$

$$\beta_{\mathcal{S}}^{(k+1)} - \hat{\beta}_{\mathcal{S}} = \mathcal{J}\psi(\hat{\beta}_{\mathcal{S}})(\beta_{\mathcal{S}}^{(k)} - \hat{\beta}_{\mathcal{S}}) + \mathcal{O}(\|\beta_{\mathcal{S}}^{(k)} - \hat{\beta}_{\mathcal{S}}\|^2) \ , \tag{3.27}$$

and

$$\rho\left(\mathcal{J}\psi(\hat{\beta}_{\mathcal{S}})\right) < 1 \ . \tag{3.28}$$

Proof Proposition 3.5 is a direct consequence of Theorems 2.13 and 2.16. ■

Figure 3.4 shows the performance of Anderson extrapolation on a ℓ^2-regularized logistic regression problem:

$$\arg\min_{\beta \in \mathbb{R}^p} \sum_{i=1}^n \log(1 + e^{-y_i X_{i:} \beta}) + \frac{\lambda}{2} \|\beta\|_2^2 \ . \tag{3.29}$$

One can see that despite the better theoretical properties of the pseudo-symmetrized coordinate descent, Anderson acceleration on coordinate descent seems to work better on the cyclic coordinate descent. We thus choose to apply Anderson extrapolation on

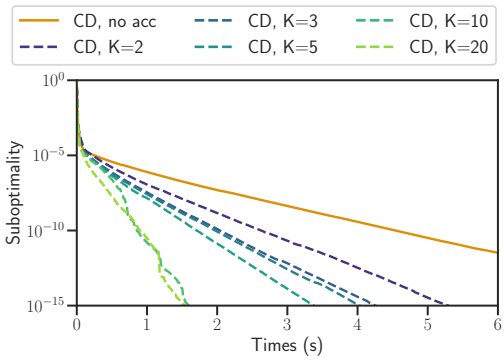


Figure 3.5 – **Influence of K, quadratic, *rcv1*.** Influence of the number of iterates K used to perform Anderson extrapolation with coordinate descent (CD) on a quadratic with the *rcv1* dataset (2000 first columns).

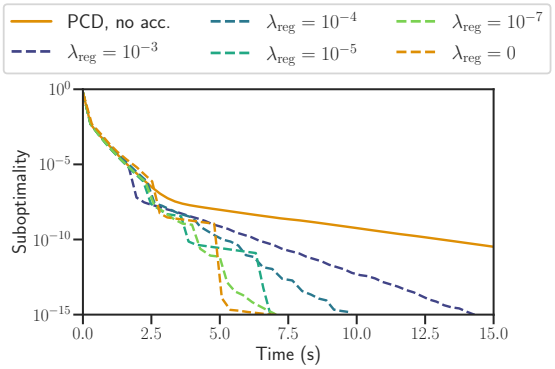


Figure 3.6 – **Influence of λ_{reg}, sparse logitic regression, *rcv1*.** Influence of the regularization amount when solving a sparse logistic regression using Anderson extrapolation with proximal coordinate descent (PCD) on the *rcv1* dataset, $K = 5$, $\lambda = \lambda_{\max}/30$.

the cyclic coordinate descent (Algorithm 3.3), while adding a step checking the decrease of the objective function in order to ensure convergence.

Finally, we can also use Algorithm 3.3 in the non smooth case where $g = \|\cdot\|_1$, since coordinate descent achieves support identification when the solution is unique, after which the objective becomes differentiable. There is therefore a linear structure after a sufficient number of iterations.

3.3 Experiments

An implementation relying on numpy, numba and cython (McKinney, 2012; Lam et al., 2015; Behnel et al., 2011), with scripts to reproduce the figures, is available at: `https://mathurinm.github.io/andersoncd`.

We first show how we set the hyperparameters of Anderson extrapolation (Section 3.3.1). Then we show that Anderson extrapolation applied to proximal coordinate descent outperforms other first order algorithms on standard machine learning problems (Section 3.3.2).

3.3.1 Parameter setting

Anderson extrapolation relies on 2 hyperparameters: the number of extrapolated points K, and the amount of regularization eventually used when solving the linear system to obtain the coefficients $c \in \mathbb{R}^K$. Based on the conclusions of this section, we fix these parameters for all the subsequent experiments in Section 3.3.2: *no regularization and* $K = 5$.

Influence of the regularization. Scieur et al. (2016) provided accelerated complexity rates for *regularized* Anderson extrapolation: a term $\lambda_{\text{reg}} \|c\|^2$ is added to the objective of Equation (3.2). The closed-form formula for the coefficients is then $(U^\top U + \lambda_{\text{reg}} \operatorname{Id}_K)^{-1}\mathbf{1}_K / \mathbf{1}_K^\top (U^\top U + \lambda_{\text{reg}} \operatorname{Id}_K)^{-1}\mathbf{1}_K$.

However, similarly to Mai and Johansson (2019) and Poon and Liang (2020) we observed that regularizing the linear system does not seem necessary, and can even hurt the convergence speed. Figure 3.6 shows the influence of the regularization parameter on the convergence on the *rcv1* dataset for a sparse logistic regression problem, with $K = 5$ and $\lambda = \lambda_{\max}/30$. The more the optimization problem is regularized, the more the convergence speed is deteriorated. Thus *we choose not to regularize* when solving the linear system for the extrapolation coefficients. We simply check if the extrapolated point yields a lower objective function than the current regular iterate (see Algorithm 3.3).

Influence of K. Figure 3.5 shows the impact of K on the convergence speed. Although the performance depends on K, it seems that the dependency is loose, as for $K \in \{10, 20\}$ the acceleration is roughly the same. Therefore, we do not treat K as a parameter and fix it to $K = 5$.

Computational overhead of Anderson extrapolation. With nnz the number of nonzero coefficients, K epochs (*i.e.,* K updates of all coordinates) of CD *without Anderson acceleration* cost:

$$K\text{nnz}(X) \ .$$

Every K epochs, Algorithm 3.3 requires to solve a $K \times K$ linear system. Thus, K epochs of CD *with Anderson acceleration* cost:

$$\underbrace{K\text{nnz}(X)}_{K \text{ passes of CD}} + \underbrace{K^2\text{nnz}(w)}_{\text{form } U^\top U} + \underbrace{K^3}_{\text{solve system}} \ .$$

With our choice, $K = 5$, the overhead of Anderson acceleration is marginal compared to a gradient call: $K^2 + K\text{nnz}(w) \ll \text{nnz}(X)$. This can be observed in Figures 3.7 to 3.9: even before acceleration actually occurs, Anderson PCD is not slower than regular PCD.

3.3.2 Numerical comparison on machine learning problems

We compare multiple algorithms to solve popular machine learning problems: the Lasso, the elastic net, the sparse logistic regression and the group Lasso. The compared algorithms are the following:

- Proximal gradient descent (PGD, Lions and Mercier 1979; Combettes and Wajs 2005).
- Nesterov-like inertial PGD (FISTA, Beck and Teboulle 2009).
- Anderson accelerated PGD (Mai and Johansson, 2019; Poon and Liang, 2020).
- Proximal coordinate descent (PCD, Tseng and Yun 2009b).
- Proximal coordinate descent with random index selection (PRCD, Richtárik and Takáč 2014).
- Inertial PCD (Lin et al., 2014; Fercoq and Richtárik, 2015).
- Anderson accelerated PCD (ours, Algorithm 3.3).

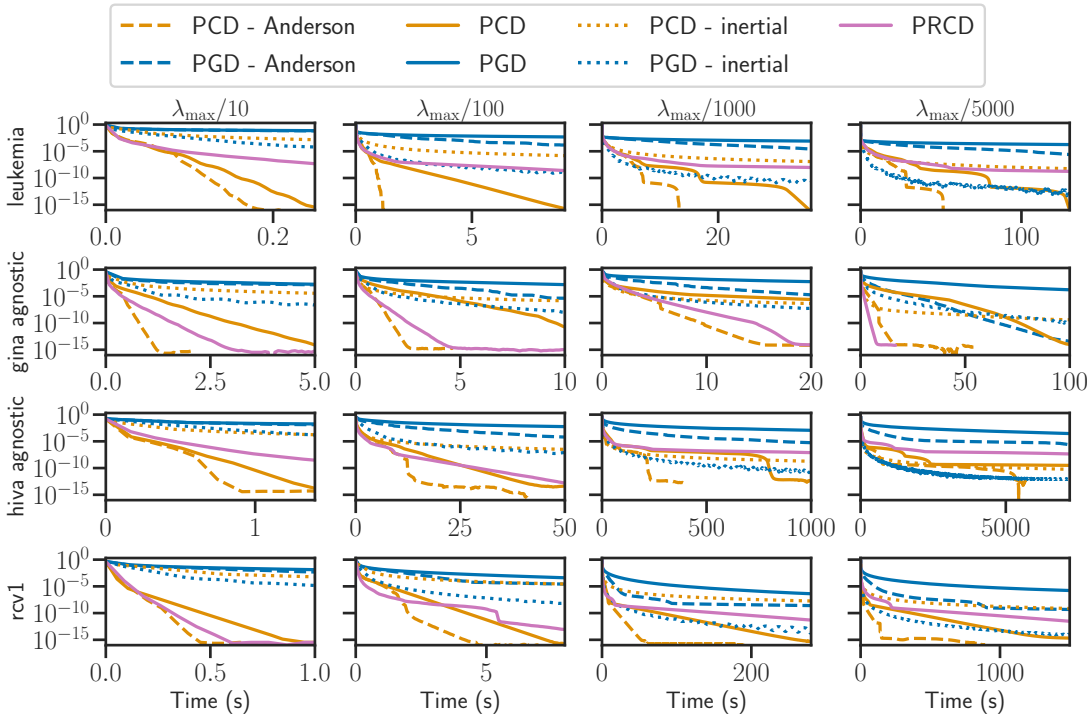


Figure 3.7 – **Lasso, suboptimality.** Suboptimality as a function of time for the Lasso on multiple datasets and values of λ.

We use datasets from `libsvm` (Fan et al., 2008) and openml (Feurer et al., 2019) (Table 3.1), varying as much as possible to demonstrate the versatility of our approach.

Table 3.1 – Datasets characteristics

name	n	p	density
gina agnostic	3468	970	1
hiva agnostic	4229	1617	1
leukemia	72	7129	1
rcv1_train	20 242	19 960	$3.7\,10^{-3}$
real-sim	72 309	20 958	$2.4\,10^{-3}$
news20	19 996	632 983	$6.1\,10^{-4}$

Lasso. Figure 3.7 shows the suboptimality $\Phi(\beta^{(k)}) - \Phi(\hat{\beta})$ of the algorithms on the Lasso problem:

$$\arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2}\|y - X\beta\|^2 + \lambda \left\|\beta\right\|_1 \quad , \tag{3.30}$$

as a function of time for multiple datasets and values of λ. We parametrize λ as a fraction of $\lambda_{\max} = \|X^\top y\|_\infty$, smallest regularization strength for which $\hat{\beta} = 0$. Figure 3.7 highlights the superiority of proximal coordinate descent over proximal gradient descent for Lasso problems on real-world datasets, and the benefits of extrapolation for coordinate descent. It shows that Anderson extrapolation can lead to a significant gain of performance. In particular Figure 3.7 shows that without restart, inertial coordinate

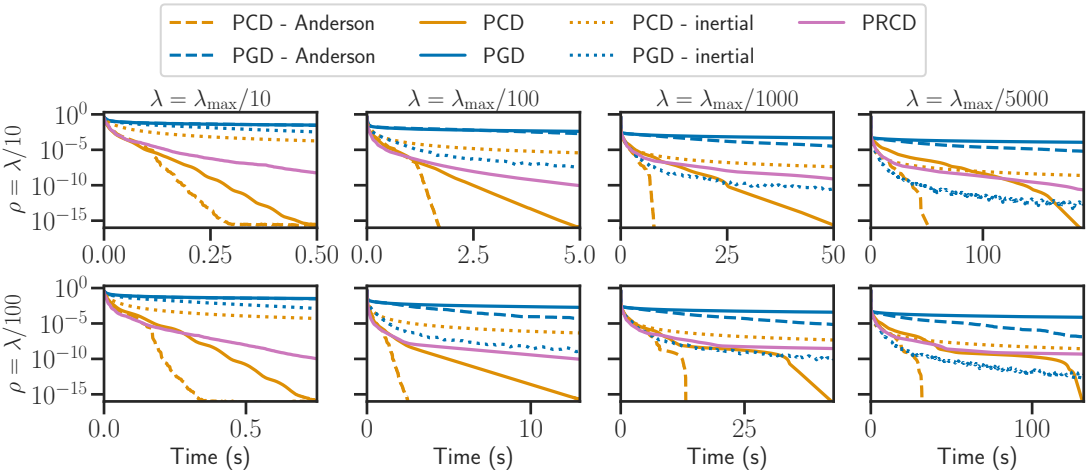


Figure 3.8 – **Enet, suboptimality.** Suboptimality as a function of time for the elastic net on Leukemia dataset, for multiple values of λ and ρ.

descent (Lin et al., 2014; Fercoq and Richtárik, 2015) can slow down the convergence, despite its accelerated rate. Note that the smaller the value of λ, the harder the optimization: when λ decreases, more time is needed to reach a fixed suboptimality. The smaller λ is (*i.e.,* the harder the problem), the more efficient Anderson extrapolation is.

Elastic net. Anderson extrapolation is easy to extend to other estimators than the Lasso. Figure 3.8 show the superiority of the Anderson extrapolation approach over proximal gradient descent and its accelerated version for the elastic net problem (Zou and Hastie, 2005):

$$\arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2}\|y - X\beta\|^2 + \lambda\|\beta\|_1 + \frac{\rho}{2}\|\beta\|_2^2 \ . \tag{3.31}$$

In particular, we observe that the more difficult the problem, the more useful the Anderson extrapolation: it is visible on Figure 3.8 that going from $\rho = \lambda/10$ to $\rho = \lambda/100$ lead to an increase in time to achieve similar suboptimality for the classical proximal coordinate descent, whereas the impact is more limited on the coordinate descent with Anderson extrapolation.

Finally, for a nonquadratic data-fit, here sparse logistic regression, we still demonstrate the applicability of extrapolated coordinate descent.

Sparse logistic regression. Figure 3.9 represents the suboptimality as a function of time on a sparse logistic regression problem:

$$\arg\min_{\beta \in \mathbb{R}^p} \sum_{i=1}^{n} \log(1 + e^{-y_i X_{i:}\beta}) + \lambda \left\|\beta\right\|_1 \ , \tag{3.32}$$

for multiple datasets and values of λ. We parametrize λ as a fraction of $\lambda_{\max} = \|X^\top y\|_\infty/2$. As for the Lasso and the elastic net, the smaller the value of λ, the harder the problem and Anderson CD outperforms its competitors.

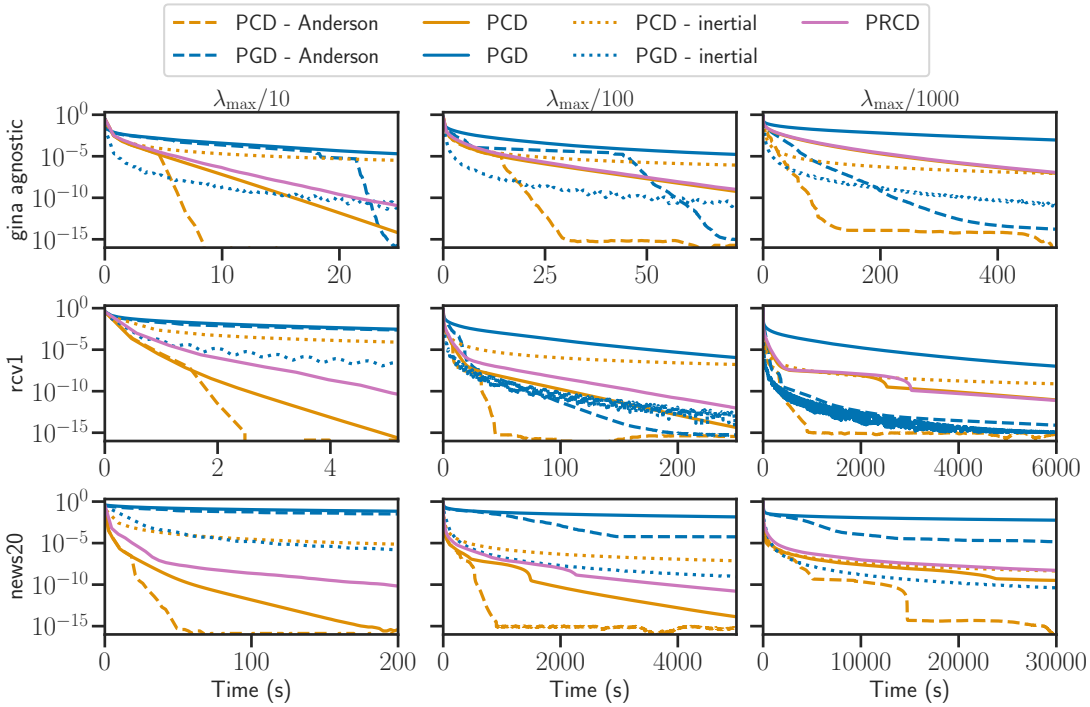


Figure 3.9 – **ℓ_1-regularised logistic regression, suboptimality.** Suboptimality as a function of time for ℓ_1-regularized logistic regression on multiple datasets and values of λ.

Group Lasso. In this section we consider the group Lasso, with a design matrix $X \in \mathbb{R}^{n \times p}$, a target $y \in \mathbb{R}^n$, and a partition $\mathcal{G}$ of $[p]$ (elements of the partition being the disjoints groups):

$$\arg\min_{\beta \in \mathbb{R}^p} \frac{1}{2}\|y - X\beta\|^2 + \lambda \sum_{g \in \mathcal{G}} \|\beta_g\| \ , \tag{3.33}$$

where for $g \in \mathcal{G}$, $\beta_g \in \mathbb{R}^{|g|}$ is the subvector of β composed of coordinates in g. the group Lasso can be solved via proximal gradient descent and by block coordinate descent (BCD), the latter being amenable to Anderson acceleration. As Figure 3.10 shows, the superiority of Anderson accelerated block coordinate descent is on par with the one observed on the problems studied above.

3.4 Conclusion

In this work, we have proposed to accelerate coordinate descent using Anderson extrapolation. We have exploited the fixed point iterations followed by coordinate descent iterates on multiple machine learning problems to improve their convergence speed. We have circumvented the non-symmetricity of the iteration matrices by proposing a pseudo-symmetric version for which accelerated convergence rates have been derived. In practice, we have performed an extensive validation to demonstrate large benefits on multiple datasets and problems of interests. For future works, the excellent performance of Anderson extrapolation for cyclic coordinate descent calls for a more refined analysis of the known bounds, through a better analysis of the spectrum and numerical range of the iteration matrices.

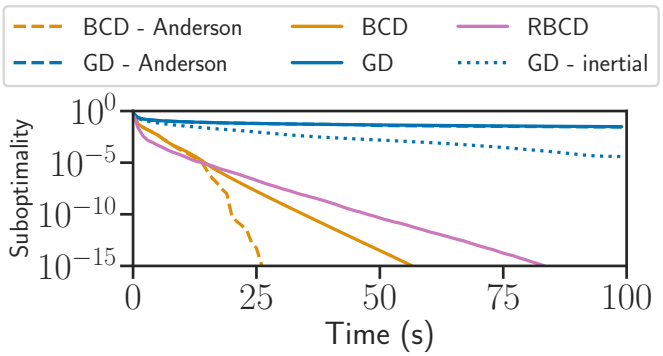


Figure 3.10 – **Group Lasso, suboptimality.** Suboptimality as a function of time for the group Lasso on the *Leukemia* dataset, $\lambda = \lambda_{\max}/100$. Groups are artificially taken as consecutive blocks of 5 features.

Part II

Hyperparameter selection, between statistics and optimization

4

On the statistical aspects of partial smoothing

Contents

In Chapter 3 we saw how to efficiently solve "smooth + nonsmooth separable" optimization problems. In this chapter we are interested in selecting the regularization parameter of Lasso-type problems, trading the data-fidelity term against the sparsity enforcing term. In high dimensional sparse regression, there exist estimators for which the optimal regularization parameter has a closed-form formula and is independent from the true noise level: pivotal estimators. The canonical pivotal estimator is the square-root Lasso, formulated along with its derivatives as a "nonsmooth + nonsmooth" optimization problem. Modern techniques to solve these include smoothing the data fitting term, to benefit from fast efficient proximal algorithms. In this chapter we show minimax sup-norm convergence rates for non smoothed and smoothed, single task and multitask square-root Lasso-type estimators. Thanks to our theoretical analysis, we provide some guidelines on how to set the smoothing hyperparameter, and illustrate on synthetic data the interest of such guidelines.

This chapter is based on the following work, accepted at AISTATS 2020:

- M. Massias, **Q. Bertrand**, A. Gramfort, and J. Salmon. Support recovery and sup-norm convergence rates for sparse pivotal estimation. In *AISTATS*, 2020a

4.1 Introduction

Since the mid 1990's and the development on the Lasso (Tibshirani, 1996), a vast literature has been devoted to sparse regularization for high dimensional regression. Statistical analysis of the Lasso showed that it achieves optimal rates (up to log factor, Bickel et al. 2009); see also Bühlmann and van de Geer (2011) for an extensive review. Yet, this estimator requires a specific calibration to achieve such an appealing rate: the regularization parameter must be proportional to the noise level. This quantity is generally unknown to the practitioner, hence the development of methods which are adaptive with respect to the noise level. An interesting candidate with such a property is the square-root Lasso ($\sqrt{\text{Lasso}}$, Belloni et al. 2011) defined for an observation vector $y \in \mathbb{R}^n$, a design matrix $X \in \mathbb{R}^{n \times p}$ and a regularization parameter λ by

$$\arg\min_{\beta \in \mathbb{R}^p} \frac{1}{\sqrt{n}} \|y - X\beta\|_2 + \lambda \|\beta\|_1 \ . \tag{4.1}$$

It has been shown to be *pivotal* with respect to the noise level by Belloni et al. (2011): the optimal regularization parameter of their analysis does not depend on the true noise level. This feature is also encountered in practice as illustrated by Figure 4.1 (see details on the framework in Section 4.4.1).

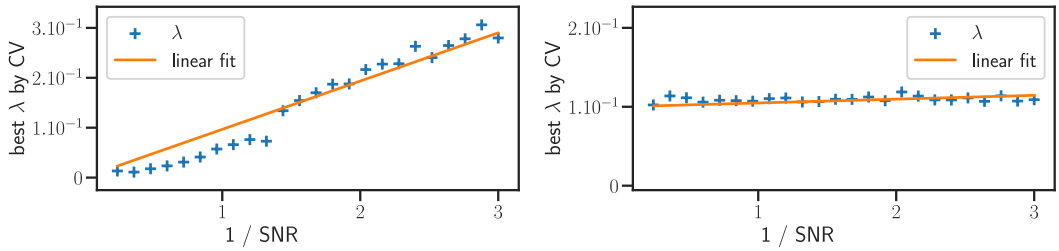


Figure 4.1 – Lasso (left) and square-root Lasso (right) optimal regularization parameters λ determined by cross validation on prediction error (blue), as a function of the noise level on simulated values of y. As indicated by theory, the Lasso's optimal λ grows linearly with the noise level, while it remains constant for the square-root Lasso.

Despite this theoretical benefit, solving the square-root Lasso requires tackling a "nonsmooth + nonsmooth" optimization problem. To do so, one can resort to conic programming (Belloni et al., 2011) or primal-dual algorithms (Chambolle and Pock, 2011) for which practical convergence may rely on hard-to-tune hyperparameters. Another approach is to use variational formulations of norms, *e.g.,* expressing the absolute value as $|x| = \min_{\sigma > 0} \frac{x^2}{2\sigma} + \frac{\sigma}{2}$ (Bach et al. 2012, Sec. 5.1, Micchelli et al. 2010). This leads to *concomitant estimation* (Huber and Dutter, 1974), that is, optimization problems over the regression parameters and an additional variable. In sparse regression, the seminal concomitant approach is the concomitant Lasso (Owen, 2007):

$$\arg\min_{\beta \in \mathbb{R}^p, \sigma > 0} \frac{1}{2n\sigma} \|y - X\beta\|_2^2 + \frac{\sigma}{2} + \lambda \left\|\beta\right\|_1 \ , \tag{4.2}$$

which yields the same estimate $\hat{\beta}$ as Problem (4.1) whenever $y - X\hat{\beta} \neq 0$. Problem (4.2) is more amenable: it is jointly convex, and the data fitting term is differentiable. Nevertheless, the data fitting term is still not smooth, as σ can approach 0 arbitrarily:

proximal solvers cannot be applied safely. A solution is to introduce a constraint $\sigma \geq \underline{\sigma}$ (Ndiaye et al., 2017), which amounts to *smoothing* (Nesterov, 2005; Beck and Teboulle, 2012) the square-root Lasso, *i.e.,* replacing its nonsmooth data fit by a smooth approximation (see details in Section 4.1.4).

There exist a straightforward way to generalize the square-root Lasso to the multitask setting (observations $Y \in \mathbb{R}^{n \times T}$): the multitask square-root Lasso,

$$\underset{\mathrm{B} \in \mathbb{R}^{p \times T}}{\arg\min} \frac{1}{\sqrt{nT}} \left\| Y - X\mathrm{B} \right\|_F + \lambda \left\| \mathrm{B} \right\|_{2,1} \quad , \tag{4.3}$$

where $\|\mathrm{B}\|_{2,1}$ is the ℓ_1 norm of the ℓ_2 norms of the rows. Another extension of the square-root Lasso to the multitask case is the multivariate square-root Lasso[1] (van de Geer, 2016, Sec. 3.8):

$$\underset{\mathrm{B} \in \mathbb{R}^{p \times T}}{\arg\min} \frac{1}{\sqrt{nT(n \wedge T)}} \left\| Y - X\mathrm{B} \right\|_* + \lambda \left\| \mathrm{B} \right\|_{2,1} \quad . \tag{4.4}$$

It is also shown by van de Geer (2016) that when $Y - X\hat{\mathrm{B}}$ is full rank, Problem (4.4) also admits a concomitant formulation, this time with an additional matrix variable:

$$\underset{\substack{\mathrm{B} \in \mathbb{R}^{p \times T} \\ S \succ 0}}{\arg\min} \frac{1}{2nT} \|Y - X\mathrm{B}\|_{S^{-1}}^2 + \frac{1}{2n} \operatorname{Tr}(S) + \lambda \left\| \mathrm{B} \right\|_{2,1} \quad . \tag{4.5}$$

In the analysis of the square-root Lasso (4.1), the non-differentiability at 0 can be avoided by excluding the corner case where the residuals $y - X\hat{\beta}$ vanish. However, analysis of the multivariate square-root Lasso through its concomitant formulation (4.5) has a clear weakness: it requires excluding rank deficient residuals cases, which is far from being a corner case. As illustrated in Figure 4.2, the full rank assumption made by van de Geer and Stucky (2016, Lemma 1) or Molstad (2019, Rem. 1) is not realistic, even for $T \geq n$ and high values of λ (see Section 4.4 for the setting's details). Motivated by numerical applications, Massias et al. (2018a) introduced a lower bound on the smallest eigenvalue of S ($S \succeq \underline{\sigma}\, \mathrm{Id}_n$) in Problem (4.5) to circumvent this issue. As we will see in Chapter 5, this amounts to smoothing the nuclear norm.

Our goal is to prove sup-norm convergence rates and support recovery guarantees for the estimators introduced above, and their smoothed counterparts.

Related works. The statistical properties of the Lasso have been studied under various frameworks and assumptions. Bickel et al. (2009) showed that with high probability, $\|X(\hat{\beta} - \beta^*)\|_2$ vanishes at the minimax rate (prediction convergence), whereas Lounici (2008) proved the sup-norm convergence and the support recovery of the Lasso (estimation convergence), *i.e.,* controlled the quantity $\|\hat{\beta} - \beta^*\|_\infty$. The latter result was extended to the multitask case by Lounici et al. (2011).

Since then, other Lasso-type estimators have been proposed and studied, such as the square-root Lasso (Belloni et al., 2011) or the scaled Lasso (Sun and Zhang, 2012). In the multitask case, Liu et al. (2015) introduced the Calibrated Multivariate Regression, and van de Geer and Stucky (2016); Molstad (2019) studied the multivariate square-root Lasso. These estimators have been proved to converge *in prediction.* However, apart

[1] modified here with a row-sparse penalty instead of ℓ_1

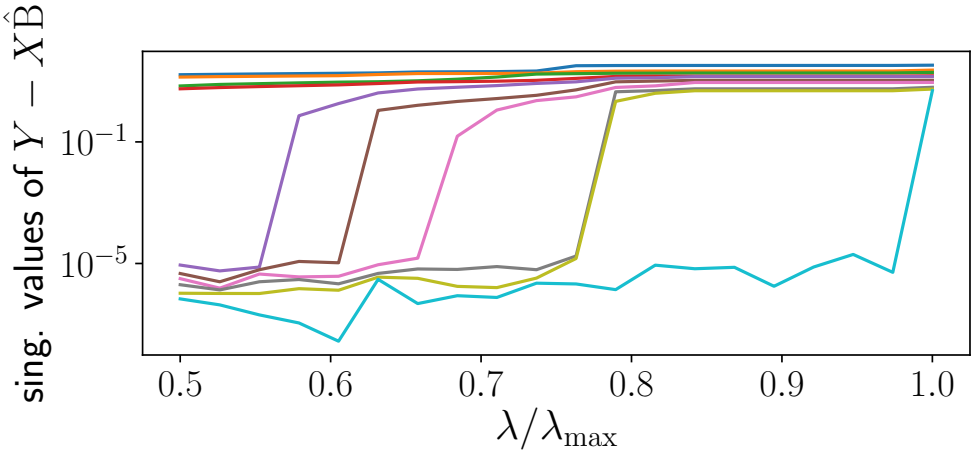


Figure 4.2 – Singular values of the residuals $Y - X\hat{\mathrm{B}}$ of the multivariate square-root Lasso ($n = 10, T = 20, p = 30$), as a function of λ. The observation matrix Y is full rank, but the residuals are rank deficient even for high values of the regularization parameter, invalidating the classical assumptions needed for statistical analysis.

from Bunea et al. (2014) for a particular group square-root Lasso, we are not aware of other works showing sup-norm convergence[2] of these estimators.

Within the framework introduced by Lounici (2008), our contributions are the following:

- We prove sup-norm convergence and support recovery of the multitask square-root Lasso and its smoothed version.
- We prove sup-norm convergence and support recovery of the *multivariate square-root Lasso* (van de Geer and Stucky, 2016, Sec. 2.2), and a smoothed version of it.
- Theoretical analysis leads to guidelines for the setting of the smoothing parameter $\underline{\sigma}$. In particular, as soon as $\underline{\sigma} \leq \sigma^*/\sqrt{2}$, the "optimal" λ and the sup-norm bounds obtained do not depend on $\underline{\sigma}$.
- We show on synthetic data the support recovery performance is little sensitive to the smoothing parameter $\underline{\sigma}$ as long as $\underline{\sigma} \leq \sigma^*/\sqrt{2}$.

Our contributions with respect to the existing literature are summarized in Table 4.1.

Notation. For any $\mathrm{B} \in \mathbb{R}^{p\times T}$ we define $\mathcal{S}(\mathrm{B}) \triangleq \{j \in [p] \,:\, ||\mathrm{B}_{j:}||_2 \neq 0\}$ the row-wise support of B. We write $\mathcal{S}_*$ for the row-wise support of the true coefficient matrix $\mathrm{B}^* \in \mathbb{R}^{p\times T}$. For any $\mathrm{B} \in \mathbb{R}^{p\times T}$ and any subset $\mathcal{S}$ of $[p]$ we denote $\mathrm{B}_{\mathcal{S}}$ the matrix in $\mathbb{R}^{p\times T}$ which has the same values as B on the rows with indices in $\mathcal{S}$ and vanishes on the complement $\mathcal{S}^c$. The estimated regression coefficients are written $\hat{\mathrm{B}}$, their difference with the true parameter B^* is noted $\Delta \triangleq \hat{\mathrm{B}} - \mathrm{B}^*$. The residuals at the optimum are noted $\hat{\mathrm{E}} \triangleq Y - X\hat{\mathrm{B}}$. For $a < b$, $[x]_a^b \triangleq \max(a, \min(x, b))$ is the clipping of x at levels a and b.

Model. Consider the multitask linear regression model:

$$Y = X\mathrm{B}^* + \mathrm{E} \ , \tag{4.6}$$

[2] of particular interest: combined with a large coefficients assumption, it implies support identification

where $Y \in \mathbb{R}^{n\times T}$, $X \in \mathbb{R}^{n\times p}$ is the deterministic design matrix, $\mathrm{B}^* \in \mathbb{R}^{p\times T}$ are the true regression coefficients and $\mathrm{E} \in \mathbb{R}^{n\times T}$ models a centered noise.

For an estimator $\hat{\mathrm{B}}$ of B^*, we aim at controlling $\|\hat{\mathrm{B}} - \mathrm{B}^*\|_{2,\infty}$ with high probability, and showing support recovery guarantees provided the non-zero coefficients are large enough. To prove such results, the following assumptions are classical: Gaussianity and independence of the noise, and *mutual incoherence*.

Assumption 4.1. *The entries of* E *are i.i.d.* $\mathcal{N}(0, \sigma^{*2})$ *random variables.*

Assumption 4.2 (Mutual incoherence)**.** *The* Gram matrix $\Psi \triangleq \frac{1}{n}X^\top X$ *satisfies*

$$\Psi_{jj} = 1 \ , \ \textit{and} \ \max_{j'\neq j} |\Psi_{jj'}| \leq \tfrac{1}{7\alpha s}, \forall j \in [p] \ , \tag{4.7}$$

for some integer $s \geq 1$ *and some constant* $\alpha > 1$.

Mutual incoherence of the design matrix (Assumption 4.2) implies the Restricted Eigenvalue Property introduced by Bickel et al. (2009).

Lemma 4.3 (Restricted Eigenvalue Property, Lounici 2008, Lemma 2)**.** *If Assumption 4.2 is satisfied, then:*

$$\min_{\substack{\mathcal{S}\subset[p]\\ |\mathcal{S}|\leq s}} \min_{\substack{\Delta\neq 0\\ \|\Delta_{\mathcal{S}^c}\|_{2,1}\leq 3\|\Delta_{\mathcal{S}}\|_{2,1}}} \frac{1}{\sqrt{n}}\frac{\|X\Delta\|_F}{\|\Delta_{\mathcal{S}}\|_F} \geq \sqrt{1-\tfrac{1}{\alpha}} > 0 \ . \tag{4.8}$$

In particular, with the choice $\Delta \triangleq \hat{\mathrm{B}} - \mathrm{B}^*$*, if* $\left\|\Delta_{\mathcal{S}_*^c}\right\|_{2,1} \leq 3\|\Delta_{\mathcal{S}_*}\|_{2,1}$*, the following bound holds:*

$$\frac{1}{n}\|X\Delta\|_F^2 \geq \left(1 - \frac{1}{\alpha}\right)\|\Delta_{\mathcal{S}_*}\|_F^2 \ . \tag{4.9}$$

4.1.1 Motivation and general proof structure

Structure of all proofs. We prove results of the following form for several estimators $\hat{\mathrm{B}}$ (summarized in Table 4.1): for some parameter λ independent of the noise level σ^*, with high probability,

$$\frac{1}{T}\|\hat{\mathrm{B}} - \mathrm{B}^*\|_{2,\infty} \leq C\frac{1}{\sqrt{nT}}\sqrt{\frac{\log p}{T}}\sigma^* \ . \tag{4.10}$$

Then, assuming a signal strong enough such that

$$\min_{j\in\mathcal{S}^*}\frac{1}{T}\|\mathrm{B}^*_{j:}\|_2 > 2C\frac{1}{\sqrt{nT}}\sqrt{\frac{\log p}{T}}\sigma^* \ , \tag{4.11}$$

on the same event, for some $\eta > 0$,

$$\hat{\mathcal{S}} \triangleq \{j \in [p] : \frac{1}{T}\|\hat{\mathrm{B}}_{j:}\|_2 > C(3+\eta)\lambda\sigma^*\} \tag{4.12}$$

matches the true sparsity pattern: $\hat{\mathcal{S}} = \mathcal{S}^*$.

We explain here the general sketch proofs for all the estimators. We assume that Assumption 4.2 holds and then place ourselves on an event $\mathcal{A}$ such that $\|X^\top Z\|_{2,\infty} \leq \lambda/2$ (for a $Z \in \partial f(\mathrm{E})$, where f is the data fitting term) in order to use Lemma 4.5 (b), which links the control of $\|\Psi(\hat{\mathrm{B}} - \mathrm{B}^*)\|_{2,\infty}$ to the control of $\|\hat{\mathrm{B}} - \mathrm{B}^*\|_{2,\infty}$. To obtain sup-norm convergence it remains for each estimator to:

Table 4.1 – Summary of estimators (MT: multitask, MV: multivariate)

Name	$f(\mathrm{E})$	Sup-norm cvg	Pred. cvg
MT $\sqrt{\text{Lasso}}$ (4.3)	$\frac{1}{\sqrt{nT}}\|\mathrm{E}\|_F$	Bunea et al. (2014)	Bunea et al. (2014)
MT concomitant Lasso	$\min_{\sigma>0} \frac{1}{2nT\sigma}\|\mathrm{E}\|_F^2 + \frac{\sigma}{2}$	us	Li et al. (2016)
MT smooth. conco. Lasso (4.26)	$\min_{\sigma>\underline{\sigma}} \frac{1}{2nT\sigma}\|\mathrm{E}\|_F^2 + \frac{\sigma}{2}$	us	Li et al. (2016)
MV $\sqrt{\text{Lasso}}$ (4.4)	$\frac{1}{n}\|\mathrm{E}/\sqrt{T}\|_*$	us	Molstad (2019)
MV conco. $\sqrt{\text{Lasso}}$ (4.5)	$\min_{S\succ 0} \frac{1}{2nT}\|\mathrm{E}\|_{S^{-1}}^2 + \frac{1}{2n}\operatorname{Tr}(S)$	us	Molstad (2019)
MV SGCL (4.45)	$\min_{\bar{\sigma}\succeq S\succeq \underline{\sigma}} \frac{1}{2nT}\|\mathrm{E}\|_{S^{-1}}^2 + \frac{1}{2n}\operatorname{Tr}(S)$	us	

- control the probability of the event $\mathcal{A}$ with classical concentration inequalities.
- control the quantity $\|\Psi(\hat{\mathrm{B}} - \mathrm{B}^*)\|_{2,\infty}$, with:
  - first order optimality conditions, which provide a bound on $\|X^\top Z\|_{2,\infty}$: $\|X^\top \hat{Z}\|_{2,\infty} \leq \lambda$ for a $\hat{Z} \in \partial f(\hat{\mathrm{E}})$,
  - the definition of the event $\mathcal{A}$,
  - for some estimators, an additional assumption (Assumption 4.8).

Next, we detail the lemmas used in this strategy.

4.1.2 Preliminary lemma

First we show that if Assumption 4.2 holds, then a bound "in prediction", on $\|\Psi\Delta\|_{2,\infty}$, leads to a bound "in estimation", on $\|\Delta\|_{2,\infty}$.

Lemma 4.4. *Let Ψ, α and s satisfy Assumption 4.2, let $\hat{\mathrm{B}}$ be an estimator satisfying: $\|\Delta_{\mathcal{S}_*^c}\|_{2,1} \leq 3\|\Delta_{\mathcal{S}_*}\|_{2,1}$, then:*

a) $\|\Delta_{\mathcal{S}_*}\|_F \leq \frac{\alpha}{\alpha-1} 4\sqrt{s}\|\Psi\Delta\|_{2,\infty}$,

b) $\|\Delta\|_{2,1} \leq \frac{\alpha}{\alpha-1} 16s\|\Psi\Delta\|_{2,\infty}$,

c) $\|\Delta\|_{2,\infty} \leq \left(1 + \frac{16}{7(\alpha-1)}\right)\|\Psi\Delta\|_{2,\infty}$.

Proof For Lemma 4.4 (a), the idea is to upper and lower bound $\frac{1}{n}\|X\Delta\|_F^2$. First we bound $\|\Delta\|_{2,1}$:

$$\begin{aligned}\|\Delta\|_{2,1} &= \|\Delta_{S_*^c}\|_{2,1} + \|\Delta_{S_*}\|_{2,1} \\ &\leq 4\|\Delta_{S_*}\|_{2,1} \\ &\leq 4\sqrt{s}\|\Delta_{S_*}\|_F \ . \end{aligned} \tag{4.13}$$

Now we can upper bound $\frac{1}{n}\|X\Delta\|_F^2$ with Hölder inequality and Equation (4.13):

$$\begin{aligned}\tfrac{1}{n}\|X\Delta\|_F^2 &= \langle \Delta, \Psi\Delta\rangle \\ &\leq \|\Delta\|_{2,1}\|\Psi\Delta\|_{2,\infty} \\ &\leq 4\sqrt{s}\|\Delta_{S_*}\|_F\|\Psi\Delta\|_{2,\infty} \ . \end{aligned} \tag{4.14}$$

By Equation (4.9) and Equation (4.14):

$$\begin{aligned}(1-\tfrac{1}{\alpha})\|\Delta_{S_*}\|_F^2 &\leq \tfrac{1}{n}\|X\Delta\|_F^2 \\ &\leq 4\sqrt{s}\|\Delta_{S_*}\|_F\|\Psi\Delta\|_{2,\infty} \\ \|\Delta_{S_*}\|_F &\leq \frac{\alpha}{\alpha-1}4\sqrt{s}\|\Psi\Delta\|_{2,\infty} \ . \end{aligned} \tag{4.15}$$

Lemma 4.4 (b) is a direct consequence of Equation (4.13) and lemma 4.4 (a):

$$\begin{aligned}\|\Delta\|_{2,1} &\leq 4\sqrt{s}\|\Delta_{S_*}\|_F \\ &\leq \frac{\alpha}{\alpha-1}16s\|\Psi\Delta\|_{2,\infty} \ . \end{aligned} \tag{4.16}$$

Finally, for Lemma 4.4 (c), for any $j \in [p]$,

$$\begin{aligned}\left(\Psi\Delta\right)_{j:} &= \Delta_{j:} + \textstyle\sum_{j'\neq j}\Psi_{j'j}\Delta_{j':} \\ ||\left(\Psi\Delta\right)_{j:} - \Delta_{j:}||_2 &\leq \textstyle\sum_{j'\neq j}|\Psi_{jj'}| \times ||\Delta_{j':}||_2 \\ ||\left(\Psi\Delta\right)_{j:} - \Delta_{j:}||_2 &\leq \frac{1}{7\alpha s}\textstyle\sum_{j'\neq j}\|\Delta_{j':}\| \\ \|\Delta\|_{2,\infty} &\leq \|\Psi\Delta\|_{2,\infty} + \tfrac{1}{7\alpha s}\|\Delta\|_{2,1} \\ &\leq \left(1+\tfrac{16}{7(\alpha-1)}\right)\|\Psi\Delta\|_{2,\infty} \ . \end{aligned} \tag{4.17}$$

using Assumption 4.2 and lemma 4.4 (b). ■

We now provide conditions leading to $\|\Delta_{\mathcal{S}_*^c}\|_{2,1} \leq 3\|\Delta_{\mathcal{S}_*}\|_{2,1}$, to be able to apply Lemma 4.3. In this section we consider estimators of the form

$$\hat{\mathrm{B}} \triangleq \underset{\mathrm{B}\in\mathbb{R}^{p\times T}}{\arg\min}\, f(Y - X\mathrm{B}) + \lambda\|\mathrm{B}\|_{2,1} \ , \tag{4.18}$$

for a proper, lower semi-continuous and convex function $f : \mathbb{R}^{n\times T} \to \mathbb{R}$ (see the summary in Table 4.1). Fermat's rule for Problem (4.18) reads:

$$0 \in X^\top \partial f(\hat{\mathrm{E}}) + \lambda\partial\|\cdot\|_{2,1}(\hat{\mathrm{B}}) \ , \tag{4.19}$$

Hence, we can find $\hat{Z} \in \partial f(\hat{\mathrm{E}})$ such that

$$\|X^\top \hat{Z}\|_{2,\infty} \leq \lambda \ . \tag{4.20}$$

Lemma 4.5. *Consider an estimator based on Problem (4.18), and assume that there exists $Z \in \partial f(\mathrm{E})$ such that $\|X^\top Z\|_{2,\infty} \leq \lambda/2$. Then:*

a) $\|\Delta_{\mathcal{S}_*^c}\|_{2,1} \leq 3\|\Delta_{\mathcal{S}_*}\|_{2,1}$,

b) if Ψ and α satisfy Assumption 4.2,

$$\|\Delta\|_{2,\infty} \leq \left(1+\frac{16}{7(\alpha-1)}\right)\|\Psi\Delta\|_{2,\infty} \ .$$

Proof For Lemma 4.5 (a), we use the minimality of $\hat{\mathrm{B}}$:

$$f(\hat{\mathrm{E}}) - f(\mathrm{E}) \leq \lambda\|\mathrm{B}^*\|_{2,1} - \lambda\|\hat{\mathrm{B}}\|_{2,1} \ . \tag{4.21}$$

We upper bound the right hand side of Equation (4.21), using $\|\hat{\mathrm{B}}\|_{2,1} = \|\hat{\mathrm{B}}_{\mathcal{S}_*}\|_{2,1} + \|\hat{\mathrm{B}}_{\mathcal{S}_*^c}\|_{2,1}$, $\mathrm{B}^*_{\mathcal{S}_*^c} = 0$ and with the triangle inequality:

$$\begin{aligned}\|\mathrm{B}^*\|_{2,1} - \|\hat{\mathrm{B}}\|_{2,1} &= \|\mathrm{B}^*_{\mathcal{S}_*}\|_{2,1} - \|\hat{\mathrm{B}}_{\mathcal{S}_*}\|_{2,1} - \|\hat{\mathrm{B}}_{\mathcal{S}_*^c}\|_{2,1} \\ &= \|\mathrm{B}^*_{\mathcal{S}_*}\|_{2,1} - \|\hat{\mathrm{B}}_{\mathcal{S}_*}\|_{2,1} - \|\Delta_{\mathcal{S}_*^c}\|_{2,1} \\ &\leq \|(\mathrm{B}^* - \hat{\mathrm{B}})_{\mathcal{S}_*}\|_{2,1} - \|\Delta_{\mathcal{S}_*^c}\|_{2,1} \\ &\leq \|\Delta_{\mathcal{S}_*}\|_{2,1} - \|\Delta_{\mathcal{S}_*^c}\|_{2,1} \ . \end{aligned} \tag{4.22}$$

We now aim at finding a lower bound of the left hand side of Equation (4.21). By convexity of f, $\partial f(\mathrm{E}) \neq \varnothing$. Picking $Z \in \partial f(\mathrm{E})$ such that $\|X^\top Z\|_{2,\infty} \leq \frac{\lambda}{2}$ yields:

$$\begin{aligned} f(Y - X\hat{\mathrm{B}}) - f(Y - X\mathrm{B}^*) &\geq -\left\langle Z, X(\hat{\mathrm{B}} - \mathrm{B}^*)\right\rangle \\ &\geq -\left\langle X^\top Z, \Delta\right\rangle \\ &\geq -\|X^\top Z\|_{2,\infty}\|\Delta\|_{2,1} \\ &\geq -\frac{1}{2}\lambda\|\Delta\|_{2,1} \ . \end{aligned}$$

Combining Equations (4.21) to (4.23) leads to:

$$\begin{aligned} -\frac{1}{2}\|\Delta\|_{2,1} &\leq \|\Delta_{\mathcal{S}_*}\|_{2,1} - \|\Delta_{\mathcal{S}_*^c}\|_{2,1} \\ \|\Delta_{\mathcal{S}_*^c}\|_{2,1} &\leq 3\|\Delta_{\mathcal{S}_*}\|_{2,1} \ . \end{aligned} \tag{4.23}$$

Proof of Lemma 4.5 (b) is a direct application of Lemmas 4.5 (a) and 4.4 (c). ■

Equipped with these Assumptions and Lemmas, we will show that the considered estimators reach the minimimax lower bounds, which we recall in the following.

4.1.3 Minimax lower bounds

As said in Section 4.1.1, our goal is to provide convergence rates on the quantity $\|\hat{\mathrm{B}} - \mathrm{B}^*\|_{2,\infty}$. To show that our bounds are "optimal" we recall that the considered estimators achieve minimax rate (up to a logarithmic factor). Indeed, under some additional assumptions controlling the conditioning of the design matrix, one can show (Lounici et al., 2011) minimax lower bounds.

Assumption 4.6. *For all* $\Delta \in \mathbb{R}^{p\times T}\backslash\{0\}$ *such that* $|\mathcal{S}(\Delta)| \leq 2|\mathcal{S}^*|$*:*

$$\underline{\kappa} \leq \frac{\|X\Delta\|_F^2}{n\|\Delta\|_F^2} \leq \bar{\kappa} \ . \tag{4.24}$$

Provided Assumptions 4.1 and 4.6 hold true, Lounici et al. (2011, Thm. 6.1) proved the following minimax lower bound (with an absolute constant R):

$$\inf_{\hat{\mathrm{B}}} \sup_{\substack{\mathrm{B}^* \, s.t. \\ |\mathcal{S}(\mathrm{B}^*)|\leq s}} \mathbb{E}\left(\frac{1}{T}\|\hat{\mathrm{B}} - \mathrm{B}^*\|_{2,\infty}\right) \geq \frac{R\sigma^*}{\bar{\kappa}\sqrt{n}}\sqrt{1 + \frac{\log(ep/s)}{T}} \ .$$

4.1.4 Smoothing

Some of the pivotal estimators studied here are obtained via a technique called smoothing. For $L > 0$, a convex function ϕ is L-smooth (*i.e.,* its gradient is L-Lipschitz) if and only if its Fenchel conjugate ϕ^* is $\frac{1}{L}$-strongly convex (Hiriart-Urruty and Lemaréchal, 1993, Thm 4.2.1). Therefore, given a smooth function ω, a principled way to smooth a function f is to add the strongly convex ω^* to f^*, thus creating a strongly convex function, whose Fenchel transform is a smooth approximation of f. Formally, given a smooth convex function ω, the ω-smoothing of f is $(f^* + \omega^*)^*$. By properties of the Fenchel transform, the latter is also equal to $f \,\square\, \omega$ whenever f is convex (Bauschke and Combettes, 2011, Prop. 13.21).

Proposition 4.7. *Let* $\omega_{\underline{\sigma}} = \frac{1}{2\underline{\sigma}}\|\cdot\|_F^2 + \frac{\underline{\sigma}}{2}$. *The* $\omega_{\underline{\sigma}}$*-smoothing of the Frobenius norm is equal to:*

$$\left(\omega_{\underline{\sigma}} \,\square \|\cdot\|_F\right)(Z) = \begin{cases} \|Z\|_F \;, & \text{if } \|Z\|_F \leq \underline{\sigma} \;, \\ \frac{1}{2\underline{\sigma}}\|Z\|_F^2 + \frac{\underline{\sigma}}{2} \;, & \text{if } \|Z\|_F \geq \underline{\sigma} \;. \end{cases}$$
$$= \min_{\sigma \geq \underline{\sigma}} \frac{1}{2\sigma}\|Z\|_F^2 + \frac{\sigma}{2} \;. \tag{4.25}$$

4.2 Multitask square-root Lasso

It is clear that the multitask square-root Lasso (Problem (4.3)) suffers from the same numerical weaknesses as the square-root Lasso. A more amenable version is introduced in Chapter 5. The smoothed multitask square-root Lasso is obtained by replacing the nonsmooth function $\|\cdot\|_F$ with a smooth approximation, depending on a parameter $\underline{\sigma} > 0$:

$$\operatorname*{arg\,min}_{\mathrm{B} \in \mathbb{R}^{p \times T}} \left(\|\cdot\|_F \,\square \left(\tfrac{1}{2\underline{\sigma}}\|\cdot\|^2 + \tfrac{\underline{\sigma}}{2}\right)\right)\left(\tfrac{Y - X\mathrm{B}}{\sqrt{nT}}\right) + \lambda \left\|\mathrm{B}\right\|_{2,1} \;. \tag{4.26}$$

Plugging the expression of the smoothed Frobenius norm (4.25), the problem formulation becomes:

$$(\hat{\mathrm{B}}, \hat{\sigma}) \in \operatorname*{arg\,min}_{\substack{\mathrm{B} \in \mathbb{R}^{p \times T} \\ \sigma \geq \underline{\sigma}}} \frac{1}{2nT\sigma}\|Y - X\mathrm{B}\|_F^2 + \frac{\sigma}{2} + \lambda \left\|\mathrm{B}\right\|_{2,1} \;, \tag{4.27}$$

where the data fitting term is $(nT\underline{\sigma})^{-1}$-smooth with respect to B. We show that estimators (4.3) and (4.26) reach the minimax lower bound, with a regularization parameter independent of σ^*. For that, another assumption is needed.

Assumption 4.8 (van de Geer 2016, Lemma 3.1)**.** *There exists* $\eta > 0$ *verifying*

$$\lambda \|\mathrm{B}^*\|_{2,1} \leq \eta \sigma^* \;. \tag{4.28}$$

Proposition 4.9. *Let* $\hat{\mathrm{B}}$ *denote the multitask square-root Lasso* (4.3) *or its smoothed version* (4.26)*. Let Assumption 4.1 be satisfied, let* α *and* η *satisfy Assumptions 4.2 and 4.8. For* $C = \left(1 + \frac{16}{7(\alpha - 1)}\right)$, $A > \sqrt{2}$ *and* $\lambda = \frac{2\sqrt{2}}{\sqrt{nT}}\left(1 + A\sqrt{(\log p)/T}\right)$, *if* $\underline{\sigma} \leq \frac{\sigma^*}{\sqrt{2}}$ *then with probability at least* $1 - p^{1 - A^2/2} - (1 + e^2)e^{-nT/24}$,

$$\frac{1}{T}\|\hat{\mathrm{B}} - \mathrm{B}^*\|_{2,\infty} \leq C(3 + \eta)\lambda\sigma^* \;. \tag{4.29}$$

Moreover provided that

$$\min_{j\in\mathcal{S}^*}\frac{1}{T}\|\mathrm{B}^*_{j:}\|_2 > 2C(3+\eta)\lambda\sigma^* \ , \tag{4.30}$$

then, with the same probability, the estimated support

$$\hat{\mathcal{S}} \triangleq \{j\in[p] : \frac{1}{T}\|\hat{\mathrm{B}}_{j:}\|_2 > C(3+\eta)\lambda\sigma^*\} \tag{4.31}$$

recovers the true sparsity pattern: $\hat{\mathcal{S}} = \mathcal{S}^*$.

Proof We first bound $\|\Psi\Delta\|_{2,\infty}$. Let $\mathcal{A}_1$ be the event

$$\mathcal{A}_1 \triangleq \left\{\frac{\|X^\top \mathrm{E}\|_{2,\infty}}{\sqrt{nT}\|\mathrm{E}\|_F} \leq \frac{\lambda}{2}\right\} \cap \left\{\frac{\sigma^*}{\sqrt{2}} < \frac{\|\mathrm{E}\|_F}{\sqrt{nT}} < 2\sigma^*\right\} \ . \tag{4.32}$$

By Lemma 4.16 (h), $\mathbb{P}(\mathcal{A}_1) \geq 1 - p^{1-A^2/2} - (1+e^2)e^{-nT/24}$. For both estimators, on $\mathcal{A}_1$ we have:

$$\begin{aligned} n\|\Psi\Delta\|_{2,\infty} &= \|X^\top(\hat{\mathrm{E}} - \mathrm{E})\|_{2,\infty} \\ &\leq \|X^\top\hat{\mathrm{E}}\|_{2,\infty} + \|X^\top\mathrm{E}\|_{2,\infty} \\ &\leq \|X^\top\hat{\mathrm{E}}\|_{2,\infty} + \lambda nT\sigma^* \ , \end{aligned} \tag{4.33}$$

hence we need to bound $\|X^\top\hat{\mathrm{E}}\|_{2,\infty}$. We do so using optimality conditions, that yield for Problem (4.3), with $\hat{\mathrm{E}} \neq 0$,

$$\begin{aligned} \|X^\top \tfrac{\hat{\mathrm{E}}}{\|\hat{\mathrm{E}}\|_F}\|_{2,\infty} &\leq \lambda\sqrt{nT} \\ \tfrac{1}{nT}\|X^\top\hat{\mathrm{E}}\|_{2,\infty} &\leq \lambda\tfrac{\|\hat{\mathrm{E}}\|_F}{\sqrt{nT}} \ , \end{aligned} \tag{4.34}$$

and the last equation is still valid if $\hat{\mathrm{E}} = 0$. For Problem (4.26), the optimality conditions yield:

$$\begin{cases} \frac{1}{nT}\|X^\top\hat{\mathrm{E}}\|_{2,\infty} \leq \lambda\frac{\|\hat{\mathrm{E}}\|_F}{\sqrt{nT}} \ , & \text{if } \frac{\|\hat{\mathrm{E}}\|_F}{\sqrt{nT}} \geq \underline{\sigma} \ , \\ \frac{1}{nT}\|X^\top\hat{\mathrm{E}}\|_{2,\infty} \leq \lambda\underline{\sigma} \ , & \text{otherwise} \ . \end{cases} \tag{4.35}$$

Therefore,

$$\tfrac{1}{nT}\|X^\top\hat{\mathrm{E}}\|_{2,\infty} \leq \lambda\max\left(\tfrac{\|\hat{\mathrm{E}}\|_F}{\sqrt{nT}}, \underline{\sigma}\right) \ . \tag{4.36}$$

It now remains to bound $\|\hat{\mathrm{E}}\|_F$ for both estimators, which is done with Assumption 4.8: for Problem (4.3), by minimality of the estimator,

$$\begin{aligned} \tfrac{1}{\sqrt{nT}}\|\hat{\mathrm{E}}\|_F + \lambda\|\hat{\mathrm{B}}\|_{2,1} &\leq \tfrac{1}{\sqrt{nT}}\|\mathrm{E}\|_F + \lambda\|\mathrm{B}^*\|_{2,1} \\ \tfrac{1}{\sqrt{nT}}\|\hat{\mathrm{E}}\|_F &\leq \tfrac{1}{\sqrt{nT}}\|\mathrm{E}\|_F + \lambda\|\mathrm{B}^*\|_{2,1} \\ &\leq 2\sigma^* + (1+\eta)\sigma^* \\ &\leq (3+\eta)\sigma^* \ , \end{aligned} \tag{4.37}$$

and we can obtain the same bound in the case of Problem (4.26): by the minimality of the estimator we have on $\mathcal{A}_1$:

$$\left(\|\cdot\|_F \square \left(\tfrac{1}{2\underline{\sigma}} + \tfrac{\underline{\sigma}}{2}\right)\|\cdot\|_F^2\right)(\tfrac{\hat{\mathrm{E}}}{\sqrt{nT}}) + \lambda\|\hat{\mathrm{B}}\|_{2,1} \leq \left(\|\cdot\|_F \square \left(\tfrac{1}{2\underline{\sigma}}\|\cdot\|_F^2 + \tfrac{\underline{\sigma}}{2}\right)\right)(\tfrac{\mathrm{E}}{\sqrt{nT}}) + \lambda\|\mathrm{B}^*\|_{2,1}$$

$$\frac{1}{\sqrt{nT}}\|\hat{\mathrm{E}}\|_F + \lambda\|\hat{\mathrm{B}}\|_{2,1} \leq \left(\|\cdot\|_F \,\square\, \left(\frac{1}{2\underline{\sigma}}\|\cdot\|_F^2 + \frac{\underline{\sigma}}{2}\right)\left(\frac{\mathrm{E}}{\sqrt{nT}}\right)\right) + \lambda\|\mathrm{B}^*\|_{2,1}$$

$$\text{since } \|\cdot\|_F \leq \left(\|\cdot\|_F \,\square\, \left(\frac{1}{2\underline{\sigma}}\|\cdot\|_F^2 + \frac{\underline{\sigma}}{2}\right)\right)$$

$$\begin{aligned}
\frac{1}{\sqrt{nT}}\|\hat{\mathrm{E}}\|_F &\leq \frac{1}{\sqrt{nT}}\|\mathrm{E}\|_F + \lambda\|\mathrm{B}^*\|_{2,1} && \text{since } \frac{1}{\sqrt{nT}}\|\mathrm{E}\|_F \geq \frac{\sigma^*}{\sqrt{2}} \geq \underline{\sigma} \\
&\leq 2\sigma^* + \lambda\|\mathrm{B}^*\|_{2,1} && \text{since } \frac{1}{\sqrt{nT}}\|\mathrm{E}\|_F \leq 2\sigma^* \\
&\leq 2\sigma^* + (1+\eta)\sigma^* && \text{since } \lambda\|\mathrm{B}^*\|_{2,1} \leq (1+\eta)\sigma^* \\
\frac{1}{\sqrt{nT}}\|\hat{\mathrm{E}}\|_F &\leq (3+\eta)\sigma^* \ .
\end{aligned}$$

Combining Equations (4.33), (4.34), (4.36) and (4.37) we have in both cases:

$$\frac{1}{T}\|\Psi\Delta\|_{2,\infty} \leq (3+\eta)\lambda\sigma^* \ . \tag{4.38}$$

Finally we exhibit an element of $\partial f(\mathrm{E})$ to apply Lemma 4.5 (b). Recall that $f = \frac{1}{\sqrt{nT}}\|\cdot\|_F$ for Problem (4.3), and $f = \|\cdot\|_F \,\square\, \left(\frac{1}{2\underline{\sigma}}\|\cdot\|_F^2 + \frac{\underline{\sigma}}{2}\right)\left(\frac{\cdot}{\sqrt{nT}}\right)$ for Problem (4.26). On $\mathcal{A}_1$, $\partial f(\mathrm{E})$ is a singleton for both estimators, whose element is $\mathrm{E}/(\|\mathrm{E}\|_F\sqrt{nT})$.

Additionally, on $\mathcal{A}_1$ the inequality $\frac{1}{\sqrt{nT}}\frac{\|X^\top \mathrm{E}\|_{2,\infty}}{\|\mathrm{E}\|_F} \leq \frac{\lambda}{2}$ holds, meaning we can apply Lemma 4.5 (b) with $Z = \mathrm{E}/(\|\mathrm{E}\|_F\sqrt{nT})$. This proves the bound on $\|\Delta\|_{2,\infty}$. Then, the support recovery property easily follows from Lounici et al. (2009, Cor. 4.1). ■

Single task case. For the purpose of generality, we proved convergence results for the multitask versions of the square-root/concomitant Lasso and its smoothed version, but the results are also interesting in the single task setting: in this case ($T = 1$) it is possible to achieve tighter convergence rates.

Proposition 4.10. *Let Assumption 4.1 be satisfied, let α satisfy Assumption 4.2 and let η satisfy Assumption 4.8. Let $C = 2\left(1 + \frac{16}{7(\alpha-1)}\right)$ and*

$$\lambda = A\sqrt{2\log p/n} \ . \tag{4.39}$$

Then with probability at least $1 - p^{1-A^2/8} - (1+e^2)e^{-n/24}$,

$$\frac{1}{T}\|\hat{\beta} - \beta^*\|_{2,\infty} \leq C(2+\eta)\lambda\sigma \ . \tag{4.40}$$

Moreover if

$$\min_{j\in\mathcal{S}^*} |\beta_j^*| > 2C(2+\eta)\lambda\sigma \ , \tag{4.41}$$

then with the same probability:

$$\hat{\mathcal{S}} = \{j \in [p] : |\hat{\beta}_j| > C(2+\eta)\lambda\sigma\} \tag{4.42}$$

estimate correctly the true sparsity pattern:

$$\hat{\mathcal{S}} = \mathcal{S}^* \ . \tag{4.43}$$

Proof All the inequalities leading to Equation (4.38) still hold. The control of the event $\mathcal{A}_1$ can be tighter in the single-task case. Since $\mathcal{A}_1 = \left\{\frac{1}{n}\|X^\top\varepsilon\|_\infty \leq \frac{\lambda}{2}\frac{\sigma^*}{\sqrt{2}}\right\} \cap \left\{\frac{\sigma^*}{\sqrt{2}} < \frac{\|\varepsilon\|_F}{\sqrt{nT}} < 2\sigma^*\right\}$, with $\lambda = A\sqrt{2\log p/n}$, concentration in equalities from Lemmas 4.16 (b) and 4.16 (e) in Section 4.A leads to:

$$\mathcal{P}(\mathcal{A}_1) \geq 1 - p^{1-A^2/8} - (1+e^2)e^{-n/24} \ . \tag{4.44}$$

■

4.3 Multivariate square-root Lasso

Here we show that the multivariate square-root Lasso[3] and its smoothed version also reach the minimax rate. Recall that the multivariate square-root Lasso is Problem (4.4). For the numerical reasons mentioned above, as well as to get rid of the invertibility assumption of $\hat{\mathrm{E}}^\top\hat{\mathrm{E}}$, we consider the smoothed estimator of Massias et al. (2018a):

$$\underset{\substack{\mathrm{B}\in\mathbb{R}^{p\times T}\\ \bar{\sigma}\,\mathrm{Id}_n \succeq S \succeq \underline{\sigma}\,\mathrm{Id}_n}}{\arg\min} \ \frac{1}{2nT}\|Y - X\mathrm{B}\|^2_{S^{-1}} + \frac{\mathrm{Tr}\,S}{2n} + \lambda\left\|\mathrm{B}\right\|_{2,1} \ . \tag{4.45}$$

The variable introduced by concomitant formulation is now a matrix S, corresponding to the square root of the noise covariance estimate. The multivariate square-root Lasso (4.4) and its concomitant formulation (4.5) have the same solution in B provided $\hat{\mathrm{E}}^\top\hat{\mathrm{E}}$ is invertible. In this case, the solution of Problem (4.5) in S is $\hat{S} = (\frac{1}{T}\hat{\mathrm{E}}\hat{\mathrm{E}}^\top)^{\frac{1}{2}}$.

Problem (4.45) is actually a small modification of Massias et al. (2018a), where we have added the second constraint $S \preceq \bar{\sigma}\,\mathrm{Id}_n$. $\bar{\sigma}$ can for example be set as $\|(\frac{1}{T}YY^\top)^{1/2}\|_2$, as Figure 4.2 illustrates that this is the order of magnitude of $\|\hat{S}\|_2$. Because of these constraints, the solution in S is different from that of Problem (4.5). We write a singular value decomposition of $\frac{1}{\sqrt{T}}\hat{\mathrm{E}}$: $UDV^\top$, with $D = \mathrm{diag}(\gamma_i) \in \mathbb{R}^{n\times n}$, $U \in \mathbb{R}^{n\times n}$ and $V \in \mathbb{R}^{T\times n}$ such that $U^\top U = V^\top V = \mathrm{Id}_n$. Then the solution in S to Problem (4.45) is $\hat{S} = U\,\mathrm{diag}\left([\gamma_i]_{\underline{\sigma}}^{\bar{\sigma}}\right)U^\top$ (this result can be derived from Massias et al. 2018a, Prop. 2). $\hat{S}$ can be used to bound $\|X^\top\hat{\mathrm{E}}\|_{2,\infty}$:

Lemma 4.11. *For the concomitant multivariate square-root Lasso* (4.5) *and the smoothed concomitant multivariate square-root* (4.45) *we have:*

$$\|X^\top\hat{\mathrm{E}}\|_{2,\infty} \leq \|\hat{S}\|_2\|X^\top\hat{S}^{-1}\hat{\mathrm{E}}\|_{2,\infty} \ . \tag{4.46}$$

Proof

Concomitant multivariate square-root ($\hat{S} = (\hat{\mathrm{E}}\hat{\mathrm{E}}^\top)^{1/2}$). We recall that $UDV^\top$ is a singular value decomposition of $\frac{1}{\sqrt{T}}\hat{\mathrm{E}}$, with $D = \mathrm{diag}(\gamma_i) \in \mathbb{R}^{r\times r}$, $U \in \mathbb{R}^{n\times r}$ and $V \in \mathbb{R}^{T\times r}$ such that $U^\top U = V^\top V = \mathrm{Id}_r$.

[3] we keep the name of van de Geer (2016), although a better name in our opinion would be the (multitask) trace norm Lasso, but the name is used by Grave et al. (2011) when the nuclear norm is used as a regularizer

We have, observing that $\hat{S}^{-1}\hat{\mathrm{E}} = (UD^2U^\top)^{-1/2}\sqrt{T}UDV^\top = \sqrt{T}UV^\top$:

$$X^\top\hat{\mathrm{E}} = \sqrt{T}X^\top UDV^\top \tag{4.47}$$
$$= \sqrt{T}X^\top UV^\top VDV^\top \tag{4.48}$$
$$= X^\top\hat{S}^{-1}\hat{\mathrm{E}}VDV^\top \ . \tag{4.49}$$

Therefore,

$$\|X^\top\hat{\mathrm{E}}\|_{2,\infty} \leq \|VDV^\top\|_2\|X^\top\hat{S}^{-1}\hat{\mathrm{E}}\|_{2,\infty} \tag{4.50}$$
$$\leq \|\hat{S}\|_2\|X^\top\hat{S}^{-1}\hat{\mathrm{E}}\|_{2,\infty} \ . \tag{4.51}$$

Equation (4.46) also holds for Problem (4.45).

Smoothed concomitant multivariate square-root ($\hat{S} = U\operatorname{diag}([\gamma_i]_{\underline{\sigma}}^{\bar{\sigma}})U^\top$). We recall that $UDV^\top$ is a singular value decomposition of $\frac{1}{\sqrt{T}}\hat{\mathrm{E}}$, with $D = \operatorname{diag}(\gamma_i) \in \mathbb{R}^{n\times n}$, $U \in \mathbb{R}^{n\times n}$ and $V \in \mathbb{R}^{T\times n}$ such that $U^\top U = V^\top V = \mathrm{Id}_n$.

Observing that

$$\hat{S}^{-1}\hat{\mathrm{E}} = U\operatorname{diag}([\gamma_i]_{\underline{\sigma}}^{\bar{\sigma}})^{-1}U^\top\sqrt{T}UDV^\top = \sqrt{T}U\operatorname{diag}([\gamma_i]_{\underline{\sigma}}^{\bar{\sigma}})^{-1}\operatorname{diag}(\gamma_i)V^\top \ , \tag{4.52}$$

we have

$$\begin{aligned} X^\top\hat{\mathrm{E}} &= \sqrt{T}X^\top U\operatorname{diag}(\gamma_i)V^\top \\ &= \sqrt{T}X^\top U\operatorname{diag}([\gamma_i]_{\underline{\sigma}}^{\bar{\sigma}})^{-1}\operatorname{diag}(\gamma_i)\operatorname{diag}([\gamma_i]_{\underline{\sigma}}^{\bar{\sigma}})V^\top \\ &= \sqrt{T}X^\top U\operatorname{diag}([\gamma_i]_{\underline{\sigma}}^{\bar{\sigma}})^{-1}\operatorname{diag}(\gamma_i)V^TV\operatorname{diag}([\gamma_i]_{\underline{\sigma}}^{\bar{\sigma}})V^\top \\ &= \sqrt{T}X^\top\hat{S}^{-1}\hat{\mathrm{E}}V\operatorname{diag}([\gamma_i]_{\underline{\sigma}}^{\bar{\sigma}})V^\top \ . \end{aligned}$$

Therefore,

$$\|X^\top\hat{\mathrm{E}}\|_{2,\infty} \leq \|X^\top\hat{S}^{-1}\hat{\mathrm{E}}\|_{2,\infty}\|V\operatorname{diag}([\gamma_i]_{\underline{\sigma}}^{\bar{\sigma}})V^\top\|_2 \tag{4.53}$$
$$\leq \|X^\top\hat{S}^{-1}\hat{\mathrm{E}}\|_{2,\infty}\|\hat{S}\|_2 \ . \tag{4.54}$$

■

We can prove the minimax sup-norm convergence of these two estimators, using the following assumptions.

Assumption 4.12. *For the multivariate square-root Lasso, $\hat{\mathrm{E}}^\top\hat{\mathrm{E}}$ is invertible, and there exists η such that $\|(\frac{1}{T}\hat{\mathrm{E}}^\top\hat{\mathrm{E}})^{\frac{1}{2}}\|_2 \leq (2+\eta)\sigma^*$.*

We get rid of this very strong hypothesis for the smoothed version, as the estimated noise covariance is invertible because of the constraint $S \succeq \underline{\sigma}\,\mathrm{Id}_n$, and we can control its operator norm via the constraint $S \preceq \bar{\sigma}\,\mathrm{Id}_n$. We still need an assumption on $\underline{\sigma}$ and $\bar{\sigma}$.

Assumption 4.13. *$\underline{\sigma}$, $\bar{\sigma}$ and η verify: $\underline{\sigma} \leq \frac{\sigma^*}{\sqrt{2}}$ and $\bar{\sigma} = (2+\eta)\sigma^*$ with $\eta \geq 1$.*

Proposition 4.14. *For the multivariate square-root Lasso (4.4) (*resp. *its smoothed version (4.45)), let Assumption 4.1 be satisfied, let α satisfy Assumption 4.2 and let η satisfy Assumption 4.12 (*resp. *let $\underline{\sigma}, \bar{\sigma}, \eta$ satisfy Assumption 4.13). Let $C = \Big(1 +$*

$\frac{16}{7(\alpha-1)}\Big)$, $A \geq \sqrt{2}$, *and* $\lambda = \frac{2\sqrt{2}}{\sqrt{nT}}(1 + A\sqrt{(\log p)/T})$. *Then there exists* $c \geq 1/64$ *such that with probability at least* $1 - p^{1-A^2/2} - 2ne^{-cT/n}$,

$$\frac{1}{T}\|\hat{\mathrm{B}} - \mathrm{B}^*\|_{2,\infty} \leq C(3+\eta)\lambda\sigma^* \ . \tag{4.55}$$

Moreover if

$$\min_{j\in\mathcal{S}^*} \frac{1}{T}\|\mathrm{B}^*_{j:}\|_2 > 2C(3+\eta)\lambda\sigma^* \ , \tag{4.56}$$

then with the same probability:

$$\hat{\mathcal{S}} \triangleq \{j \in [p] : \frac{1}{T}\|\hat{\mathrm{B}}_{j:}\|_2 > C(3+\eta)\lambda\sigma^*\} \tag{4.57}$$

correctly estimates the true sparsity pattern: $\hat{\mathcal{S}} = \mathcal{S}^*$.

Proof Let $\mathcal{A}_2$ be the event:

$$\left\{\frac{\|X^\top \mathrm{E}\|_{2,\infty}}{nT} \leq \frac{\lambda\sigma^*}{2\sqrt{2}}\right\} \cap \{2\sigma^* \operatorname{Id}_T \succ (\tfrac{\mathrm{E}^\top\mathrm{E}}{n})^{\frac{1}{2}} \succ \tfrac{\sigma^*}{\sqrt{2}} \operatorname{Id}_T\} \ . \tag{4.58}$$

By Lemma 4.16 (i), $\mathbb{P}(\mathcal{A}_2) \geq 1 - p^{1-A^2/2} - 2ne^{-cT/n}$ ($c \leq 1/64$). When the multivariate square-root Lasso residuals are full rank, the optimality conditions for Problems (4.4) and (4.45) read the same, but with differents $\hat{S}$ (introduced above):

$$\|X^\top \hat{S}^{-1}\hat{\mathrm{E}}\|_{2,\infty} \leq \lambda T n \ . \tag{4.59}$$

With Lemma 4.11 and eq. (4.59) and Assumption 4.12 for the multivariate square-root Lasso (or Assumption 4.13 for its smoothed version):

$$\begin{aligned} n\|\Psi\Delta\|_{2,\infty} &= \|X^\top(\mathrm{E} - \hat{\mathrm{E}})\|_{2,\infty} \\ &\leq \|X^\top\hat{\mathrm{E}}\|_{2,\infty} + \|X^\top\mathrm{E}\|_{2,\infty} \\ &\leq \lambda T n\|\hat{S}\|_2 + \|X^\top\mathrm{E}\|_{2,\infty} \\ &\leq \lambda(2+\eta)Tn\sigma^* + \|X^\top\mathrm{E}\|_{2,\infty} \ . \end{aligned} \tag{4.60}$$

Then on the event $\mathcal{A}_2$:

$$\begin{aligned} \frac{1}{T}\|\Psi\Delta\|_{2,\infty} &\leq \lambda(2+\eta)\sigma^* + \tfrac{1}{nT}\|X^\top\mathrm{E}\|_{2,\infty} \\ &\leq \big(3+\eta\big)\,\lambda\sigma^* \ . \end{aligned} \tag{4.61}$$

Finally we exhibit an element of $\partial f(\mathrm{E})$ to apply Lemma 4.5 (b). Recall that $f = \frac{1}{n\sqrt{T}}\|\cdot\|_*$ for Problem (4.5), and $f = \min_{\bar\sigma \operatorname{Id}_n \succeq S \succeq \underline\sigma \operatorname{Id}_n} \frac{1}{2nT}\|\cdot\|^2_{S^{-1}} + \frac{\operatorname{Tr} S}{2n}$ for Problem (4.45). We also recall that for a full rank matrix $A \in \mathbb{R}^{n\times T}$ (Koltchinskii et al., 2011, Sec. 2):

$$\partial\|A\|_* = \{(AA^\top)^{-1/2}A\} \ . \tag{4.62}$$

On $\mathcal{A}_2$, $\partial f(\mathrm{E})$ is a singleton for both estimators, whose element is $(\mathrm{E}\mathrm{E}^\top)^{-1/2}\mathrm{E}/(n\sqrt{T})$. Additionally on $\mathcal{A}_2$:

$$\begin{aligned} \tfrac{1}{n\sqrt{T}}\|X^\top(\mathrm{E}\mathrm{E}^\top)^{-1/2}\mathrm{E}\|_{2,\infty} &\leq \tfrac{1}{nT}\|X^\top\mathrm{E}\|_{2,\infty}\|(\tfrac{\mathrm{E}\mathrm{E}^\top}{T})^{-1/2}\|_2 \\ &\leq \frac{\lambda\sigma^*}{2\sqrt{2}} \times \frac{\sqrt{2}}{\sigma^*} \leq \frac{\lambda}{2} \ , \end{aligned} \tag{4.63}$$

meaning we can apply Lemma 4.5 (b) with $Z = \mathrm{E}(\mathrm{E}^\top \mathrm{E})^{-1/2}/n\sqrt{T}$. This proves the bound on $\|\Delta\|_{2,\infty}$. Then, the support recovery property easily follows from Lounici et al. (2009, Cor. 4.1). ■

4.4 Experiments

We first describe the setting of Figures 4.1 and 4.2. Then we show that empirically that results given by Propositions 4.9 and 4.14 hold in practice. The signal-to-noise ratio (SNR) is defined as $\frac{\|X\mathrm{B}^*\|_F}{\|Y - X\mathrm{B}\|_F}$.

4.4.1 Pivotality of the square-root Lasso

In this experiment the matrix X consists of the 10 000 first columns of the *climate* dataset ($n = 864$). We generate β^* with 20 non-zero entries. Random Gaussian noise is added to $X\beta^*$ to create y, with a noise variance σ^* controlling the SNR.

For each SNR value, both for the Lasso and the square-root Lasso, we compute the optimal λ on a grid between $\lambda_{\max}$ (the estimator specific smallest regularization level yielding a 0 solution), using cross validation on prediction error on left out data. For each SNR, results are averaged over 10 realizations of y.

Figure 4.1 shows that, in accordance with theory, the optimal λ for the Lasso depends linearly on the noise level, while the square-root Lasso achieves pivotality.

4.4.2 Rank deficiency experiment

For $(n, T, p) = (10, 20, 30)$, we simulate data: entries of X are i.i.d. $\mathcal{N}(0, 1)$, B^* has 5 non zeros rows, and Gaussian noise is to $X\mathrm{B}^*$ added to result in a SNR of 1. We reformulate Problem (4.4) as a Conic Program, and solve it with the SCS solver of `cvxpy` (O'Donoghue et al., 2016; Diamond and Boyd, 2016) for various values of λ ($\lambda_{\max}$ is the smallest regularization value yielding a null solution). We then plot the singulars values of the residuals at optimum, shown on Figure 4.2.

Since the problem is reformulated as a Conic Program and solved approximately (precision $\epsilon = 10^{-6}$), the residuals are not exact; however the sudden drop of singular values of $Y - X\hat{\mathrm{B}}$ must be interpreted as the singular value being exactly 0. One can see that even for very high values of λ, the residuals are rank deficient while the matrix Y is not. This is most likely due to the trace penalty on S in the equivalent formulation of Problem (4.5), encouraging singular values to be 0. Therefore, even on simple toy data, the hypothesis used by van de Geer and Stucky (2016); Molstad (2019) does not hold, justifying the need for smoothing approaches, both from practical and theoretical point of views.

4.4.3 (Multitask) smoothed concomitant Lasso

Here we illustrate, as indicated by theory, that when the smoothing parameter $\underline{\sigma}$ is sufficiently small, the multitask SCL is able to recover the true support (Proposition 4.9). More precisely, when $\underline{\sigma} \leq \sigma^*/\sqrt{2}$, there exist a λ, independent of $\underline{\sigma}$ and σ^*, such that the multitask SCL recovers the true support with high probability. We use $(n, T, p) = (50, 50, 1000)$. The design X is random with Toeplitz-correlated features

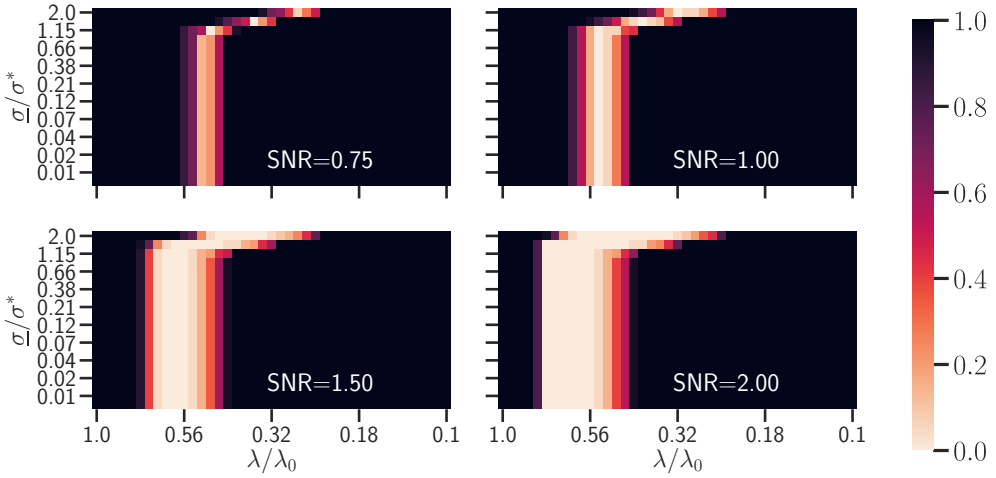


Figure 4.3 – (Synthetic data, $n = 50$, $p = 1000$, $T = 20$) Hard recovery loss for different values of SNR for the multitask SCL.

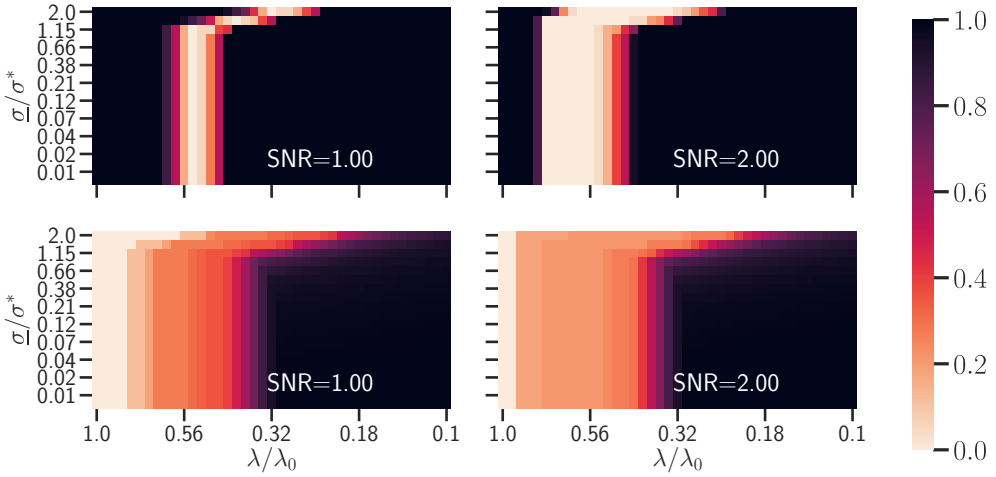


Figure 4.4 – (Synthetic data, $n = 50$, $p = 1000$, $T = 20$) Hard recovery loss (top) and percent of non-zeros coefficients (bottom) for different values of SNR: SNR $= 1$ (left), SNR $= 2$ (right) for the multitask SCL.

with parameter $\rho_X = 0.5$ (correlation between $X_{:i}$ and $X_{:j}$ is $\rho_X^{|i-j|}$), and its columns have unit Euclidean norm. The true coefficient B^* has 5 non-zeros rows whose entries are i.i.d. $\mathcal{N}(0, 1)$.

Comments on Figures 4.3 and 4.4 The multitask SCL relies on two hyperparameters: the penalization coefficient λ and the smoothing parameter $\underline{\sigma}$, whose influence we study here. The goal is to show empirically that when $\underline{\sigma} \leq \sigma^*/\sqrt{2}$ the optimal λ does not depend on the smoothing parameter $\underline{\sigma}$. We vary λ and $\underline{\sigma}$ on a grid: for each pair $(\lambda, \underline{\sigma})$ we solve the multitask SCL. For each solution $\hat{\mathrm{B}}^{(\lambda,\underline{\sigma})}$ we then compute a metric, the hard recovery (Figure 4.3) or the size of the support (Figure 4.4). The metrics are averaged over 100 realizations of the noise. Figure 4.3 shows the latter graph for different values of SNR. We can see that when $\underline{\sigma} \leq \sigma^*$, support recovery is achieved for λ independent of $\underline{\sigma}$. As soon as $\underline{\sigma} > \sigma^*$ the optimal λ depends on $\underline{\sigma}$. When $\underline{\sigma}$

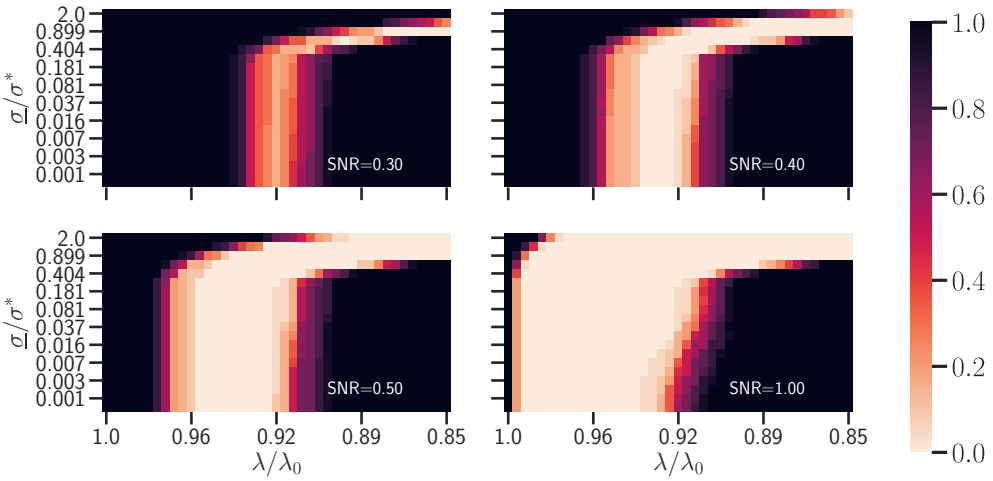


Figure 4.5 – (Synthetic data, $n = 150$, $p = 500$, $T = 100$) Hard recovery loss for different values of SNR for the SGCL.

reaches a large enough value (*i.e.*, σ^*) then the recovery profile is modified: the optimal λ decreases as $\underline{\sigma}$ grows. This is logical, since as soon as the constraint is saturated, the (multitask) SCL boils down to a multitask Lasso with regularization parameter $\lambda\underline{\sigma}$:

$$\hat{\mathrm{B}} \triangleq \underset{\mathrm{B}\in\mathbb{R}^{p\times T}}{\arg\min} \frac{1}{2nT} \left\|Y - X\mathrm{B}\right\|_F^2 + \lambda\underline{\sigma} \left\|\mathrm{B}\right\|_{2,1} \quad . \tag{4.64}$$

Figure 4.4 shows that with a fixed λ higher values of $\underline{\sigma}$ may lead to smaller support size, see *e.g.*, $\lambda/\lambda_0 = 0.32$.

4.4.4 Smoothed generalized concomitant Lasso (SGCL)

The experimental setting is the same as before, except here we used $(n, T, p) = (150, 100, 500)$. Figure 4.5 illustrates Proposition 4.14. When $\underline{\sigma} \leq \sigma^*$, there exist a λ that does not depend on $\underline{\sigma}$ and such that SGCL finds the true support $\mathcal{S}^*$. However, as before, when $\underline{\sigma} \geq \sqrt{2}\sigma^*$, λ depends on $\underline{\sigma}$.

4.5 Conclusion

We have proved sup norm convergence rates and support recovery for a family of sparse estimators derived from the square-root Lasso. We showed that they are pivotal too: the optimal regularization parameter does not depend on the noise level. We showed that their smoothed versions retain these properties while being simpler to solve, and requiring more realistic assumptions to be analyzed. These findings were corroborated numerically, in particular for the influence of the smoothing parameter.

Appendix

4.A Concentration inequalities

The following theorem is a powerful tool to show a lot of concentration inequalities:

Theorem 4.15 (Giraud 2014, Thm B.6 p. 221)**.** *Assume that $F : \mathbb{R}^d \to \mathbb{R}$ is 1-Lipschitz and z has $\mathcal{N}(0, \sigma^2 \operatorname{Id}_d)$ as a distribution, then there exists a variable ξ, exponentially distributed with parameter 1, such that:*

$$F(z) \leq \mathbb{E}[F(z)] + \sqrt{2\xi} \ . \tag{4.65}$$

Lemma 4.16. *a) Let $\mathcal{C}_1 \triangleq \left\{ \frac{1}{n} \|X^\top \varepsilon\|_\infty \leq \frac{\lambda}{2} \right\}$. Take $\lambda = A\sigma^* \sqrt{(\log p)/n}$ and $A > 2\sqrt{2}$, then:*

$$\mathbb{P}(\mathcal{C}_1) \geq 1 - 2p^{1-\frac{A^2}{8}} \ . \tag{4.66}$$

b) Let $\mathcal{C}_1' \triangleq \left\{ \frac{1}{n} \|X^\top \varepsilon\|_\infty \leq \frac{\lambda}{2} \frac{\sigma^*}{\sqrt{2}} \right\}$. Take $\lambda = A\sqrt{(2\log p)/n}$ and $A > 2\sqrt{2}$, then:

$$\mathbb{P}(\mathcal{C}_1) \geq 1 - 2p^{1-\frac{A^2}{8}} \ . \tag{4.67}$$

c) Let $\mathcal{C}_2 \triangleq \left\{ \frac{1}{nT} \|X^\top \mathrm{E}\|_{2,\infty} \leq \frac{\lambda}{2} \right\}$. Take $\lambda = \frac{2\sigma^*}{\sqrt{nT}} \left(1 + A\sqrt{\frac{\log p}{T}} \right)$ and $A > \sqrt{2}$, then:

$$\mathbb{P}(\mathcal{C}_2) \geq 1 - p^{1-A^2/2} \ . \tag{4.68}$$

Another possible control is Lounici et al. (2009, Proof of Lemma 3.1, p. 6). Let $A > 8$ and $\lambda = \frac{2\sigma^*}{\sqrt{nT}} \sqrt{1 + \frac{A \log p}{\sqrt{T}}}$, then:

$$\mathbb{P}(\mathcal{C}_2) \geq 1 - p^{\min(8 \log p, A\sqrt{T}/8)} \ . \tag{4.69}$$

d) Let $\mathcal{C}_3 \triangleq \left\{ \frac{1}{nT} \|X^\top \mathrm{E}\|_{2,\infty} \leq \frac{\lambda \sigma^*}{2\sqrt{2}} \right\}$. Take $\lambda = \frac{2\sqrt{2}}{\sqrt{nT}} \left(1 + A\sqrt{\frac{\log p}{T}} \right)$ and $A > \sqrt{2}$, then:

$$\mathbb{P}(\mathcal{C}_3) \geq 1 - p^{1-A^2/2} \ . \tag{4.70}$$

e) Let $\mathcal{C}_4 \triangleq \left\{ \frac{\sigma^*}{\sqrt{2}} < \frac{\|\varepsilon\|_F}{\sqrt{nT}} < 2\sigma^* \right\}$. Then:

$$\mathbb{P}(\mathcal{C}_4) \geq 1 - (1 + e^2) e^{-nT/24} \ . \tag{4.71}$$

f) Let $\mathcal{C}_5 \triangleq \left\{ \left(\frac{\mathrm{E}\mathrm{E}^\top}{T} \right)^{\frac{1}{2}} \succ \frac{\sigma^*}{\sqrt{2}} \right\}$. Then with $c \geq \frac{1}{32}$:

$$\mathbb{P}(\mathcal{C}_5) \geq 1 - n e^{-cT/(2n)} \ . \tag{4.72}$$

g) Let $\mathcal{C}_6 \triangleq \left\{ 2\sigma^* \succ \left(\frac{\mathrm{E}\mathrm{E}^\top}{T} \right)^{\frac{1}{2}} \right\}$. *Then with* $c \geq \frac{1}{32}$*:*

$$\mathbb{P}(\mathcal{C}_6) \geq 1 - ne^{-cT/n} \ . \tag{4.73}$$

h) Let us recall that $\mathcal{A}_1 = \left\{ \frac{\|X^\top \mathrm{E}\|_{2,\infty}}{\sqrt{nT}\|\mathrm{E}\|_F} \leq \frac{\lambda}{2} \right\} \cap \left\{ \frac{\sigma^*}{\sqrt{2}} < \frac{\|\mathrm{E}\|_F}{\sqrt{nT}} < 2\sigma^* \right\}$*, we have*

$$\mathbb{P}(\mathcal{A}_1) \geq 1 - p^{1-A^2/2} - (1+e^2)e^{-nT/24} \ . \tag{4.74}$$

i) Let us recall that $\mathcal{A}_2 = \left\{ \frac{\|X^\top \mathrm{E}\|_{2,\infty}}{nT} \leq \frac{\lambda\sigma^*}{2\sqrt{2}} \right\} \cap \left\{ 2\sigma^* \mathrm{Id}_T \succ (\frac{\mathrm{E}\mathrm{E}^\top}{n})^{\frac{1}{2}} \succ \frac{\sigma^*}{\sqrt{2}} \mathrm{Id}_T \right\}$

Proof Lemma 4.16 (a):

$$\begin{aligned}
\mathbb{P}\left(\mathcal{C}_1^c\right) &\leq p\,\mathbb{P}\left(\left|X_{:1}^\top \varepsilon\right| \geq n\lambda/2\right) \\
&\leq p\,\mathbb{P}\left(\left|\varepsilon_1\right| \geq \sqrt{n}\lambda/2\right) && (\left\|X_{:1}\right\| = \sqrt{n}) \\
&\leq p\,\mathbb{P}\left(\left|\varepsilon_1\right|/\sigma \geq \sqrt{n}\lambda/(2\sigma)\right) \\
&\leq 2p\exp\left(-\frac{n\lambda^2}{8\sigma^2}\right) && \text{(Theorem 4.15)} \\
&\leq 2p^{1-\frac{A^2}{8}} && (\lambda = A\sigma\sqrt{(\log p)/n}) \ .
\end{aligned} \tag{4.75}$$

Lemma 4.16 (b) is a direct consequence of Lemma 4.16 (a).

Lemma 4.16 (c): since E is isotropic, the law of $u^\top \mathrm{E}$ is the same for all vectors $u \in \mathbb{R}^n$ of same norm. In particular, $X_{:1}^\top \mathrm{E}$ and $\sqrt{n}e_1^\top \mathrm{E} = \sqrt{n}\mathrm{E}_{1:}$ have the same law.

The variable $\frac{1}{\sigma}\|\mathrm{E}_{1:}\|_2$ is a *chi variable with* T *degrees of freedom*, and

$$\frac{1}{\sigma}\mathbb{E}[\|\mathrm{E}_{1:}\|_2] = \frac{\sqrt{2}\Gamma(\frac{T+1}{2})}{\Gamma(\frac{T}{2})} \in \left[\frac{T}{\sqrt{T+1}}, \sqrt{T}\right] \ , \tag{4.76}$$

where the bound can be proved by recursion. We have:

$$\begin{aligned}
\mathbb{P}\left(\mathcal{C}_2^c\right) &\leq p\,\mathbb{P}\left(\|X_{:1}^\top \mathrm{E}\|_2 \geq Tn\lambda/2\right) \\
&\leq p\,\mathbb{P}\left(\|\mathrm{E}_{1:}\|_2 \geq T\sqrt{n}\lambda/2\right) && \text{(by isotropy of E)} \\
&\leq p\,\mathbb{P}\left(\|\mathrm{E}_{1:}\|_2 \geq \sigma\sqrt{T} + A\sigma\sqrt{\log p}\right) && (\lambda = \tfrac{2\sigma}{T\sqrt{n}}(\sqrt{T} + A\sqrt{\log p})) \\
&\leq p\,\mathbb{P}\left(\|\mathrm{E}_{1:}\|_2 \geq \mathbb{E}(\|\mathrm{E}_{1:}\|_2) + A\sigma\sqrt{\log p}\right) && (\sigma\sqrt{T} \geq \mathbb{E}(\|\mathrm{E}_{1:}\|_2) \\
&\leq p^{1-\frac{A^2}{2}} && \text{(Theorem 4.15)} \ .
\end{aligned} \tag{4.77}$$

The proof of the other control of $\mathcal{A}_2$ can be found in Lounici et al. (2009, Proof of Lemma 3.1, p. 6).

Lemma 4.16 (d) is a direct consequence of Lemma 4.16 (c).

Proof of Lemma 4.16 (e) can be found in Giraud (2014, Proof of Lemma 5.4 p. 112), who control the finer event $\{\frac{\sigma}{\sqrt{2}} \leq \frac{\|\varepsilon\|}{\sqrt{n}} \leq (2 - \frac{1}{\sqrt{2}})\sigma\}$.

Proof of Lemmas 4.16 (f) and 4.16 (g) are particular cases of Gittens and Tropp (2011, Cor. 7.2, p. 15).

Proof of Lemma 4.16 (h) is done using Lemmas 4.16 (d) and 4.16 (e). Indeed we have $A_1 \supset \left\{\frac{1}{nT}\|X^\top \mathrm{E}\|_{2,\infty} \leq \frac{\lambda\sigma^*}{2\sqrt{2}}\right\} \cap \left\{\frac{\sigma^*}{\sqrt{2}} < \frac{\|\mathrm{E}\|_F}{\sqrt{nT}} < 2\sigma^*\right\} = \mathcal{C}_3 \cap \mathcal{C}_4$. Hence $\mathbb{P}(\mathcal{C}_1) \geq 1 - \mathbb{P}(\mathcal{C}_3^c) - \mathbb{P}(\mathcal{C}_4^c) \geq 1 - p^{1-A^2/2} - (1 + e^2)e^{-nT/24}$.

Proof of Lemma 4.16 (i) is done using Lemmas 4.16 (d), 4.16 (f) and 4.16 (g). Indeed $A_2 = \mathcal{C}_3 \cap \mathcal{C}_5 \cap \mathcal{C}_6$. Hence $\mathbb{P}(\mathcal{A}_2) \geq 1 - \mathbb{P}(\mathcal{C}_3^c) - \mathbb{P}(\mathcal{C}_5^c) - \mathbb{P}(\mathcal{C}_6^c) \geq 1 - p^{1-A^2/2} - 2ne^{-cT/(2n)}$. ■

5 Application to neuro-imaging

Contents

This chapters provide an application of the pivotal estimators presented in Chapter 4: the multivariate square-root Lasso. In addition to have a regularization parameter independent from the noise level, the (smoothed version of the) multivariate square-root Lasso can cope with complex noise structure by using non-averaged measurements. The resulting optimization problem consists in a smoothed trace norm as a data fit, and an usual $\ell_{2,1}$-norm as a penalty. It is jointly convex and amenable to efficient block coordinate descent. Practical benefits are extensively demonstrated on real M/EEG neuro-imaging data, including visual and auditory cognitive experiments.

This chapter is based on the following work, accepted at NeurIPS 2019:

- **Q. Bertrand**, M. Massias, A. Gramfort, and J. Salmon. Handling correlated and repeated measurements with the smoothed multivariate square-root Lasso. In *NeurIPS*, 2019

5.1 Introduction

In many statistical applications, the number of parameters p is much larger than the number of observations n. A popular approach to tackle linear regression problems in such scenarios is to consider convex ℓ_1-type penalties, as popularized by Tibshirani (1996). The use of these penalties relies on a regularization parameter λ trading data fidelity versus sparsity. Unfortunately, Bickel et al. (2009) showed that, in the case of white Gaussian noise, the optimal λ depends linearly on the standard deviation of the noise – referred to as *noise level*. Because the latter is rarely known in practice, one can jointly estimate the noise level and the regression coefficients, following pioneering work on concomitant estimation (Huber and Dutter, 1974; Huber, 1981). Adaptations to sparse regression (Owen, 2007) have been analyzed under the names of square-root Lasso (Belloni et al., 2011) or scaled Lasso (Sun and Zhang, 2012). Generalizations

have been proposed in the multitask setting, the canonical estimator being Multi-Task Lasso (Obozinski et al., 2010).

The latter estimators take their roots in a white Gaussian noise model. However some real-world data (such as magneto-electroencephalographic data) are contaminated with strongly non-white Gaussian noise (Engemann and Gramfort, 2015). From a statistical point of view, the non-uniform noise level case has been widely explored: Daye et al. (2012); Wagener and Dette (2012); Kolar and Sharpnack (2012); Dalalyan et al. (2013). In a more general case, with a correlated Gaussian noise model, estimators based on non-convex optimization problems were proposed (Lee and Liu, 2012) and analyzed for sub-Gaussian covariance matrices (Chen and Banerjee, 2017) through the lens of penalized maximum likelihood estimation (MLE). Other estimators (Rothman et al., 2010; Rai et al., 2012) assume that the inverse of the covariance (the *precision matrix*) is sparse, but the underlying optimization problems remain non-convex. A convex approach to regression with correlated noise, the Smooth Generalized Concomitant Lasso (SGCL) was proposed by Massias et al. (2018a). Relying on smoothing techniques (Moreau, 1965; Nesterov, 2005; Beck and Teboulle, 2012), the SGCL jointly estimates the regression coefficients and the noise *co-standard deviation matrix* (the square root of the noise covariance matrix). However, in applications such as M/EEG, the number of parameters in the co-standard deviation matrix ($\approx 10^4$) is typically equal to the number of observations, making it statistically hard to estimate accurately.

In this chapter we consider applications to M/EEG data in the context of neuroscience. M/EEG data consists in recordings of the electric and magnetic fields at the surface or close to the head. Here we tackle the *source localization* problem, which aims at estimating which regions of the brain are responsible for the observed electro-magnetic signals: this problem can be cast as a multitask high dimensional linear regression (Ndiaye et al., 2015). MEG and EEG data are obtained from heterogeneous types of sensors: magnetometers, gradiometers and electrodes, leading to samples contaminated with different noise distributions, and thus non-white Gaussian noise. Moreover the additive noise in M/EEG data is correlated between sensors and rather strong: the noise variance is commonly even stronger that the signal power. It is thus customary to make several repetitions of the same cognitive experiment, *e.g.,* showing 50 times the same image to a subject in order to record 50 times the electric activity of the visual cortex. The multiple measurements are then classically averaged across the experiment's repetitions in order to increase the signal-to-noise ratio. In other words, popular estimators for M/EEG usually discard the individual observations, and rely on Gaussian i.i.d. noise models (Ou et al., 2009; Gramfort et al., 2013).

In this work we propose Concomitant Lasso with Repetitions (CLaR), an estimator that is

- Designed to exploit all available measurements collected during repetitions of experiments.
- Defined as the solution of a *convex* minimization problem, handled efficiently by proximal block coordinate descent techniques.
- Built thanks to an *explicit* connection with nuclear norm smoothing. This can also be viewed as a partial smoothing of the multivariate square-root Lasso (van de Geer and Stucky, 2016).
- Shown (through extensive benchmarks with respect to existing estimators) to leverage experimental repetitions to improve support identification,

- Available as open source code to reproduce all the experiments.

In Section 5.2, we recall the framework of concomitant estimation, and introduce CLaR. In Section 5.2.2, we detail the properties of CLaR, and derive an algorithm to solve it. Finally, Section 5.3 is dedicated to experimental results.

5.2 Concomitant estimation with correlated noise

5.2.1 Model and proposed estimator

Probabilistic model. Let r be the number of repetitions of the experiment. The r observation matrices are denoted $Y^{(1)}, \dots, Y^{(r)} \in \mathbb{R}^{n \times T}$ with n the number of sensors/samples and T the number of tasks/time samples. The mean over the repetitions of the observation matrices is written $\bar{Y} = \frac{1}{r}\sum_{l=1}^{r} Y^{(l)}$. Let $X \in \mathbb{R}^{n \times p}$ be the design (or gain) matrix, with p features stored column-wise: $X = [X_{:1}|\dots|X_{:p}]$, where for a matrix $A \in \mathbb{R}^{m \times n}$ its j^{th} column (*resp.* row) is denoted $A_{:j} \in \mathbb{R}^{m \times 1}$ (*resp.* $A_{j:} \in \mathbb{R}^{1 \times n}$. The matrix $\mathrm{B}^* \in \mathbb{R}^{p \times T}$ contains the coefficients of the linear regression model.

Each measurement (*i.e.,* repetition of the experiment) follows the model:

$$\forall l \in [r], \quad Y^{(l)} = X\mathrm{B}^* + S^*\mathrm{E}^{(l)} \ , \tag{5.1}$$

where the entries of $\mathrm{E}^{(l)}$ are i.i.d. samples from standard normal distributions, the $\mathrm{E}^{(l)}$'s are independent, and $S^* \in \mathcal{S}^n_{++}$ is the co-standard deviation matrix, and $\mathcal{S}^n_{++}$ (*resp.* $\mathcal{S}^n_+$) stands for the set of positive (*resp.* semi-definite positive) matrices. Note that even if the observations $Y^{(1)}, \dots, Y^{(r)}$ differ because of the noise $\mathrm{E}^{(1)}, \dots, \mathrm{E}^{(r)}$, B^* and the noise structure S^* are shared across repetitions.

Notation. The unit ℓ_p ball is written $\mathcal{B}_p$, $p \in [1, \infty)$. For $a, b \in \mathbb{R}$, we denote $(a)_+ = \max(a, 0)$, $a \vee b = \max(a, b)$ and $a \wedge b = \min(a, b)$. The block soft-thresholding operator at level $\tau > 0$, is denoted $\mathrm{BST}(\cdot, \tau)$, and reads for any vector x, $\mathrm{BST}(x, \tau) = \Big(1 - \tau/\|x\|\Big)_+ x$. Let $d \in \mathbb{N}$, and let $\mathcal{C}$ be a closed and convex subset of $\mathbb{R}^d$.

To leverage the multiple repetitions while taking into account the noise structure, we introduce the Concomitant Lasso with Repetitions (CLaR):

Definition 5.1. *CLaR estimates the parameters of Model (5.1) by solving:*

$$(\hat{\mathrm{B}}^{\mathrm{CLaR}}, \hat{\mathrm{S}}^{\mathrm{CLaR}}) \in \underset{\substack{\mathrm{B} \in \mathbb{R}^{p \times T} \\ S \succeq \underline{\sigma}\, \mathrm{Id}_n}}{\arg\min}\, f(\mathrm{B}, S) + \lambda \left\|\mathrm{B}\right\|_{2,1}, \ \textit{with}\ f(\mathrm{B}, S) \triangleq \sum_{l=1}^{r} \frac{\left\|Y^{(l)} - X\mathrm{B}\right\|^2_{S^{-1}}}{2nTr} + \frac{\mathrm{Tr}(S)}{2n}, \tag{5.2}$$

where $\lambda > 0$ controls the sparsity of $\hat{\mathrm{B}}^{\mathrm{CLaR}}$ and $\underline{\sigma} > 0$ controls the smallest eigenvalue of $\hat{\mathrm{S}}^{\mathrm{CLaR}}$.

In low SNR settings, a standard way to deal with strong noise is to use the averaged observation $\bar{Y} \in \mathbb{R}^{n \times T}$ instead of the raw observations. The associated model reads:

$$\bar{Y} = X\mathrm{B}^* + \tilde{S}^*\tilde{\mathrm{E}} \ , \tag{5.3}$$

with $\tilde{S}^* \triangleq S^*/\sqrt{r}$ and $\tilde{\mathrm{E}}$ has *i.i.d.* entries drawn from a standard normal distribution. The SNR[1] is multiplied by $\sqrt{r}$, yet the number of samples goes from rnT to nT, making it statistically difficult to estimate the $\mathcal{O}(n^2)$ parameters of S^*. CLaR generalizes

[1] See the definition we consider in eq. (5.46).

the Smoothed Generalized Concomitant Lasso (Massias et al., 2018a), which has the drawback of only targeting averaged observations:

Definition 5.2 (SGCL, Massias et al. 2018a)**.** *SGCL estimates the parameters of Model (5.3), by solving:*

$$(\hat{\mathrm{B}}^{\mathrm{SGCL}}, \hat{\mathrm{S}}^{\mathrm{SGCL}}) \in \underset{\substack{\mathrm{B}\in\mathbb{R}^{p\times T} \\ \tilde{S}\succeq \underline{\sigma}/\sqrt{r}\,\mathrm{Id}_n}}{\arg\min} \; \tilde{f}(\mathrm{B}, \tilde{S}) + \lambda \left\|\mathrm{B}\right\|_{2,1}, \; \textit{with } \tilde{f}(\mathrm{B}, \tilde{S}) \triangleq \frac{\|\bar{Y} - X\mathrm{B}\|^2_{\tilde{S}^{-1}}}{2nT} + \frac{\mathrm{Tr}(\tilde{S})}{2n}. \tag{5.4}$$

Remark 5.3. *Note that* $\hat{\mathrm{S}}^{\mathrm{CLaR}}$ *estimates* S^**, while* $\hat{\mathrm{S}}^{\mathrm{SGCL}}$ *estimates* $\tilde{S}^* = S^*/\sqrt{r}$*. Since we impose the constraint* $\hat{\mathrm{S}}^{\mathrm{CLaR}} \succeq \underline{\sigma}\,\mathrm{Id}_n$*, we rescale the constraint so that* $\hat{\mathrm{S}}^{\mathrm{SGCL}} \succeq \underline{\sigma}/\sqrt{r}\,\mathrm{Id}_n$ *in (5.4) for future comparisons. Also note that CLaR and SGCL are the same when* $r = 1$ *and* $Y^{(1)} = \bar{Y}$*.*

The justification for CLaR is the following: if the quadratic loss $\left\|Y - X\mathrm{B}\right\|^2$ were used, the parameters of Model (5.1) could be estimated by using either $\|\bar{Y} - X\mathrm{B}\|^2$ or $\frac{1}{r}\sum\|Y^{(l)} - X\mathrm{B}\|^2$ as a data fitting term. Yet, both alternatives yield the same solutions as the two terms are equal up to constants. Hence, the quadratic loss does not leverage the multiple repetitions and ignores the noise structure. On the contrary, the more refined data fitting term of CLaR allows to take into account the individual repetitions, leading to improved performance in applications.

5.2.2 Properties of the proposed data fitting term

Let us analyze the data fitting term of CLaR, by connecting it to the Schatten 1-norm. Let us define the following smoothing function:

$$\omega_{\underline{\sigma}}(\cdot) \triangleq \frac{1}{2}\left(\|\cdot\|^2 + n\right)\underline{\sigma}\;, \tag{5.5}$$

and the inf-convolution of functions f_1 and f_2, $f_1 \,\square\, f_2(y) \triangleq \inf_x f_1(x) + f_2(y - x)$. The name "smoothing" used in this chapter comes from the following fact: if f_1 is a closed proper convex function, then $f_1^* + \frac{1}{2}\|\cdot\|^2$ is strongly convex, and thus its Fenchel transform $(f_1^* + \frac{1}{2}\|\cdot\|^2)^* = (f_1^* + (\frac{1}{2}\|\cdot\|^2)^*)^* = (f_1 \,\square\, \frac{1}{2}\|\cdot\|^2)^{**} = f_1 \,\square\, \frac{1}{2}\|\cdot\|^2$ is smooth.

The next propositions are key to our framework and show the connection between the SGCL, CLaR and the Schatten 1-norm:

Proposition 5.4 (Smoothing of the trace norm)**.** *For the choice* $\omega(\cdot) = \frac{1}{2}\|\cdot\|^2 + \frac{n}{2}$*, and with* $n \leq T$*, the* $\omega_{\underline{\sigma}}$*-smoothing of the Schatten-1 norm,* i.e.*, the function* $\|\cdot\|_{\mathscr{S},1} \,\square\, \omega_{\underline{\sigma}} : \mathbb{R}^{n\times T} \mapsto \mathbb{R}$*, has the following closed-form formula, for all* $Z \in \mathbb{R}^{n\times T}$*:*

$$(\|\cdot\|_{\mathscr{S},1} \,\square\, \omega_{\underline{\sigma}})(Z) = \min_{S\succeq \underline{\sigma}\,\mathrm{Id}_n} \tfrac{1}{2}\left\|Z\right\|^2_{S^{-1}} + \tfrac{1}{2}\,\mathrm{Tr}(S)\;. \tag{5.6}$$

Proof The proof relies on a direct calculus. We will evaluate each member of Equation (5.6) on a matrix $Z \in \mathbb{R}^{n\times T}$, through a singular value decomposition of Z (Lemma 5.6).

Proposition 5.5 (Usual properties of inf-convolution)**.** *Let $g : \mathbb{R}^d \to \mathbb{R}$, $h : \mathbb{R}^d \to \mathbb{R}$ be closed proper convex functions. Then, the following holds (see Parikh et al. 2013, p. 136):*

$$h^{**} = h \ , \tag{5.7}$$
$$(h \,\square\, g)^* = h^* + g^* \ , \tag{5.8}$$
$$\left(\underline{\sigma} g\left(\frac{\cdot}{\underline{\sigma}}\right)\right)^* = \underline{\sigma} g^* \ , \tag{5.9}$$
$$\|\cdot\|_p^* = \iota_{\mathcal{B}_{p^*}}, \ \textit{where} \ \frac{1}{p} + \frac{1}{p^*} = 1 \ , \tag{5.10}$$
$$(h + \delta)^* = h^* - \delta, \quad \forall \delta \in \mathbb{R} \ , \tag{5.11}$$
$$\left(\tfrac{1}{2}\|\cdot\|^2\right)^* = \tfrac{1}{2}\|\cdot\|^2 \ . \tag{5.12}$$

We now compute each member of Equation (5.6) on a matrix $Z \in \mathbb{R}^{n \times T}$,

Lemma 5.6. *Let $Z \in \mathbb{R}^{n \times T}$ and $V \operatorname{diag}(\gamma_1, \dots, \gamma_n, 0, \dots, 0) W^\top$ its singular value decomposition, then for the choice $\omega(\cdot) = \frac{1}{2}\|\cdot\|^2 + \frac{n}{2}$, and with $n \leq T$:*

a) $\left(\|\cdot\|_{\mathscr{S},1} \,\square\, \omega_{\underline{\sigma}}\right)(Z) = \frac{1}{2\underline{\sigma}} \sum_{\gamma_i \leq \underline{\sigma}} \gamma_i^2 + \sum_{\gamma_i > \underline{\sigma}} \gamma_i + \frac{1}{2} \sum_{\gamma_i \leq \underline{\sigma}} \underline{\sigma}$.

b) $\min_{S \succeq \underline{\sigma} \operatorname{Id}_n} \frac{1}{2} \operatorname{Tr}[Z^\top S^{-1} Z] + \frac{1}{2} \operatorname{Tr}(S) = \frac{1}{2\underline{\sigma}} \sum_{\gamma_i \leq \underline{\sigma}} \gamma_i^2 + \sum_{\gamma_i > \underline{\sigma}} \gamma_i + \frac{1}{2} \sum_{\gamma_i \leq \underline{\sigma}} \underline{\sigma}$.

We first compute the value of the right-hand member in Equation (5.6)

Proof (Lemma 5.6 (a)).

$$
\begin{aligned}
\left(\|\cdot\|_{\mathscr{S},1} \square \omega_{\underline{\sigma}}\right)(Z) &= \left(\|\cdot\|_{\mathscr{S},1} \square \omega_{\underline{\sigma}}\right)^{**}(Z) \quad \text{(using eq. (5.7))} \\
&= \left(\|\cdot\|^*_{\mathscr{S},1} + \omega^*_{\underline{\sigma}}\right)^*(Z) \quad \text{(using eq. (5.8))} \\
&= \left(\iota_{\mathcal{B}_{\mathscr{S},\infty}} + \tfrac{\underline{\sigma}}{2}\|\cdot\|^2 - \frac{n}{2}\underline{\sigma}\right)^*(Z) \quad \text{(using eq. (5.10))} \\
&= \left(\tfrac{\underline{\sigma}}{2}\|\cdot\|^2 + \iota_{\mathcal{B}_{\mathscr{S},\infty}}\right)^*(Z) + \frac{n}{2}\underline{\sigma} \quad \text{(using eq. (5.11))} \\
&= \sup_{U \in \mathbb{R}^{n\times T}} \left(\langle U, Z\rangle - \frac{\underline{\sigma}}{2}\|U\|^2 - \iota_{\mathcal{B}_{\mathscr{S},\infty}}(U)\right) + \frac{n}{2}\underline{\sigma} \\
&= \sup_{U \in \mathcal{B}_{\mathscr{S},\infty}} \left(\langle U, Z\rangle - \frac{\underline{\sigma}}{2}\|U\|^2\right) + \frac{n}{2}\underline{\sigma} \\
&= -\inf_{U \in \mathcal{B}_{\mathscr{S},\infty}} \left(\frac{\underline{\sigma}}{2}\|U\|^2 - \langle U, Z\rangle\right) + \frac{n}{2}\underline{\sigma} \\
&= -\underline{\sigma} \cdot \inf_{U \in \mathcal{B}_{\mathscr{S},\infty}} \left(\tfrac{1}{2}\|U\|^2 - \left\langle U, \tfrac{Z}{\underline{\sigma}}\right\rangle\right) + \frac{n}{2}\underline{\sigma} \\
&= -\underline{\sigma} \cdot \inf_{U \in \mathcal{B}_{\mathscr{S},\infty}} \left(\tfrac{1}{2}\|U - \tfrac{Z}{\underline{\sigma}}\|^2 - \tfrac{1}{2\underline{\sigma}^2}\|Z\|^2\right) + \frac{n}{2}\underline{\sigma} \\
&= \tfrac{1}{2\underline{\sigma}}\|Z\|^2 - \tfrac{\underline{\sigma}}{2} \cdot \inf_{U \in \mathcal{B}_{\mathscr{S},\infty}} \left(\|U - \tfrac{Z}{\underline{\sigma}}\|^2\right) + \frac{n}{2}\underline{\sigma} \\
&= \tfrac{1}{2\underline{\sigma}}\|Z\|^2 - \tfrac{\underline{\sigma}}{2}\|\Pi_{\mathcal{B}_{\mathscr{S},\infty}}\left(\tfrac{Z}{\underline{\sigma}}\right) - \tfrac{Z}{\underline{\sigma}}\|^2 + \frac{n}{2}\underline{\sigma} \quad \text{(with } \Pi_{\mathcal{B}_{\mathscr{S},\infty}} \text{ the projection on } \mathcal{B}_{\mathscr{S},\infty}) \\
&= \tfrac{1}{2\underline{\sigma}}\|Z\|^2 + \frac{n}{2}\underline{\sigma} - \tfrac{\underline{\sigma}}{2}\left(\|V \operatorname{diag}\left(\tfrac{\gamma_1}{\underline{\sigma}} \wedge 1 - \tfrac{\gamma_1}{\underline{\sigma}}, \dots, \tfrac{\gamma_n}{\underline{\sigma}} \wedge 1 - \tfrac{\gamma_n}{\underline{\sigma}}\right) W^\top\|^2\right) \\
&\text{(using } \Pi_{\mathcal{B}_{\mathscr{S},\infty}}\left(\tfrac{Z}{\underline{\sigma}}\right) = V \operatorname{diag}\left(\tfrac{\gamma_1}{\underline{\sigma}} \wedge 1, \dots, \tfrac{\gamma_n}{\underline{\sigma}} \wedge 1\right) W^\top \text{, see Beck 2017, Example 7.31)} \\
&= \tfrac{1}{2\underline{\sigma}}\|Z\|^2 + \frac{n}{2}\underline{\sigma} - \tfrac{\underline{\sigma}}{2}\left(\sum_{i=1}^n \left(\tfrac{\gamma_i}{\underline{\sigma}} \wedge 1 - \tfrac{\gamma_i}{\underline{\sigma}}\right)^2\right) \\
&= \tfrac{1}{2\underline{\sigma}}\|Z\|^2 + \frac{n}{2}\underline{\sigma} - \tfrac{\underline{\sigma}}{2}\left(\frac{1}{\underline{\sigma}^2}\sum_{i=1}^n \left(\gamma_i \wedge \underline{\sigma} - \gamma_i\right)^2\right) \\
&= \tfrac{1}{2\underline{\sigma}}\|Z\|^2 + \frac{n}{2}\underline{\sigma} - \tfrac{\underline{\sigma}}{2}\left(\frac{1}{\underline{\sigma}^2}\sum_{\gamma_i > \underline{\sigma}} \left(\gamma_i \wedge \underline{\sigma} - \gamma_i\right)^2\right) \\
&= \tfrac{1}{2\underline{\sigma}}\|Z\|^2 + \frac{n}{2}\underline{\sigma} - \tfrac{\underline{\sigma}}{2}\left(\frac{1}{\underline{\sigma}^2}\sum_{\gamma_i > \underline{\sigma}} \left(\underline{\sigma} - \gamma_i\right)^2\right) \\
&= \tfrac{1}{2\underline{\sigma}}\|Z\|^2 + \frac{n}{2}\underline{\sigma} - \tfrac{\underline{\sigma}}{2}\left(\frac{1}{\underline{\sigma}^2}\sum_{\gamma_i > \underline{\sigma}} \left(\underline{\sigma}^2 + \gamma_i^2 - 2\underline{\sigma}\gamma_i\right)\right) \\
&= \tfrac{1}{2\underline{\sigma}}\|Z\|^2 + \frac{n}{2}\underline{\sigma} - \tfrac{\underline{\sigma}}{2}\left(\sum_{\gamma_i > \underline{\sigma}} 1 + \frac{1}{\underline{\sigma}^2}\sum_{\gamma_i > \underline{\sigma}} \gamma_i^2 - 2\frac{1}{\underline{\sigma}}\sum_{\gamma_i > \underline{\sigma}} \gamma_i\right) \\
&= \tfrac{1}{2\underline{\sigma}}\|Z\|^2 + \frac{n}{2}\underline{\sigma} - \frac{\underline{\sigma}}{2}\sum_{\gamma_i > \underline{\sigma}} 1 - \frac{1}{2\underline{\sigma}}\sum_{\gamma_i > \underline{\sigma}} \gamma_i^2 + \sum_{\gamma_i > \underline{\sigma}} \gamma_i \\
&= \frac{1}{2\underline{\sigma}}\sum_{\gamma_i \le \underline{\sigma}} \gamma_i^2 + \sum_{\gamma_i > \underline{\sigma}} \gamma_i + \frac{1}{2}\sum_{\gamma_i \le \underline{\sigma}} \underline{\sigma} \ .
\end{aligned}
$$

■

We now compute the value of the left-hand member in Equation (5.6)

Proof (Lemma 5.6 (b)). The minimum of $\min_{S \succeq \underline{\sigma} \operatorname{Id}_n} \frac{1}{2} \operatorname{Tr}[Z^\top S^{-1} Z] + \frac{1}{2} \operatorname{Tr}(S)$ is attained at $\hat{S} = V \operatorname{diag}(\gamma_1 \vee \underline{\sigma}, \dots, \gamma_n \vee \underline{\sigma}) V^\top$ (see Massias et al. 2018a, Prop. 2). Thus we have:

$$\begin{aligned}
\min_{S \succeq \underline{\sigma} \operatorname{Id}_n} \frac{1}{2} \operatorname{Tr}[Z^\top S^{-1} Z] + \frac{1}{2} \operatorname{Tr}(S) &= \frac{1}{2} \operatorname{Tr}[Z^\top \hat{S}^{-1} Z] + \frac{1}{2} \operatorname{Tr}(\hat{S}) \\
&= \frac{1}{2} \operatorname{Tr}[\hat{S}^{-1} Z Z^\top] + \frac{1}{2} \operatorname{Tr}(\hat{S}) \\
&= \frac{1}{2} \operatorname{Tr}[V \operatorname{diag}(\gamma_1^2 / (\gamma_1 \vee \underline{\sigma}), \dots, \gamma_n^2 / (\gamma_n \vee \underline{\sigma})) V^\top] \\
&+ \frac{1}{2} \operatorname{Tr}[V \operatorname{diag}(\gamma_1 \vee \underline{\sigma}, \dots, \gamma_n \vee \underline{\sigma}) V^\top] \\
&= \underbrace{\frac{1}{2} \sum_{i=1}^n \frac{\gamma_i^2}{\gamma_i \vee \underline{\sigma}}}_{\frac{1}{2} \sum_{\gamma_i > \underline{\sigma}} \gamma_i + \frac{1}{2\underline{\sigma}} \sum_{\gamma_i \leq \underline{\sigma}} \gamma_i^2} + \underbrace{\frac{1}{2} \sum_{i=1}^n \gamma_i \vee \underline{\sigma}}_{\frac{1}{2} \sum_{\gamma_i > \underline{\sigma}} \gamma_i + \frac{1}{2} \sum_{\gamma_i \leq \underline{\sigma}} \underline{\sigma}} \\
&= \frac{1}{2\underline{\sigma}} \sum_{\gamma_i \leq \underline{\sigma}} \gamma_i^2 + \sum_{\gamma_i > \underline{\sigma}} \gamma_i + \frac{1}{2} \sum_{\gamma_i \leq \underline{\sigma}} \underline{\sigma} \ .
\end{aligned}$$

■

Proof (Proposition 5.4). Proposition 5.4 is a direct consequence of Lemmas 5.6 (a) and 5.6 (b). ■

■

Properties similar to Proposition 5.4 can be traced back to van de Geer and Stucky (2016, Sec 2.2), who introduced the multivariate square-root Lasso:

$$\hat{\mathrm{B}} \in \operatorname*{arg\,min}_{\mathrm{B} \in \mathbb{R}^{p \times T}} \frac{1}{n\sqrt{T}} \|\bar{Y} - X\mathrm{B}\|_{\mathscr{S},1} + \lambda \left\|\mathrm{B}\right\|_{2,1} \ , \tag{5.13}$$

and showed that if $(\bar{Y} - X\hat{\mathrm{B}})(\bar{Y} - X\hat{\mathrm{B}})^\top \succ 0$, the latter optimization problem admits a variational[2] formulation:

$$(\hat{\mathrm{B}}, \hat{\mathrm{S}}) \in \operatorname*{arg\,min}_{\substack{\mathrm{B} \in \mathbb{R}^{p \times T}, \\ \tilde{S} \succ 0}} \frac{1}{2nT} \|\bar{Y} - X\mathrm{B}\|_{S^{-1}}^2 + \frac{\operatorname{Tr}(S)}{2n} + \lambda \left\|\mathrm{B}\right\|_{2,1} . \tag{5.14}$$

In other words Proposition 5.4 generalizes van de Geer (2016, Lemma 3.4) for all matrices $\bar{Y} - X\hat{\mathrm{B}}$, getting rid of the condition $(\bar{Y} - X\hat{\mathrm{B}})(\bar{Y} - X\hat{\mathrm{B}})^\top \succ 0$. In the present contribution, the problem formulation in Proposition 5.4 is motivated by computational aspects, as it helps to address the combined non-smoothness of the data fitting term $\|\cdot\|_{\mathscr{S},1}$ and the penalty term $\|\cdot\|_{2,1}$. Note that another smoothing of the nuclear norm was proposed in Argyriou et al. (2008); Bach et al. (2012, Sec. 5.2):

$$Z \mapsto \min_{S \succ 0} \frac{1}{2} \operatorname{Tr}[Z^\top S^{-1} Z] + \frac{1}{2} \operatorname{Tr}(S) + \frac{\sigma^2}{2} \operatorname{Tr}(S^{-1}) \ , \tag{5.15}$$

[2] also called *concomitant* formulation since minimization is performed over an additional variable (Owen, 2007; Ndiaye et al., 2017).

which is a $\underline{\sigma}$-smooth $n\underline{\sigma}$-approximation of $\|\cdot\|_{\mathscr{S},1}$, therefore less precise than ours (Lemma 5.8).

First let us recall the definition of a smoothable function and a μ-smooth approximation of Beck and Teboulle (2012, Def. 2.1):

Definition 5.7 (Smoothable function, μ-smooth approximation)**.** *Let $g : \mathbb{E} \to]-\infty, +\infty]$ be a closed and proper convex function, and let $E \subseteq \mathrm{dom}(g)$ be a closed convex set. The function g is called* (α, δ, K)-smoothable *on E if there exists δ_1, δ_2 satisfying $\delta_1 + \delta_2 = \delta > 0$ such that for every μ there exists a continuously differentiable convex function $g_\mu : \mathbb{E} \to]-\infty, +\infty[$ such that the following holds:*

a) $g(x) - \delta_1\mu \leq g_\mu(x) \leq g(x) + \delta_2\mu$ for every $x \in E$.

b) The function ∇g_μ has a Lipschitz constant which is less than or equal to $K + \frac{\alpha}{\mu}$:

$$\|\nabla g_\mu(x) - \nabla g_\mu(y)\| \leq \left(K + \frac{\alpha}{\mu}\right) \|x - y\| \quad \text{for every } x, y \in E \ . \tag{5.16}$$

The function g is called a μ-smooth *approximation of g with parameters (α, δ, K).*

The nuclear norm $\|\cdot\|_{\mathscr{S},1}$ is nonsmooth (and not even differentiable at 0), but one can construct a smooth approximation of the nuclear norm based on the following variational formula, if $ZZ^\top \succ 0$:

$$\|Z\|_{\mathscr{S},1} = \min_{S \succ 0} \frac{1}{2} \mathrm{Tr}[Z^\top S^{-1} Z] + \frac{1}{2} \mathrm{Tr}(S) \ , \tag{5.17}$$

see van de Geer (2016, Lemma 3.4). When $ZZ^\top \not\succ 0$, one can approximate $\|\cdot\|_{\mathscr{S},1}$ with

$$\min_{S \succeq \underline{\sigma}\,\mathrm{Id}} \frac{1}{2} \mathrm{Tr}[Z^\top S^{-1} Z] + \frac{1}{2} \mathrm{Tr}(S) = \|\cdot\|_{\mathscr{S},1} \square\, \omega_{\underline{\sigma}} \ , \tag{5.18}$$

as shown in Lemma 5.8. It can be shown that this approximation stays close to the nuclear norm.

Lemma 5.8. *Let $Z \in \mathbb{R}^{n \times T}$, $(\gamma_1, \ldots, \gamma_n)$ be its singular values, and $n \leq T$*

a) $\|\cdot\|_{\mathscr{S},1} \square\, \omega_{\underline{\sigma}}$ is a $\underline{\sigma}$-smooth approximation of $\|\cdot\|_{\mathscr{S},1}$ with parameters $(1, \frac{n}{2}, 0)$. More precisely: $\|\cdot\|_{\mathscr{S},1} \square\, \omega_{\underline{\sigma}}$ has a $\underline{\sigma}$-Lipschitz gradient and

$$0 \leq \|\cdot\|_{\mathscr{S},1} \square\, \omega_{\underline{\sigma}} - \|\cdot\|_{\mathscr{S},1} = \frac{\underline{\sigma}}{2} \sum_{\gamma_i < \underline{\sigma}} \left(1 - \frac{\gamma_i}{\underline{\sigma}}\right)^2 \leq \frac{\underline{\sigma}}{2} n \ . \tag{5.19}$$

b) $Z \mapsto \min_{S \succ 0} \frac{1}{2} \mathrm{Tr}[Z^\top S^{-1} Z] + \frac{1}{2} \mathrm{Tr}(S) + \frac{\underline{\sigma}^2}{2} \mathrm{Tr}(S^{-1})$ is a $\underline{\sigma}$-smooth approximation of $\|\cdot\|_{\mathscr{S},1}$ with parameters $(1, n, 0)$. More precicely: $Z \mapsto \min_{S \succ 0} \frac{1}{2} \mathrm{Tr}[Z^\top S^{-1} Z] + \frac{1}{2} \mathrm{Tr}(S) + \frac{\underline{\sigma}^2}{2} \mathrm{Tr}(S^{-1})$ has a gradient $\underline{\sigma}$-Lipschitz and

$$0 \leq \min_{S \succ 0} \frac{1}{2} \mathrm{Tr}[Z^\top S^{-1} Z] + \frac{1}{2} \mathrm{Tr}(S) + \frac{\underline{\sigma}^2}{2} \mathrm{Tr}(S^{-1}) - \|Z\|_{\mathscr{S},1} = \underline{\sigma} \sum_i \frac{1}{\sqrt{1 + \frac{\gamma_i^2}{\underline{\sigma}^2}} + \frac{\gamma_i}{\underline{\sigma}}} \leq \underline{\sigma} n \ . \tag{5.20}$$

c) It can be shown that with a fixed Lipschitz constant, the proposed smoothing is (at least) uniformly a twice better approximation. This can be quantified even more precisely:

$$0 \leq \underbrace{\left(\|\cdot\|_{\mathscr{S},1} \,\square\, \omega_{\underline{\sigma}}\right)(Z) - \|Z\|_{\mathscr{S},1}}_{\mathrm{Err}_1(Z)} \leq \frac{1}{2}\left(\underbrace{\min_{S\succ 0} \frac{1}{2}\operatorname{Tr}[Z^\top S^{-1} Z] + \frac{1}{2}\operatorname{Tr}(S) + \frac{\underline{\sigma}^2}{2}\operatorname{Tr}(S^{-1}) - \|Z\|_{\mathscr{S},1}}_{\mathrm{Err}_2(Z)}\right) \tag{5.21}$$

More precisely

$$\frac{1}{2}\mathrm{Err}_2(Z) - \mathrm{Err}_1(Z) = \frac{\underline{\sigma}}{2}\sum_{\gamma_i \geq \sigma} \underbrace{\left(\sqrt{1 + \frac{\gamma_i^2}{\underline{\sigma}^2}} - \frac{\gamma_i}{\underline{\sigma}}\right)}_{\geq 0} + \frac{\underline{\sigma}}{2}\sum_{\gamma_i < \sigma} \underbrace{\left(\frac{1}{\sqrt{1 + \frac{\gamma_i^2}{\underline{\sigma}^2}} + \frac{\gamma_i}{\underline{\sigma}}} - (1 + \frac{\gamma_i}{\underline{\sigma}})^2\right)}_{\geq 0}\ , \tag{5.22}$$

which means that for a fixed smoothing constant $\underline{\sigma}$, our smoothing is at least twice uniformly better. Moreover the proposed smoothing can be much better, in particular when a lot a singular values are around $\underline{\sigma}$.

Proof (Lemma 5.8 (a)). Since ω is 1-smooth, Beck and Teboulle (2012, Thm. 4.1) shows that $\|\cdot\|_{\mathscr{S},1} \,\square\, \omega_{\underline{\sigma}}$ is $\underline{\sigma}$-smooth.

Let $Z \in \mathbb{R}^{n\times T}$ and let $\gamma_1, \ldots, \gamma_n$ be its singular value decomposition:

$$\begin{aligned}
\left(\|\cdot\|_{\mathscr{S},1} \,\square\, \omega_{\underline{\sigma}}\right)(Z) - \|Z\|_{\mathscr{S},1} &= \frac{1}{2}\sum_{i=1}^n \frac{\gamma_i^2}{\gamma_i \vee \underline{\sigma}} + \frac{1}{2}\sum_{i=1}^n \gamma_i \vee \underline{\sigma} + \frac{1}{2}\sum_{n+1}^n \underline{\sigma} - \sum_{i=1}^n \gamma_i \\
&= \frac{1}{2}\sum_{i=1}^n \left(\frac{\gamma_i^2}{\gamma_i \vee \underline{\sigma}} + \gamma_i \vee \underline{\sigma} - 2\gamma_i\right) + \frac{1}{2}\sum_{n+1}^n \underline{\sigma} \\
&= \frac{1}{2}\sum_{\gamma_i \leq \underline{\sigma}} \left(\frac{\gamma_i^2}{\gamma_i \vee \underline{\sigma}} + \gamma_i \vee \underline{\sigma} - 2\gamma_i\right) + \frac{1}{2}\sum_{n+1}^n \underline{\sigma} \\
&= \frac{1}{2}\sum_{\gamma_i \leq \underline{\sigma}} \left(\frac{\gamma_i^2}{\underline{\sigma}} + \underline{\sigma} - 2\gamma_i\right) + \frac{1}{2}\sum_{n+1}^n \underline{\sigma} \\
&= \frac{1}{2}\sum_{\gamma_i \leq \underline{\sigma}} \frac{(\gamma_i - \underline{\sigma})^2}{\underline{\sigma}}\ .
\end{aligned} \tag{5.23}$$

Hence,

$$0 \leq \left(\|\cdot\|_{\mathscr{S},1} \,\square\, \omega_{\underline{\sigma}}\right)(Z) - \|Z\|_{\mathscr{S},1} = \frac{1}{2}\sum_{\gamma_i \leq \underline{\sigma}} \frac{(\gamma_i - \underline{\sigma})^2}{\underline{\sigma}} \leq \frac{\underline{\sigma}}{2} n\ . \tag{5.24}$$

Moreover this bound is attained when $Z = 0$. ■

Proof (Lemma 5.8 (b)). Another regularization was proposed in Argyriou et al. (2008); Bach et al. (2012, p. 62):

$$\min_{S\succ 0} \underbrace{\frac{1}{2}\operatorname{Tr}[Z^\top S^{-1} Z] + \frac{1}{2}\operatorname{Tr}(S) + \frac{\underline{\sigma}^2}{2}\operatorname{Tr}(S^{-1})}_{h(S^{-1})} \ . \tag{5.25}$$

By putting the gradient of the objective function in Equation (5.25) to zero it follows that:

$$0 = \nabla h(\hat{S}^{-1}) = ZZ^\top - \hat{S}^2 + \underline{\sigma}^2 \operatorname{Id} \ , \tag{5.26}$$

leading to :

$$\hat{S} = (ZZ^\top + \underline{\sigma}^2 \operatorname{Id})^{\frac{1}{2}} \ . \tag{5.27}$$

Let $\gamma_1, \ldots, \gamma_n$ be the singular values of Z:

$\sum_{i=1}^n \sqrt{\gamma_i^2 + \underline{\sigma}^2}$ is a $\underline{\sigma}$-smooth approximation of $\sum_{i=1}^n \sqrt{\gamma_i^2} = \|Z\|_{\mathscr{S},1}$, see Beck and Teboulle (2012, Example 4.6).

$$\begin{aligned}
&\min_{S\succ 0} \frac{1}{2}\operatorname{Tr}[Z^\top S^{-1} Z] + \frac{1}{2}\operatorname{Tr}(S) + \frac{\underline{\sigma}^2}{2}\operatorname{Tr}(S^{-1}) - \|Z\|_{\mathscr{S},1} \\
&= \frac{1}{2}\sum_{i=1}^n \left(\frac{\gamma_i^2}{\sqrt{\gamma_i^2 + \underline{\sigma}^2}} + \sqrt{\gamma_i^2 + \underline{\sigma}^2} + \frac{\underline{\sigma}^2}{\sqrt{\gamma_i^2 + \underline{\sigma}^2}} \right) - \|Z\|_{\mathscr{S},1} \\
&= \frac{1}{2}\sum_{i=1}^n \left(\frac{\gamma_i^2 + \gamma_i^2 + \underline{\sigma}^2 + \underline{\sigma}^2}{\sqrt{\gamma_i^2 + \underline{\sigma}^2}} \right) - \|Z\|_{\mathscr{S},1} \\
&= \sum_{i=1}^n \sqrt{\gamma_i^2 + \underline{\sigma}^2} - \|Z\|_{\mathscr{S},1} \\
&= \sum_{i=1}^n \left(\sqrt{\gamma_i^2 + \underline{\sigma}^2} - \gamma_i \right) \\
&= \sum_{i=1}^n \frac{\underline{\sigma}^2}{\sqrt{\gamma_i^2 + \underline{\sigma}^2} + \gamma_i} \\
&= \underline{\sigma} \sum_{i=1}^n \frac{1}{\sqrt{1 + \frac{\gamma_i^2}{\underline{\sigma}^2}} + \frac{\gamma_i}{\underline{\sigma}}} \qquad (5.28) \\
&\leq \underline{\sigma} n \ . \qquad (5.29)
\end{aligned}$$

Moreover this bound is attained when $Z = 0$. ■

Proof (Lemma 5.8 (c)). Using the formulas of Err_1 (Equation (5.23)) and Err_2 (Equation (5.28)), Equation (5.22) is direct. In Equation (5.22) the positivity of the first sum is trivial, the positivity of the second can be obtained with an easy function study. ■

Other alternatives to exploit the multiple repetitions without simply averaging them, would consist in investigating other Schatten p-norms:

$$\underset{\mathrm{B}\in\mathbb{R}^{p\times T}}{\arg\min}\ \frac{1}{\sqrt{rT}}\|[Y^{(1)}-X\mathrm{B}|\dots|Y^{(r)}-X\mathrm{B}]\|_{\mathscr{S},p}+\lambda n\left\|\mathrm{B}\right\|_{2,1}\ . \tag{5.30}$$

Without smoothing, problems of the form given in Equation (5.30) present the drawback of having two nonsmooth terms, and calling for primal-dual algorithms (Chambolle and Pock, 2011) with costly proximal operators. Even if the nonsmooth Schatten 1-norm is replaced by the formula in Equation (5.6), numerical challenges remain: S can approach 0 arbitrarily, hence, the gradient with respect to S of the data fitting term is not Lipschitz over the optimization domain. Recently, Molstad (2019) proposed two algorithms to directly solve Equation (5.30): a prox-linear ADMM, and accelerated proximal gradient descent, the latter lacking convergence guarantees since the composite objective has two nonsmooth terms. Before that, van de Geer and Stucky (2016) devised a fixed point method, lacking descent guarantees. A similar problem was raised for the concomitant Lasso by Ndiaye et al. (2017) who used smoothing techniques to address it.

Finally one can explicitly link the data fitting term of the proposed estimator to the smoothed trace norm.

Proposition 5.9 (Schatten 1-norm (nuclear/trace norm) with repetitions)**.** *Let $Z^{(1)},\dots,Z^{(r)}$ be matrices in $\mathbb{R}^{n\times T}$, we define $Z\in\mathbb{R}^{n\times Tr}$ by $Z=[Z^{(1)}|\dots|Z^{(r)}]$. Then for the choice $\omega(\cdot)=\frac{1}{2}\|\cdot\|^2+\frac{n\wedge Tr}{2}$, then the following holds true:*

$$\left(\|\cdot\|_{\mathscr{S},1}\,\square\,\omega_{\underline{\sigma}}(\cdot)\right)(Z)=\min_{S\succeq\underline{\sigma}\,\mathrm{Id}_n}\frac{1}{2}\sum_{l=1}^{r}\mathrm{Tr}\left(Z^{(l)\top}S^{-1}Z^{(l)}\right)+\frac{1}{2}\mathrm{Tr}(S)\ . \tag{5.31}$$

Proof The result is a direct application of Proposition 5.4, with $Z=[Z^{(1)}|\dots|Z^{(r)}]$. It suffices to notice that $\mathrm{Tr}\,Z^\top S^{-1}Z=\sum_{l=1}^{r}\mathrm{Tr}\left(Z^{(l)\top}S^{-1}Z^{(l)}\right)$. ■

5.2.3 Properties of the proposed estimator and algorithmic details

We detail the principal results needed to solve Problem (5.2) numerically, leading to the implementation proposed in Algorithm 5.1. We first recall useful results for alternate minimization of convex composite problems.

Proposition 5.10 (Joint convexity)**.** *The function f defined in Problem (5.2) is jointly convex in (B,S). Moreover, f is convex and smooth on the feasible set, and $\|\cdot\|_{2,1}$ is convex and separable in $\mathrm{B}_{j:}$'s, thus minimizing the objective alternatively in S and in $\mathrm{B}_{j:}$'s (Algorithm 5.1) converges to a global minimum.*

Proof

$$f(\mathrm{B},S)=\frac{1}{2nTr}\sum_{1}^{r}\|Y^{(l)}-X\mathrm{B}\|_{S^{-1}}^2+\frac{1}{2n}\mathrm{Tr}(S)=\mathrm{Tr}(Z^TS^{-1}Z)+\frac{1}{2n}\mathrm{Tr}(S)\ ,$$

with $Z=\frac{1}{\sqrt{2nTr}}[Y^{(1)}-X\mathrm{B}|\dots|Y^{(r)}-X\mathrm{B}]$.

Algorithm 5.1 ALTERNATE MINIMIZATION FOR CLAR

input : $X, \bar{Y}, \underline{\sigma}, \lambda, f^{\text{dual}}, n_{\text{iter}}$
init : $\mathrm{B} = 0_{p,T}$, $S^{-1} = \underline{\sigma}^{-1}\,\mathrm{Id}_n$, $\bar{R} = \bar{Y}$, $\mathrm{cov}_Y = \frac{1}{r}\sum_{l=1}^{r} Y^{(l)}Y^{(l)\top}$ `// precomputed`
for $t = 1, \dots, n_{\text{iter}}$ **do**
  if $t = 1 \pmod{f^{\text{dual}}}$ **then** `// noise update`
    $RR^\top = \mathrm{RRT}(\mathrm{cov}_Y, Y, X, \mathrm{B})$ `// Eq. (5.40)`
    $S \leftarrow \mathrm{ClSqrt}\left(\frac{1}{Tr}RR^\top, \underline{\sigma}\right)$ `// Eq. (5.33)`
    for $j = 1, \dots, p$ **do**
      $L_j = X_{:j}^\top S^{-1} X_{:j}$
  for $j = 1, \dots, p$ **do** `// coef. update`
    $\bar{R} \leftarrow \bar{R} + X_{:j}\mathrm{B}_{j:}$ `// cheap residuals update`
    $\mathrm{B}_{j:} \leftarrow \mathrm{BST}\left(\frac{X_{:j}^\top S^{-1}\bar{R}}{L_j}, \frac{\lambda n T}{L_j}\right)$
    $\bar{R} \leftarrow \bar{R} - X_{:j}\mathrm{B}_{j:}$ `// cheap residuals update`
return B, S

First note that the (joint) function $(Z, \Sigma) \mapsto \mathrm{Tr}\, Z^\top \Sigma^{-1} Z$ is jointly convex over $\mathbb{R}^{n\times T} \times \mathcal{S}_{++}^n$, see Boyd and Vandenberghe (2004, Example 3.4). This means that f is jointly convex in (Z, S), moreover $\mathrm{B} \mapsto \frac{1}{\sqrt{2nTr}}[Y^{(1)} - X\mathrm{B}| \dots |Y^{(r)} - X\mathrm{B}]$ is linear in B, thus f is jointly convex in (B, S), meaning that $(\mathrm{B}, S) \to f + \lambda\|\cdot\|_{2,1}$ is jointly convex in (B, S) . Moreover the constraint set is convex and thus solving CLaR is a convex problem.

The function f is convex and smooth on the feasible set and $\|\cdot\|_{2,1}$ is convex in B and separable in $\mathrm{B}_{j:}$'s, thus (see Tseng 2001; Tseng and Yun 2009a) $f + \lambda\|\cdot\|_{2,1}$ can be minimized through coordinate descent in S and the $\mathrm{B}_{j:}$'s (on the feasible set). ■

Hence, for our alternate minimization implementation, we only need to consider solving problems with B or S fixed, which we detail in the next propositions. Let us first defined the clipped square root, which is used in the minimization in S (Proposition 5.12).

Definition 5.11 (Clipped Square Root)**.** *For $\Sigma \in \mathcal{S}_+^n$ with eigenvalue decomposition $\Sigma = U\,\mathrm{diag}(\gamma_1^2, \dots, \gamma_n^2)U^\top$ (U is orthogonal), let us define the* Clipped Square Root *operator:*

$$\mathrm{ClSqrt}(\Sigma, \underline{\sigma}) = U\,\mathrm{diag}(\gamma_1 \vee \underline{\sigma}, \dots, \gamma_n \vee \underline{\sigma})U^\top \quad . \tag{5.32}$$

Proposition 5.12 (Minimization in S)**.** *Let $\mathrm{B} \in \mathbb{R}^{n\times T}$ be fixed. The minimization of $f(\mathrm{B}, S)$ with respect to S with the constraint $S \succeq \underline{\sigma}\,\mathrm{Id}_n$ admits the closed-form solution:*

$$S = \mathrm{ClSqrt}\left(\frac{1}{rT}\sum_{l=1}^{r}(Y^{(l)} - X\mathrm{B})(Y^{(l)} - X\mathrm{B})^\top, \underline{\sigma}\right) \quad . \tag{5.33}$$

Proof Minimizing $f(\mathrm{B}, \cdot)$ amounts to solving

$$\underset{S\succeq\underline{\sigma}\,\mathrm{Id}_n}{\arg\min}\ \tfrac{1}{2}\|Z\|_{S^{-1}}^2 + \tfrac{1}{2}\,\mathrm{Tr}(S) \quad , \quad \text{with } Z = \frac{1}{\sqrt{r}}[Z^{(1)}| \dots |Z^{(l)}] \quad . \tag{5.34}$$

The solution is $\mathrm{ClSqrt}\left(ZZ^\top, \underline{\sigma}\right)$ (see Massias et al. 2018a, Appendix A2), with $ZZ^\top = \frac{1}{r}\sum_{l=1}^{r} Z^{(l)}Z^{(l)\top}$. ■

Proposition 5.13. *For a fixed $S \in \mathcal{S}^n_{++}$, each step of the block minimization of $f(\cdot, S) + \lambda \|\cdot\|_{2,1}$ in the j^{th} line of* B *admits a closed-form solution:*

$$\mathrm{B}_{j:} = \mathrm{BST}\left(\mathrm{B}_{j:} + \frac{X_{:j}^\top S^{-1}(\bar{Y} - X\mathrm{B})}{\|X_{:j}\|^2_{S^{-1}}}, \frac{\lambda nT}{\|X_{:j}\|^2_{S^{-1}}}\right) \ . \tag{5.35}$$

Proof The function to minimize is the sum of a smooth term $f(\cdot, S)$ and a nonsmooth but separable term, $\|\cdot\|_{2,1}$, whose proximal operator [3] can be computed:

- f is $\|X_{:j}\|^2_{S^{-1}}/nT$-smooth with respect to $\mathrm{B}_{j:}$, with partial gradient

$$\nabla_j f(\cdot, S) = -\frac{1}{nT} X_{:j}^\top S^{-1}(\bar{Y} - X\mathrm{B}) \ . \tag{5.36}$$

- $\|\mathrm{B}\|_{2,1} = \sum_{j=1}^p \|\mathrm{B}_{j:}\|$ is row-wise separable over B, with

$$\mathrm{prox}_{\lambda nT/\|X_{:j}\|^2_{S^{-1}}, \|\cdot\|}(\cdot) = \mathrm{BST}\left(\cdot, \frac{\lambda nT}{\|X_{:j}\|^2_{S^{-1}}}\right) \ . \tag{5.37}$$

Hence, proximal block-coordinate descent converges (Tseng and Yun, 2009a), and the update are given by Equation (5.35). The closed-form formula arises since the smooth part of the objective is quadratic and isotropic with respect to $\mathrm{B}_{j:}$. ■

As for other Lasso-type estimators, there exists $\lambda_{\max} \geq 0$ such that whenever $\lambda \geq \lambda_{\max}$, the estimated coefficients vanish. This $\lambda_{\max}$ helps calibrating roughly λ in practice by choosing it as a fraction of $\lambda_{\max}$.

Proposition 5.14 (Critical regularization parameter)**.** *For the CLaR estimator we have: with* $S_{\max} \triangleq \mathrm{ClSqrt}\left(\frac{1}{Tr}\sum_{l=1}^r Y^{(l)} Y^{(l)\top}, \underline{\sigma}\right)$,

$$\forall \lambda \geq \lambda_{\max} \triangleq \frac{1}{nT} \|X^\top S^{-1}_{\max} \bar{Y}\|_{2,\infty}, \quad \hat{\mathrm{B}}^{\mathrm{CLaR}} = 0 \ . \tag{5.38}$$

Proof Fermat's rules states

$$\begin{aligned} \hat{\mathrm{B}} = 0 &\Leftrightarrow 0 \in \partial\Big(f(\cdot, S_{\max}) + \lambda \|\cdot\|_{2,1}\Big)(0) \\ &\Leftrightarrow -\nabla f(\cdot, S_{\max}) \in \lambda \mathcal{B}_{\|\cdot\|_{2,\infty}} \\ &\Leftrightarrow \frac{1}{nT} \|X^\top S^{-1}_{\max} \bar{Y}\|_{2,\infty} \triangleq \lambda_{\max} \leq \lambda \ . \end{aligned} \tag{5.39}$$

■

Convex formulation benefits. Thanks to the convex formulation, convergence of Algorithm 5.1 can be ensured using the duality gap as a stopping criterion (as it guarantees a targeted sub-optimality level). In addition, convexity allows to leverage acceleration methods such as working sets strategies (Fan and Lv, 2008; Tibshirani et al., 2012;

[3] As a reminder, for a scalar $t > 0$, the proximal operator of a function $h : \mathbb{R}^d \to \mathbb{R}$ can be defined for any $x_0 \in \mathbb{R}^d$ by $\mathrm{prox}_{t,h}(x_0) = \arg\min_{x \in \mathbb{R}^d} \frac{1}{2t}\|x - x_0\|^2 + h(x)$.

Johnson and Guestrin, 2015; Massias et al., 2018b) or safe screening rules (El Ghaoui et al., 2012; Fercoq et al., 2015) while retaining theoretical convergence guarantees. Such techniques are trickier to adapt in the non-convex case, as they could change the local minima reached.

Choice of $\underline{\sigma}$. Although $\underline{\sigma}$ has a smoothing interpretation, from a practical point of view it remains an hyperparameter to set. Following guidelines from Chapter 4, $\underline{\sigma}$ is always chosen as follows: $\underline{\sigma} = \|Y\| / (1000 \times nT)$. In practice, the experimental results were little affected by the choice of $\underline{\sigma}$.

Remark 5.15. *Once* $\mathrm{cov}_Y \triangleq \frac{1}{r}\sum_1^r Y^{(l)}Y^{(l)\top}$ *is pre-computed, the cost of updating* S *does not depend on* r, i.e., *is the same as working with averaged data. Indeed, with* $R = [Y^{(1)} - X\mathrm{B}| \dots |Y^{(r)} - X\mathrm{B}]$, *the following computation can be done in* $\mathcal{O}(qn^2)$.

$$RR^\top = RRT(\mathrm{cov}_Y, Y, X, \mathrm{B}) \triangleq r\mathrm{cov}_Y + r \times (X\mathrm{B})(X\mathrm{B})^\top - r\bar{Y}^\top(X\mathrm{B}) - r(X\mathrm{B})^\top\bar{Y} \ . \tag{5.40}$$

Proof

$$\begin{aligned} RR^\top &= \sum_{l=1}^r R^{(l)}R^{(l)\top} \\ &= \sum_{l=1}^r (Y^{(l)} - X\mathrm{B})(Y^{(l)} - X\mathrm{B})^\top \\ &= \sum_{l=1}^r Y^{(l)}Y^{(l)\top} - \sum_1^r Y^{(l)}(X\mathrm{B})^\top - \sum_1^r X\mathrm{B}Y^{(l)\top} + rX\mathrm{B}(X\mathrm{B})^\top \\ &= r\mathrm{cov}_Y - r\bar{Y}^\top X\mathrm{B} - r(X\mathrm{B})^\top\bar{Y} + rX\mathrm{B}(X\mathrm{B})^\top \ . \end{aligned} \tag{5.41}$$

■

5.3 Experiments

Our Python code (with Numba compilation, Lam et al. 2015) is released as an open source package: https://github.com/QB3/CLaR. The following estimators are compared:

- CLaR defined with Problem (5.2) (ours).
- SGCL defined with Problem (5.4) (Massias et al., 2018a).
- $\ell_{2,1}$-MLE, an $\ell_{2,1}$ version of MLE (Lee and Liu, 2012; Chen and Banerjee, 2017). When minimizing $\ell_{2,1}$-Maximum Likelihood the natural parameters of the problem are the regression coefficients B and the precision matrix Σ^{-1}. Since real M/EEG covariance matrices are not full rank, one has to be algorithmically careful when Σ becomes singular. To avoid such numerical errors and to be consistent with the smoothed estimator proposed in the chapter (CLaR), let us define the (smoothed) $\ell_{2,1}$-MLE as following:

$$(\hat{\mathrm{B}}^{\ell_{2,1}-\mathrm{MLE}}, \hat{\Sigma}^{\ell_{2,1}-\mathrm{MLE}}) \in \underset{\substack{\mathrm{B}\in\mathbb{R}^{p\times T} \\ \Sigma \succeq \underline{\sigma}^2/r^2}}{\arg\min} \|\bar{Y} - X\mathrm{B}\|^2_{\Sigma^{-1}} - \log\det(\Sigma^{-1}) + \lambda\|\mathrm{B}\|_{2,1} \ , \tag{5.42}$$

- $\ell_{2,1}$-MLER, a version of the $\ell_{2,1}$-MLE with multiple repetitions:

$$(\hat{\mathrm{B}}^{\ell_{2,1}\mathrm{MLER}}, \hat{\Sigma}^{\ell_{2,1}\mathrm{MLER}}) \in \underset{\substack{\mathrm{B}\in\mathbb{R}^{p\times T}\\ \Sigma\succeq\underline{\sigma}^2}}{\arg\min} \sum_{1}^{r} \|Y^{(l)} - X\mathrm{B}\|^2_{\Sigma^{-1}} - \log\det(\Sigma^{-1}) + \lambda\|\mathrm{B}\|_{2,1} \ . \tag{5.43}$$

  Problems (5.42) and (5.43) are not convex because the objective functions are not convex in $(\mathrm{B}, \Sigma^{-1})$, however they are biconvex, *i.e.,* convex in B and convex in Σ^{-1}. Alternate minimization can be used to solve Problems (5.42) and (5.43), but without guarantees to converge toward a global minimum.

- MRCER, an $\ell_{2,1}$ penalized version of MRCE (Rothman et al., 2010) with repetitions. MRCE jointly estimates the regression coefficients (assumed to be sparse) and the precision matrix (*i.e.,* the inverse of the covariance matrix), which is supposed to be sparse as well. $\ell_{2,1}$-MRCE is defined as the solution of the following optimization problem:

$$(\hat{\mathrm{B}}^{\mathrm{MRCE}}, \hat{\Sigma}^{\mathrm{MRCE}}) \in \underset{\substack{\mathrm{B}\in\mathbb{R}^{p\times T}\\ \Sigma^{-1}\succ 0}}{\arg\min} \|\bar{Y} - X\mathrm{B}\|^2_{\Sigma^{-1}} - \log\det(\Sigma^{-1}) + \lambda\|\mathrm{B}\|_{2,1} + \mu\|\Sigma^{-1}\|_1 \ . \tag{5.44}$$

  Problem (5.44) is not convex, but can be solved heuristically (see Rothman et al. 2010 for details) by coordinate descent for the updates in $\mathrm{B}_{j:}$'s and solving a Graphical Lasso (Friedman et al., 2008) for the update in Σ^{-1}.

- MTL, the Multi-Task Lasso (Obozinski et al., 2010). It is the usual estimator used when the additive noise is supposed to be homoscedastic (with no correlation). MTL is obtained by solving

$$\hat{\mathrm{B}}^{\mathrm{MTL}} \in \underset{\mathrm{B}\in\mathbb{R}^{p\times T}}{\arg\min} \frac{1}{2nT}\|\bar{Y} - X\mathrm{B}\|^2 + \lambda\|\mathrm{B}\|_{2,1} \ . \tag{5.45}$$

Each estimator, proposed or compared to is based on an optimization problem to solve. Each optimization problem is solve with block coordinate descent, whether there is theoretical guarantees for it to converge toward a global minimum (for convex formulations, CLaR, SGCL and MTL), or not (for non-convex formulations, $\ell_{2,1}$-MLE, $\ell_{2,1}$-MLER, MRCER). The cost for the updates for each algorithm can be found in Table 5.1. The formula for the updates in $\mathrm{B}_{j:}$'s and S/Σ for each algorithm can be found in Table 5.2. Let f^{dual} be the number of updates of B for one update of S or Σ.

Table 5.1 – Algorithms cost in time summary.

	CD epoch cost	convex	dual gap cost
CLaR	$\mathcal{O}(\frac{n^3+qn^2}{f^{\mathrm{dual}}} + pn^2 + pnq)$	yes	$\mathcal{O}(rnT + p)$
SGCL	$\mathcal{O}(\frac{n^3+qn^2}{f^{\mathrm{dual}}} + pn^2 + pnq)$	yes	$\mathcal{O}(nT + p)$
$\ell_{2,1}$-MLER	$\mathcal{O}(\frac{n^3+qn^2}{f^{\mathrm{dual}}} + pn^2 + pnq)$	no	not convex
$\ell_{2,1}$-MLE	$\mathcal{O}(\frac{n^3+qn^2}{f^{\mathrm{dual}}} + pn^2 + pnq)$	no	not convex
MRCER	$\mathcal{O}(\frac{\mathcal{O}(\mathrm{glasso})}{f^{\mathrm{dual}}} + pn^2 + pnq)$	no	not convex
MTL	$\mathcal{O}(npT)$	yes	$\mathcal{O}(nT + p)$

Recalling that $\Sigma^{\text{emp}} \triangleq \frac{1}{T}(\bar{Y} - X\mathrm{B})(\bar{Y} - X\mathrm{B})^\top$ and $\Sigma^{\text{emp},r} \triangleq \frac{1}{rT}\sum_{l=1}^{r}(Y^{(l)} - X\mathrm{B})(Y^{(l)} - X\mathrm{B})^\top$, a summary of the updates in S/Σ and $\mathrm{B}_{j:}$'s for each algorithm is given in Table 5.2.

Comments on Table 5.2 The updates in S/Σ and $\mathrm{B}_{j:}$'s are given in Table 5.2. Although the updates may look similar, all the algorithms can lead to very different results, see Figures 5.6, 5.8, 5.10 and 5.12.

Table 5.2 – Algorithms updates summary

	update in $\mathrm{B}_{j:}$	update in S/Σ
CLaR	$\mathrm{B}_{j:} = \mathrm{BST}\left(\mathrm{B}_{j:} + \frac{X_{:j}^\top S^{-1}(\bar{Y} - X\mathrm{B})}{\|X_{:j}\|_{S^{-1}}^2}, \frac{\lambda n T}{\|X_{:j}\|_{S^{-1}}^2}\right)$	$S = \mathrm{ClSqrt}(\Sigma^{\text{emp},r}, \underline{\sigma})$
SGCL	$\mathrm{B}_{j:} = \mathrm{BST}\left(\mathrm{B}_{j:} + \frac{X_{:j}^\top S^{-1}(\bar{Y} - X\mathrm{B})}{\|X_{:j}\|_{S^{-1}}^2}, \frac{\lambda n T}{\|X_{:j}\|_{S^{-1}}^2}\right)$	$S = \mathrm{ClSqrt}(\Sigma^{\text{emp}}, \underline{\sigma})$
$\ell_{2,1}$-MLER	$\mathrm{B}_{j:} = \mathrm{BST}\left(\mathrm{B}_{j:} + \frac{X_{:j}^\top \Sigma^{-1}(\bar{Y} - X\mathrm{B})}{\|X_{:j}\|_{\Sigma^{-1}}^2}, \frac{\lambda n T}{\|X_{:j}\|_{\Sigma^{-1}}^2}\right)$	$\Sigma = \mathrm{Cl}(\Sigma^{\text{emp},r}, \underline{\sigma}^2)$
$\ell_{2,1}$-MLE	$\mathrm{B}_{j:} = \mathrm{BST}\left(\mathrm{B}_{j:} + \frac{X_{:j}^\top \Sigma^{-1}(\bar{Y} - X\mathrm{B})}{\|X_{:j}\|_{\Sigma^{-1}}^2}, \frac{\lambda n T}{\|X_{:j}\|_{\Sigma^{-1}}^2}\right)$	$\Sigma = \mathrm{Cl}(\Sigma^{\text{emp}}, \underline{\sigma}^2)$
MRCER	$\mathrm{B}_{j:} = \mathrm{BST}\left(\mathrm{B}_{j:} + \frac{X_{:j}^\top \Sigma^{-1}(\bar{Y} - X\mathrm{B})}{\|X_{:j}\|_{\Sigma^{-1}}^2}, \frac{\lambda n T}{\|X_{:j}\|_{\Sigma^{-1}}^2}\right)$	$\Sigma = \mathrm{glasso}(\Sigma^{\text{emp},r}, \mu)$
MTL	$\mathrm{B}_{j:} = \mathrm{BST}\left(\mathrm{B}_{j:} + \frac{X_{:j}^\top (\bar{Y} - X\mathrm{B})}{\|X_{:j}\|^2}, \frac{\lambda n T}{\|X_{:j}\|^2}\right)$	no update in S/Σ

5.3.1 Synthetic data

Here we demonstrate the ability of our estimator to recover the support *i.e.,* the ability to identify the predictive features. There are $n = 150$ observations, $p = 500$ features, $T = 100$ tasks. The design X is random with Toeplitz-correlated features with parameter $\rho_X = 0.6$ (correlation between $X_{:i}$ and $X_{:j}$ is $\rho_X^{|i-j|}$), and its columns have unit Euclidean norm. The true coefficient B^* has 30 non-zeros rows whose entries are independent and normally centered distributed. S^* is a Toeplitz matrix with parameter ρ_S. The SNR is fixed and constant across all repetitions

$$\mathrm{SNR} \triangleq \|X\mathrm{B}^*\|/\sqrt{r}\|X\mathrm{B}^* - \bar{Y}\| \ . \tag{5.46}$$

For Figures 5.1 to 5.3, the figure of merit is the ROC curve, *i.e.,* the true positive rate (TPR) against the false positive rate (FPR). For each estimator, the ROC curve is obtained by varying the value of the regularization parameter λ on a geometric grid of 160 points, from $\lambda_{\max}$ (specific to each algorithm) to $\lambda_{\min}$, the latter also being estimator specific and chosen to obtain a FPR larger than 0.4.

Influence of noise structure. Figure 5.1 represents the ROC curves for different values of ρ_S. As ρ_S increases, the noise becomes more and more correlated. From left to right, the performance of CLaR, SGCL, MRCER, $\ell_{2,1}$-MRCE, and $\ell_{2,1}$-MLER increases as they are designed to exploit correlations in the noise, while the performance of MTL decreases, as its i.i.d. Gaussian noise model becomes less and less valid.

Influence of SNR. On Figure 5.2 we can see that when the SNR is high (left), all estimators (except $\ell_{2,1}$-MLE) reach the (0, 1) point. This means that for each algorithm (except $\ell_{2,1}$-MLE), there exists a λ such that the estimated support is exactly the true

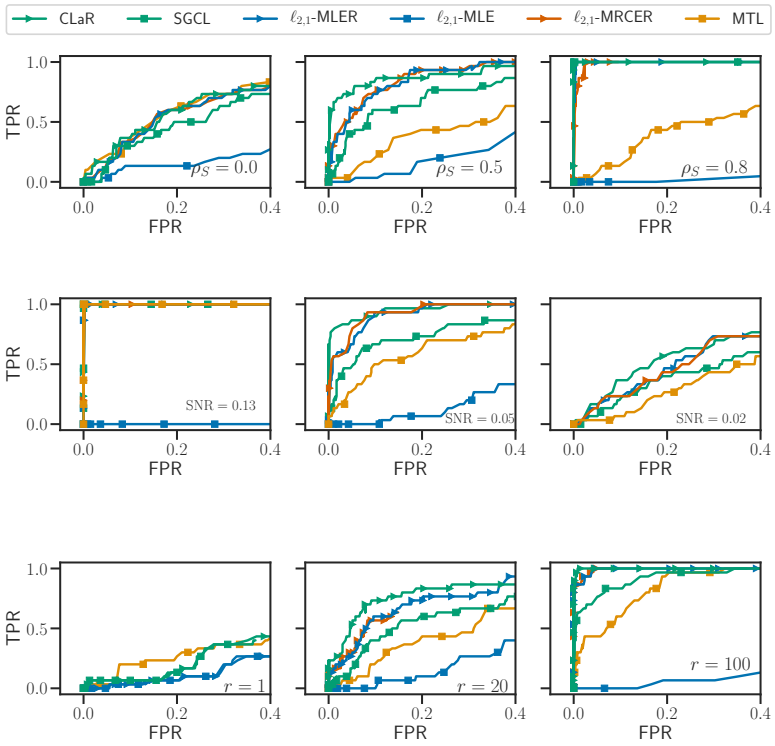


Figure 5.1 – *Influence of noise structure.* ROC curves of support recovery ($\rho_X = 0.6$, SNR $= 0.03$, $r = 20$) for different ρ_S values.

Figure 5.2 – *Influence of SNR.* ROC curves of support recovery ($\rho_X = 0.6$, $\rho_S = 0.4$, $r = 20$) for different SNR values.

Figure 5.3 – *Influence of the number of repetitions.* ROC curves of support recovery ($\rho_X = 0.6$, SNR $= 0.03$, $\rho_S = 0.4$) for different r values.

one. However, when the SNR decreases (middle), the performance of SGCL and MTL starts to drop, while that of CLaR, $\ell_{2,1}$-MLER and MRCER remains stable (CLaR performing better), highlighting their capacity to leverage multiple repetitions of measurements to handle the noise structure. Finally, when the SNR is too low (right), all algorithms perform poorly, but CLaR, $\ell_{2,1}$-MLER and MRCER still performs better.

Influence of the number of repetitions. Figure 5.3 shows ROC curves of all compared approaches for different r, starting from $r = 1$ (left) to 100 (right). Even with $r = 20$ (middle) CLaR outperforms the other estimators, and when $r = 100$ CLaR can better leverage the large number of repetitions.

5.3.2 Realistic data

We now evaluate the estimators on realistic magneto- and electroencephalography (M/EEG) data. The M/EEG recordings measure the electrical potential and magnetic fields induced by the active neurons. Data are time series of length T with n sensors and p sources mapping to locations in the brain. Because the propagation of the electromagnetic fields is driven by the linear Maxwell equations, one can assume that the relation between the measurements $Y^{(1)}, \dots, Y^{(r)}$ and the amplitudes of sources in the brain B^* is linear.

The M/EEG inverse problem consists in identifying B^*. Because of the limited number of sensors (a few hundreds in practice), as well as the physics of the problem, the M/EEG inverse problem is severely ill-posed and needs to be regularized. Moreover, the experiments being usually short (less than 1 s.) and focused on specific cognitive functions, the number of active sources is expected to be small, *i.e.,* B^* is assumed to be row-sparse. This plausible biological assumption motivates the framework of Section 5.2

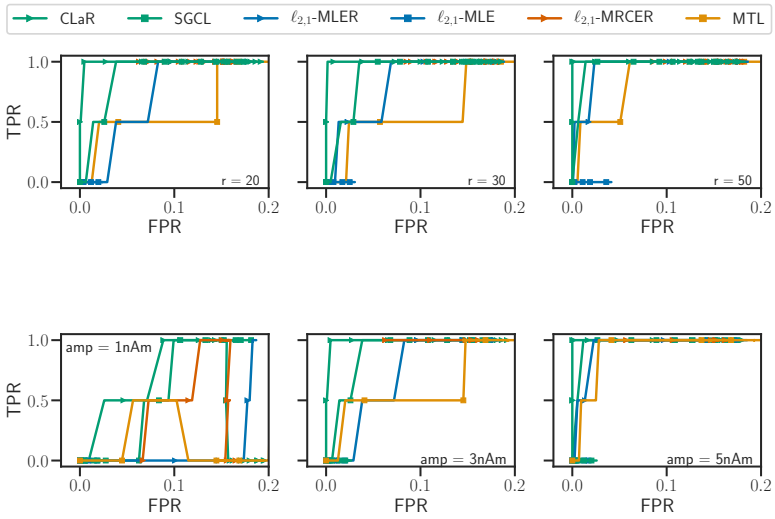


Figure 5.4 – *Influence of the number of repetitions.* ROC curves with empirical X and S and simulated B^* (amp $= 2\,$nA.m), for different number of repetitions.

Figure 5.5 – *Amplitude influence.* ROC curves with empirical X and S and simulated B^* ($r = 50$), for different amplitudes of the signal.

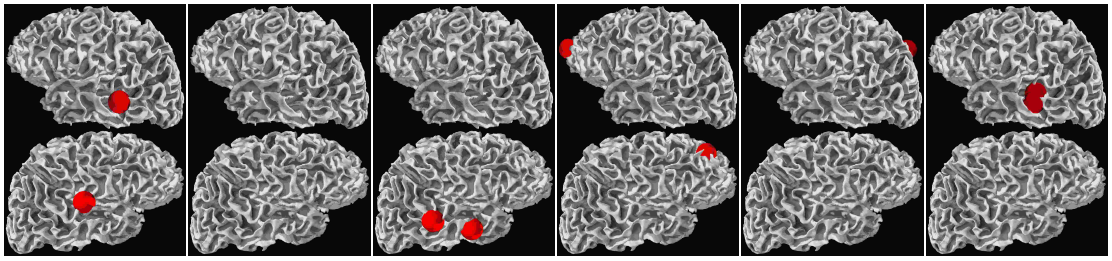
(Ou et al., 2009).

Dataset. We use the *sample* dataset [4] from the MNE software (Gramfort et al., 2014). The experimental conditions here are auditory stimulations in the right or left ear, leading to two main foci of activations in bilateral auditory cortices (*i.e.,* 2 non-zeros rows for B^*). For this experiment, we keep only the gradiometer magnetic channels. After removing one channel corrupted by artifacts, this leads to $n = 203$ signals. The length of the temporal series is $T = 100$, and the data contains $r = 50$ repetitions. We choose a source space of size $p = 1281$ which corresponds to about 1 cm distance between neighboring sources. The orientation is fixed, and normal to the cortical mantle.

Realistic MEG data simulations. We use here true empirical values for X and S by solving Maxwell equations and taking an empirical co-standard deviation matrix. To generate realistic MEG data we simulate neural responses B^* with 2 non-zeros rows corresponding to areas known to be related to auditory processing (Brodmann area 22). Each non-zero row of B^* is chosen as a sinusoidal signal with realistic frequency (5 Hz) and amplitude (amp $\sim 1 - 10$ nAm). We finally simulate r MEG signals $Y^{(l)} = XB^* + S^*E^{(l)}$, $E^{(l)}$ being matrices with i.i.d. normal entries.

Preprocessing steps for realistic and real data. When using multi-modal data without whitening, one has to rescale properly data, indeed data needs to have the same order of magnitude, otherwise some mode (for example EEG data) could be (almost) completely ignored by the optimization algorithm. The preprocessing pipeline used to rescale realistic data (Figures 5.4 and 5.5) and real data (Figures 5.6, 5.8, 5.10 and 5.12) is described in Algorithm 5.2.

[4] publicly available real M/EEG data recorded after auditory or visual stimulations.



(a) CLaR (b) SGCL (c) $\ell_{2,1}$-MLER (d) $\ell_{2,1}$-MLE (e) MRCER (f) MTL

Figure 5.6 – *Real data, left auditory stimulations ($n = 102$, $p = 7498$, $T = 76$, $r = 63$)* Sources found in the left hemisphere (top) and the right hemisphere (bottom) after left auditory stimulations.

Algorithm 5.2 PREPROCESSING STEPS FOR REALISTIC AND REAL DATA

input : $X, Y^{(1)}, \dots, Y^{(r)}$

// rescale each line of X

for $i = 1, \dots, n$ **do**

  for $l = 1, \dots, r$ **do**

    $Y^{(l)}_{i:} \leftarrow Y^{(l)}_{i:} / \|X_{i:}\|$

  $X_{i:} \leftarrow X_{i:} / \|X_{i:}\|$

// rescale each column of X

for $j = 1, \dots, p$ **do**

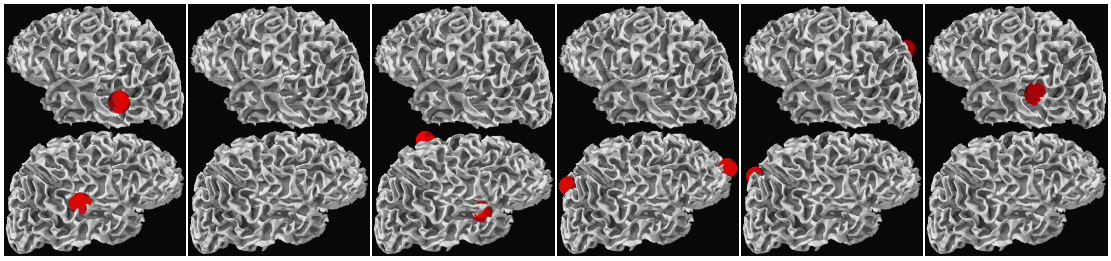
  $X_{:j} \leftarrow X_{:j} / \|X_{:j}\|$

return $X, Y^{(1)}, \dots, Y^{(r)}$

The signals being contaminated with correlated noise, if one wants to use homoscedastic solvers it is necessary to whiten the data first (and thus to have an estimation of the covariance matrix, the later often being unknown). In this experiment we demonstrate that without this whitening process, the homoscedastic solver MTL fails, as well as solvers which does not take in account the repetitions: SGCL and $\ell_{2,1}$-MLE. In this scenario CLaR, $\ell_{2,1}$-MLER and MRCER do succeed in recovering the sources, CLaR leading to the best results. As for the synthetic data, Figures 5.4 and 5.5 are obtained by varying the estimator-specific regularization parameter λ from $\lambda_{\max}$ to $\lambda_{\min}$ on a geometric grid.

Amplitude influence. Figure 5.5 shows ROC curves for different values of the amplitude of the signal. When the amplitude is high (right), all the algorithms perform well, however when the amplitude decreases (middle) only CLaR leads to good results, almost hitting the $(0, 1)$ corner. When the amplitude gets lower (left) all algorithms perform worse, CLaR still yielding the best results.

Influence of the number of repetitions. Figure 5.4 shows ROC curves for different number of repetitions r. When the number of repetitions is high (right, $r = 50$), the algorithms taking into account all the repetitions (CLaR, $\ell_{2,1}$-MLER, MRCER) perform best, almost hitting the $(0, 1)$ corner, whereas the algorithms which do not take into account all the repetitions ($\ell_{2,1}$-MLE, MTL, SGCL) perform poorly. As soon as the number of repetitions decreases (middle and left) the performance of all the algorithms except CLaR starts dropping severely. CLaR is once again the algorithm taking the most advantage of the number of repetitions.



(a) CLaR (b) SGCL (c) $\ell_{2,1}$-MLER (d) $\ell_{2,1}$-MLE (e) MRCER (f) MTL

Figure 5.7 – *Real data, right auditory stimulations ($n = 102$, $p = 7498$, $T = 76$, $r = 33$)* Sources found in the left hemisphere (top) and the right hemisphere (bottom) after right auditory stimulations.

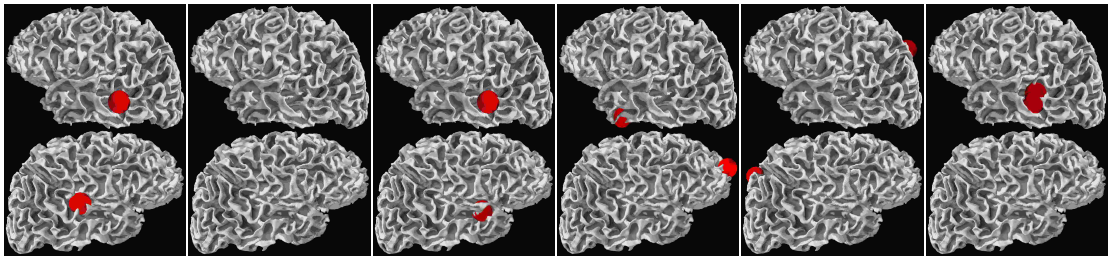
5.3.3 Real M/EEG data

As before, we use the *sample* dataset, keeping only the magnetometer magnetic channels ($n = 102$ signals). We choose a source space of size $p = 7498$ (about 5 mm between neighboring sources). The orientation is fixed, and normal to the cortical mantle. As for realistic data, X is the empirical design matrix, but this time we use the empirical measurements $Y^{(1)}, \dots, Y^{(r)}$. As two sources are expected (one in each hemisphere, in bilateral auditory cortices), we vary λ by dichotomy between $\lambda_{\max}$ (returning 0 sources) and a $\lambda_{\min}$ (returning more than 2 sources), until finding a λ giving exactly 2 sources. Results are provided in Figures 5.6 and 5.7. Running times of each algorithm are of the same order of magnitude and can be found in Figure 5.14.

Audiroy stimulations. *Comments on Figure 5.6, left auditory stimulations.* Sources found by the algorithms are represented by red spheres. SGCL, $\ell_{2,1}$-MLE and MRCER completely fail, finding sources that are not in the auditory cortices at all (SGCL sources are deep, thus not in the auditory cortices, and cannot be seen). MTL and $\ell_{2,1}$-MLER do find sources in auditory cortices, but only in one hemisphere (left for MTL and right for $\ell_{2,1}$-MLER). CLaR is the only one that finds one source in each hemisphere in the auditory cortices as expected.

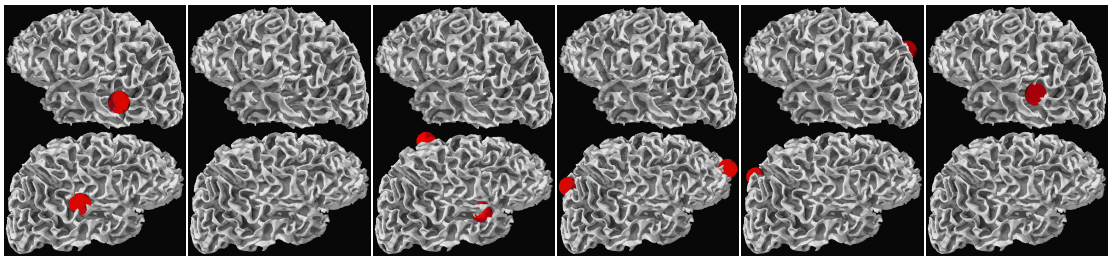
Comments on Figure 5.7, right auditory stimulations. In this experiment we only keep $r = 33$ repetitions (out of 65 available) and it can be seen that only CLaR finds correct sources, MTL finds sources only in one hemisphere and all the other algorithms do find sources that are not in the auditory cortices. This highlights the robustness of CLaR, even with a limited number of repetitions, confirming previous experiments (see Figure 5.3).

Figures 5.8 and 5.9 show the solution given by each algorithm on real data after right auditory stimulations. As two sources are expected (one in each hemisphere, in bilateral auditory cortices), we vary λ by dichotomy between $\lambda_{\max}$ (returning 0 sources) and a $\lambda_{\min}$ (returning more than 2 sources), until finding a lambda giving exactly 2 sources. Figure 5.8 (*resp.* Figure 5.9) shows the solution given by the algorithms taking in account all the repetitions (*resp.* only half of the repetitions). When the number of repetitions is high (Figure 5.8) only CLaR and $\ell_{2,1}$-MLER find one source in each auditory cortex, MTL does find sources only in one hemisphere, all the other algorithms fail by finding sources not in the auditory cortices at all. Moreover when the number of repetitions is decreasing (Figure 5.9) $\ell_{2,1}$-MLER fails and only CLaR does find 2 sources, one in



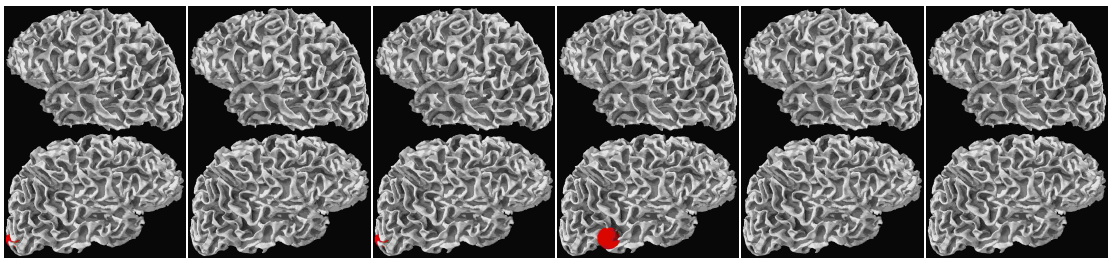
(a) CLaR (b) SGCL (c) $\ell_{2,1}$-MLER (d) $\ell_{2,1}$-MLE (e) MRCER (f) MTL

Figure 5.8 – *Real data ($n = 102$, $p = 7498$, $T = 76$, $r = 65$)* Sources found in the left hemisphere (top) and the right hemisphere (bottom) after right auditory stimulations.



(a) CLaR (b) SGCL (c) $\ell_{2,1}$-MLER (d) $\ell_{2,1}$-MLE (e) MRCER (f) MTL

Figure 5.9 – *Real data ($n = 102$, $p = 7498$, $T = 76$, $r = 33$)* Sources found in the left hemisphere (top) and the right hemisphere (bottom) after right auditory stimulations.

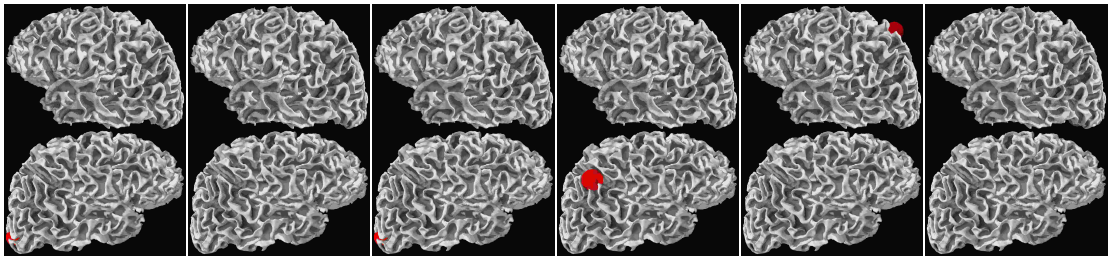


(a) CLaR (b) SGCL (c) $\ell_{2,1}$-MLER (d) $\ell_{2,1}$-MLE (e) MRCER (f) MTL

Figure 5.10 – *Real data ($n = 102$, $p = 7498$, $T = 48$, $r = 71$)* Sources found in the left hemisphere (top) and the right hemisphere (bottom) after left visual stimulations.

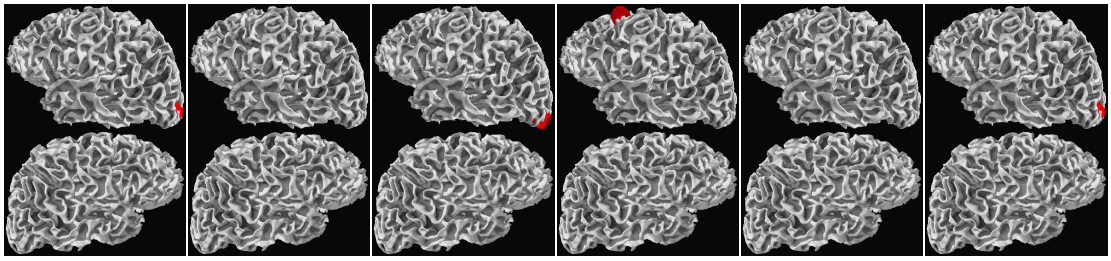
each hemisphere. Once again CLaR is more robust and performs better, even when the number of repetitions is low.

Visual stimulations. Figures 5.10 and 5.11 show the results for each algorithm after left visual stimulations. As one source is expected (in the right hemisphere), we vary λ by dichotomy between $\lambda_{\max}$ (returning 0 sources) and a $\lambda_{\min}$ (returning more than 1 sources), until finding a lambda giving exactly 1 source. When the number of repetitions is high (Figure 5.10) only CLaR and $\ell_{2,1}$-MLER do find a source in the visual cortex. When the number of repetitions decreases, CLaR and $\ell_{2,1}$-MLER still find one source in the visual cortex, other algorithms fail. This highlights this importance of taking into account the repetitions.



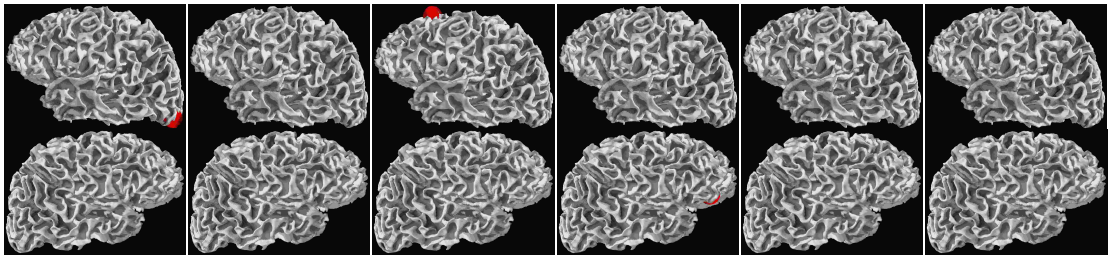
(a) CLaR (b) SGCL (c) $\ell_{2,1}$-MLER (d) $\ell_{2,1}$-MLE (e) MRCER (f) MTL

Figure 5.11 – *Real data ($n = 102$, $p = 7498$, $T = 48$, $r = 36$)* Sources found in the left hemisphere (top) and the right hemisphere (bottom) after left visual stimulations.



(a) CLaR (b) SGCL (c) $\ell_{2,1}$-MLER (d) $\ell_{2,1}$-MLE (e) MRCER (f) MTL

Figure 5.12 – *Real data ($n = 102$, $p = 7498$, $T = 48$, $r = 61$)* Sources found in the left hemisphere (top) and the right hemisphere (bottom) after right visual stimulations.



(a) CLaR (b) SGCL (c) $\ell_{2,1}$-MLER (d) $\ell_{2,1}$-MLE (e) MRCER (f) MTL

Figure 5.13 – *Real data ($n = 102$, $p = 7498$, $T = 48$, $r = 31$)* Sources found in the left hemisphere (top) and the right hemisphere (bottom) after right visual stimulations.

Figures 5.12 and 5.13 show the results for each algorithm after right visual stimulations. As one source is expected (in the left hemisphere), we vary λ by dichotomy between $\lambda_{\max}$ (returning 0 sources) and a $\lambda_{\min}$ (returning more than 1 sources), until finding a lambda giving exactly 1 source. When the number of repetitions is high (Figure 5.12) only CLaR, $\ell_{2,1}$-MLER and MTL do find a source in the visual cortex. When the number of repetitions decreases (Figure 5.13), only CLaR finds one source in the visual cortex, other algorithms fail. This highlights once again the robustness of CLaR, even with a limited number of repetitions.

Time comparison. The goal of this experiment is to show that our algorithm (CLaR) is as costly as a Multi-Task Lasso or other competitors (in the M/EEG context, *i.e.,* n not too large). The time taken by each algorithm to produce Figure 5.6 (real data, left auditory stimulations) is given in Figure 5.14. In this experiment the tolerance is set to tol$=10^{-3}$, the safe stopping criterion is duality gap $<$ tol (only avail-

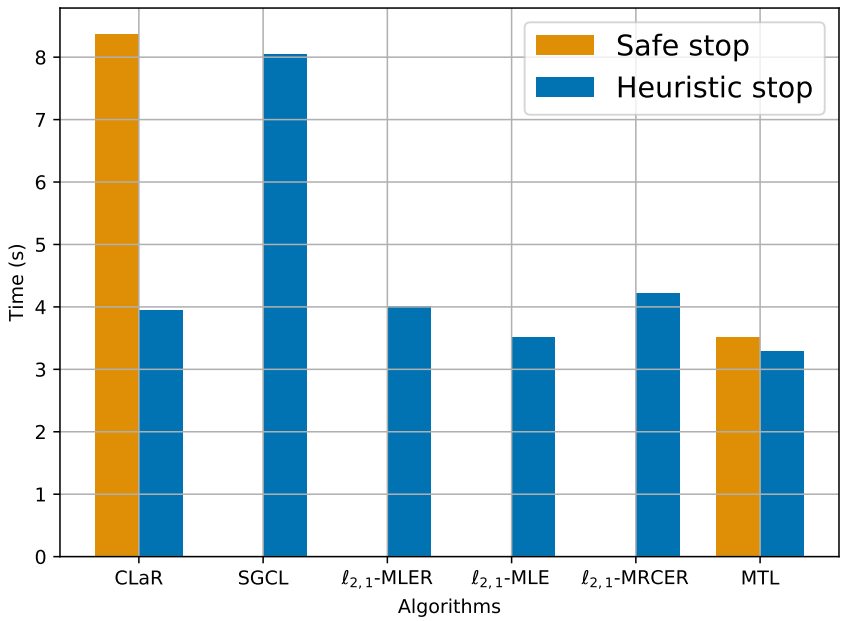


Figure 5.14 – *Time comparison, real data,* $n = 102$, $p = 7498$, $T = 54$, $r = 56$ Time for each algorithm to produce Figure 5.6.

able for convex optimization problems). The heuristic stopping criterion is "if the objective do not decrease enough anymore then stop" *i.e.,* if objective$(\mathrm{B}^{(t)}, \Sigma^{(t)}) -$ objective$(\mathrm{B}^{(t+1)}, \Sigma^{(t+1)}) <$ tol$/10$ then stop. The safe stopping criterion is only available for CLaR, SGCL and MTL (it takes too much time *i.e.,* more than 10min for SGCL to have a duality gap under the fixed tol, so we remove it).

Comment on Figure 5.14 Figure 5.14 shows that if we use the heuristic stopping criterion, CLaR is as fast the other algorithm. In addition CLaR has a safe stopping criterion which only take 2 to 3 more time than the heuristic one (less than 10sec).

Conclusion. This work introduces CLaR, a sparse estimator for multitask regression. It is designed to handle correlated Gaussian noise in the context of repeated observations, a standard framework in applied sciences such as neuro-imaging. The resulting optimization problem can be solved efficiently with state-of-the-art convex solvers, and the algorithmic cost is the same as for single repetition data. The theory of smoothing connects CLaR to the Schatten 1-Lasso in a principled manner, which opens the way to the use of more sophisticated data fitting terms. The benefits of CLaR for support recovery in the presence of non-white Gaussian noise were extensively evaluated against a large number of competitors, both on simulations and on empirical MEG data.

Part III

Hyperparameter selection, the bilevel way

6 Hyperparameter optimization

Contents

In Chapters 4 and 5 we investigated pivotal estimators, for which the regularization parameter admits a closed-form formula. However, as illustrated in Chapter 1, this route relies on strong hypotheses usually unrealistic on real data (Figure 1.5), and users still have to resort to model calibration (see Section 5.3). In this chapter we investigate another route: hyperparameter optimization. Indeed, finding the optimal hyperparameters of a model can be cast as a bilevel optimization problem, typically solved using zero-order techniques. In this work we study first-order methods when the inner optimization problem is convex but nonsmooth. Capitalizing on model identification (Theorem 2.13) and local linear convergence (Theorem 2.16) results from Chapter 2, we show that the forward-mode differentiation of proximal gradient descent and proximal coordinate descent yield sequences of Jacobians converging toward the exact Jacobian. Using implicit differentiation, we show it is possible to leverage the non-smoothness of the inner problem to speed up the computation. Finally, we provide a bound on the error made on the hypergradient when the inner optimization problem is solved approximately. Results on regression and classification problems reveal computational benefits for hyperparameter optimization, especially when multiple hyperparameters are required.

This chapter is based on the following works, one accepted to ICML 2020, the other currently under review for the Journal of Machine Learning Research:

- **Q. Bertrand**, Q. Klopfenstein, M. Blondel, S. Vaiter, A. Gramfort, and J. Salmon. Implicit differentiation of Lasso-type models for hyperparameter optimization. *ICML*, 2020

- **Q. Bertrand**, Q. Klopfenstein, M. Massias, M. Blondel, S. Vaiter, A. Gramfort, and J. Salmon. Implicit differentiation for fast hyperparameter selection in non-smooth convex learning. *arXiv preprint arXiv:2105.01637*, 2021

6.1 Introduction

Almost all models in machine learning require at least one hyperparameter, the tuning of which drastically affects accuracy. This is the case for many popular estimators, where the regularization hyperparameter controls the trade-off between a data fidelity term and a regularization term. Such estimators, including Ridge regression (Hoerl and Kennard, 1970), Lasso (Tibshirani, 1996; Chen et al., 1998), elastic net (Zou and Hastie, 2005), sparse logistic regression (Koh et al., 2007), support-vector machine/SVM (Boser et al., 1992; Platt, 1999) are often cast as an optimization problem (Table 6.1)

Table 6.1 – Examples of nonsmooth inner problems as in (6.1).

Inner problem, Φ	$f(\beta)$	$g_j(\beta_j, \lambda)$	$e^{\lambda_{\max}}$
Lasso	$\frac{1}{2n}\|y - X\beta\|^2$	$e^{\lambda}\|\beta_j\|$	$\frac{1}{n}\|X^\top y\|_\infty$
elastic net	$\frac{1}{2n}\|y - X\beta\|^2$	$e^{\lambda_1}\|\beta_j\| + \frac{1}{2}e^{\lambda_2}\beta_j^2$	$\frac{1}{n}\|X^\top y\|_\infty$
sparse log. reg.	$\frac{1}{n}\sum_{i=1}^n \ln(1 + e^{-y_i X_{i:}\beta})$	$e^{\lambda}\|\beta_j\|$	$\frac{1}{2n}\|X^\top y\|_\infty$
dual SVM	$\frac{1}{2}\|(y \odot X)^\top \beta\|^2 - \sum_{j=1}^p \beta_j$	$\iota_{[0, e^\lambda]}(\beta_j)$	–

$$\hat{\beta}^{(\lambda)} \in \underset{\beta \in \mathbb{R}^p}{\arg\min}\, \Phi(\beta, \lambda) \triangleq f(\beta) + \underbrace{\sum_{j=1}^p g_j(\beta_j, \lambda)}_{\triangleq g(\beta, \lambda)}\ , \tag{6.1}$$

with smooth $f : \mathbb{R}^p \to \mathbb{R}$ (*i.e.,* with Lipschitz gradient), proper closed convex (possibly nonsmooth) functions $g_j(\cdot, \lambda)$, and a regularization hyperparameter $\lambda \in \mathbb{R}^r$. In the examples of Table 6.1, the computation of f involves a design matrix $X \in \mathbb{R}^{n \times p}$; and the cost of computing $\nabla f(\beta)$ is $\mathcal{O}(np)$. In the SVM example, since we consider the dual problem, we chose to reverse the roles of n and p to enforce $\beta \in \mathbb{R}^p$. We often drop the λ dependency and write $\hat{\beta}$ instead of $\hat{\beta}^{(\lambda)}$ when it is clear from context.

For a fixed λ, the issue of solving efficiently Problem (6.1) has been largely explored. If the functions g_j are smooth, one can use solvers such as L-BFGS (Liu and Nocedal, 1989), SVRG (Johnson and Zhang, 2013; Zhang et al., 2013), or SAGA (Defazio et al., 2014). When the functions g_j are nonsmooth, Problem (6.1) can be tackled efficiently with stochastic algorithms (Pedregosa et al., 2017) or using working set methods (Fan and Lv, 2008; Tibshirani et al., 2012) combined with coordinate descent (Tseng and Yun, 2009a), see overview by Massias et al. (2020b). The question of *model selection*, *i.e.,* how to select the hyperparameter $\lambda \in \mathbb{R}^r$ (potentially multidimensional), is more open, especially when the dimension r of the regularization hyperparameter λ is large.

Table 6.2 – Examples of outer criteria used for hyperparameter selection.

Criterion	Problem type	Criterion $\mathcal{C}(\beta)$
Hold-out mean squared error	Regression	$\frac{1}{n}\|y^{\text{val}} - X^{\text{val}}\beta\|^2$
Stein unbiased risk estimate (SURE)[1]	Regression	$\|y - X\beta\|^2 - n\sigma^2 + 2\sigma^2\text{dof}(\beta)$
Hold-out logistic loss	Classification	$\frac{1}{n}\sum_{i=1}^{n} \ln(1 + e^{-y_i^{\text{val}} X_{i:}^{\text{val}}\beta})$
Hold-out smoothed Hinge loss[2]	Classification	$\frac{1}{n}\sum_{i=1}^{n} \ell(y_i^{\text{val}}, X_{i:}^{\text{val}}\beta)$

For the Lasso, a broad literature has been devoted to parameter tuning. Under strong hypothesis on the design matrix X, it is possible to derive guidelines for the setting of the regularization parameter λ (Lounici, 2008; Bickel et al., 2009; Belloni et al., 2011). Unfortunately, these guidelines rely on quantities which are typically unknown in practice, and Lasso users still have to resort to other techniques to select the hyperparameter λ.

A popular approach for hyperparameter selection is *hyperparameter optimization* (Kohavi and John, 1995; Hutter et al., 2015; Feurer and Hutter, 2019): one selects the hyperparameter λ such that the regression coefficients $\hat{\beta}^{(\lambda)}$ minimize a given criterion $\mathcal{C} : \mathbb{R}^p \to \mathbb{R}$. Here $\mathcal{C}$ should ensure good generalization, or avoid overcomplex models. Common examples (see Table 6.2) include the hold-out loss (Devroye and Wagner, 1979), the cross-validation loss (CV, Stone and Ramer 1965, see Arlot and Celisse 2010 for a survey), the AIC (Akaike, 1974), BIC (Schwarz, 1978) or SURE (Stein, 1981) criteria. Formally, the hyperparameter optimization problem is a bilevel optimization problem (Colson et al., 2007):

$$\begin{aligned} &\underset{\lambda \in \mathbb{R}^r}{\arg\min} \left\{ \mathcal{L}(\lambda) \triangleq \mathcal{C}\left(\hat{\beta}^{(\lambda)}\right) \right\} \\ &\text{s.t. } \hat{\beta}^{(\lambda)} \in \underset{\beta \in \mathbb{R}^p}{\arg\min}\, \Phi(\beta, \lambda) \ . \end{aligned} \tag{6.2}$$

Popular approaches to solve (the generally non-convex) Problem (6.2) include zero-order optimization (gradient-free) techniques such as grid-search, random-search (Rastrigin, 1963; Bergstra and Bengio, 2012; Bergstra et al., 2013) or Sequential Model-Based Global Optimization (SMBO), often referred to as Bayesian optimization (Mockus, 1989; Jones et al., 1998; Forrester et al., 2008; Brochu et al., 2010; Snoek et al., 2012). Grid-search is a naive discretization of Problem (6.2). It consists in evaluating the outer function $\mathcal{L}$ on a grid of hyperparameters, solving one inner optimization Problem (6.1) for each λ in the grid (see Figure 6.1). For each inner problem solution $\hat{\beta}^{(\lambda)}$, the criterion $\mathcal{C}(\hat{\beta}^{(\lambda)})$ is evaluated, and the model achieving the lowest value is selected. Random-search has a similar flavor, but one randomly selects where the criterion must be evaluated. Finally, SMBO models the objective function $\mathcal{L}$ via a function amenable to uncertainty estimates on its predictions such as a Gaussian process. Hyperparameter values are chosen iteratively to maximize a function such as the expected improvement as described, *e.g.,* by Bergstra et al. (2011). However, these zero-order methods share a common drawback: they scale exponentially with the dimension of the search space (Nesterov, 2004, Sec. 1.1.2).

[1] For a linear model $y = X\beta + \varepsilon$, with $\varepsilon \sim \mathcal{N}(0, \sigma^2)$, the degree of freedom (dof, Efron 1986) is defined as $\text{dof}(\beta) = \sum_{i=1}^{n} \text{cov}(y_i, (X\beta)_i)/\sigma^2$.

[2] The smoothed Hinge loss is given by $\ell(x) = \frac{1}{2} - x$ if $x \leq 0$, $\frac{1}{2}(1-x)^2$ if $0 \leq x \leq 1$ and 0 else.

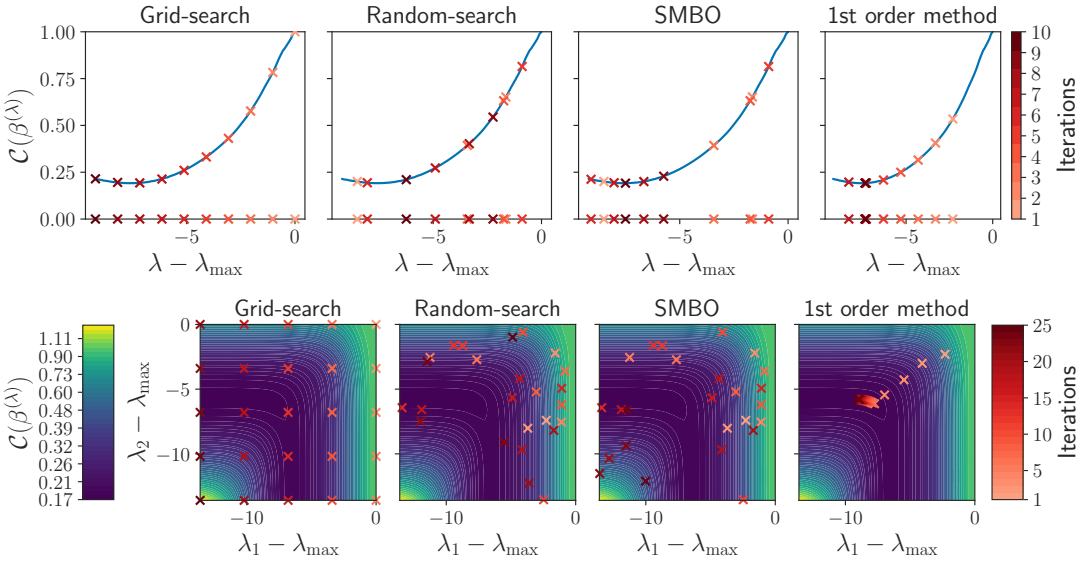


Figure 6.1 – **5-fold cross-validation error** $\mathcal{C}(\beta^{(\lambda)})$: (top) Lasso CV error with respect to λ for multiple hyperparameter optimization methods on the *real-sim* dataset, and (bottom) elastic net CV error with respect to λ_1 and λ_2 on the *rcv1* dataset. Crosses represent the 10 (top) or 25 (bottom) first error evaluations for each method.

When the hyperparameter space is continuous and the regularization path $\lambda \mapsto \hat{\beta}^{(\lambda)}$ is well-defined and weakly differentiable, first-order optimization methods are well suited to solve the bilevel optimization Problem (6.2). Using the chain rule, the gradient of $\mathcal{L}$ with respect to λ, also referred to as the *hypergradient*, evaluates to

$$\nabla_\lambda \mathcal{L}(\lambda) = \hat{\mathcal{J}}_{(\lambda)}^\top \nabla \mathcal{C}(\hat{\beta}^{(\lambda)}) \ , \tag{6.3}$$

with $\hat{\mathcal{J}}_{(\lambda)} \in \mathbb{R}^{p \times r}$ the *Jacobian* of the function $\lambda \mapsto \hat{\beta}^{(\lambda)}$,

$$\hat{\mathcal{J}}_{(\lambda)} \triangleq \begin{pmatrix} \frac{\partial \hat{\beta}_1^{(\lambda)}}{\partial \lambda_1} & \cdots & \frac{\partial \hat{\beta}_1^{(\lambda)}}{\partial \lambda_r} \\ \vdots & \cdots & \vdots \\ \frac{\partial \hat{\beta}_p^{(\lambda)}}{\partial \lambda_1} & \cdots & \frac{\partial \hat{\beta}_p^{(\lambda)}}{\partial \lambda_r} \end{pmatrix} . \tag{6.4}$$

An important challenge of applying first-order methods to solve Problem (6.2) is evaluating the hypergradient in Equation (6.3). There are three main algorithms to compute the hypergradient $\nabla_\lambda \mathcal{L}(\lambda)$: implicit differentiation (Larsen et al., 1996; Bengio, 2000) and automatic differentiation using the reverse-mode (Linnainmaa, 1970; LeCun et al., 1998) or the forward-mode (Wengert, 1964; Deledalle et al., 2014; Franceschi et al., 2017). As illustrated in Figure 6.1, once the hypergradient in Equation (6.3) has been computed, one can solve Problem (6.2) with first-order schemes, *e.g.,* gradient descent.

Contributions. We are interested in tackling the bilevel optimization Problem (6.2), with a nonsmooth inner optimization Problem (6.1). More precisely,

- We show that classical algorithms used to compute hypergradients for smooth inner problem have theoretically grounded nonsmooth counterparts. We provide in Theorem 6.9 an implicit differentiation formula for nonsmooth optimization

problems. We obtain in Theorem 6.13, for the first time in the nonsmooth case, error bounds with respect to the hypergradient when the inner problem and the linear system involved are only solved approximately. We obtain in Theorem 6.12 convergence rates on the hypergradient for iterative differentiation of nonsmooth optimization problems.

- Based on the former contributions we propose an algorithm to tackle Problem (6.2). We develop an efficient implicit differentiation algorithm to compute the hypergradient in Equation (6.3), leveraging the sparsity of the Jacobian and enabling the use of state-of-the-art solvers (Algorithm 6.5). We combine in Algorithm 6.6 this fast hypergradient computation with a gradient descent scheme to solve Problem (6.2).

- We provide extensive experiments on diverse datasets and estimators (Section 6.4). We first show that implicit differentiation significantly outperforms other hypergradient methods (Section 6.4.1). Then, leveraging sparsity, we illustrate computational benefits of first-order optimization with respect to zero-order techniques for solving Problem (6.2) on Lasso, elastic net and multiclass logistic regression (Section 6.4.2).

- We release our implementation as a high-quality, documented and tested Python package: `https://github.com/qb3/sparse-ho`.

Notation. The regularization parameter, possibly multivariate, is denoted by $\lambda = (\lambda_1, \dots, \lambda_r)^\top \in \mathbb{R}^r$. We denote $\hat{\mathcal{J}}_{(\lambda)} \triangleq (\nabla_\lambda \hat{\beta}_1^{(\lambda)}, \dots, \nabla_\lambda \hat{\beta}_p^{(\lambda)})^\top \in \mathbb{R}^{p \times r}$ the weak Jacobian (Evans and Gariepy, 1992) of $\hat{\beta}^{(\lambda)}$ with respect to λ. For a function f, its gradient restricted to the indices in a set S is denoted $\nabla_S f$. For a function $\psi : \mathbb{R}^p \times \mathbb{R}^r \mapsto \mathbb{R}^p$, we denote $\partial_z \psi$ the weak Jacobian with respect to the first variable and $\partial_\lambda \psi$ the weak Jacobian with respect to the second variable. The proximal operator of $g(\cdot, \lambda)$ can be seen as such a function ψ of β and λ (see Table 6.1 for examples):

$$\begin{aligned} \mathbb{R}^p \times \mathbb{R}^r &\to \mathbb{R}^p \\ (z, \lambda) &\mapsto \operatorname{prox}_{g(\cdot,\lambda)}(z) = \psi(z, \lambda) \ . \end{aligned}$$

In this case we denote $\partial_z \operatorname{prox}_{g(\cdot,\lambda)} \triangleq \partial_z \psi$ and $\partial_\lambda \operatorname{prox}_{g(\cdot,\lambda)} \triangleq \partial_\lambda \psi$. Since we consider only separable penalties $g(\cdot, \lambda)$, $\partial_z \operatorname{prox}_{g(\cdot,\lambda)}$ is a diagonal matrix, so to make notation lighter, we write $\partial_z \operatorname{prox}_{g(\cdot,\lambda)}$ for its diagonal. We thus have

$$\begin{aligned} \partial_z \operatorname{prox}_{g(\cdot,\lambda)} = (\partial_z \operatorname{prox}_{g_j(\cdot,\lambda)})_{j \in [p]} \in \mathbb{R}^p \quad \text{(by separability of } g) \\ \partial_\lambda \operatorname{prox}_{g(\cdot,\lambda)} \in \mathbb{R}^{p \times r} \ . \end{aligned}$$

Explicit partial derivatives formulas for usual proximal operators can be found in Table 6.3.

6.2 Bilevel optimization with smooth inner problems

The main challenge to evaluate the hypergradient $\nabla_\lambda \mathcal{L}(\lambda)$ is the computation of the Jacobian $\mathcal{J}_{(\lambda)}$. We first focus on the case where $\Phi(\cdot, \lambda)$ is convex and smooth for any λ.

Table 6.3 – Partial derivatives of proximal operators used.

$g_j(\beta_j, \lambda)$	$\text{prox}_{g_j(\cdot,\lambda)}(z_j)$	$\partial_z \text{prox}_{g_j(\cdot,\lambda)}(z_j)$	$\partial_\lambda \text{prox}_{g_j(\cdot,\lambda)}(z_j)$
$e^\lambda \beta_j^2/2$	$z_j/(1+e^\lambda)$	$1/(1+e^\lambda)$	$-z_j e^\lambda/(1+e^\lambda)^2$
$e^\lambda \lvert\beta_j\rvert$	$\text{ST}(z_j, e^\lambda)$	$\lvert\text{sign}(\text{ST}(z_j, e^\lambda))\rvert$	$-e^\lambda \text{sign}(\text{ST}(z_j, e^\lambda))$
$e^{\lambda_1}\lvert\beta_j\rvert + \frac{1}{2}e^{\lambda_2}\beta_j^2$	$\frac{\text{ST}(z_j, e^{\lambda_1})}{1+e^{\lambda_2}}$	$\frac{\lvert\text{sign}(\text{ST}(z_j, e^{\lambda_1}))\rvert}{1+e^{\lambda_2}}$	$\left(\frac{-e^{\lambda_1}\text{sign}(\text{ST}(z_j, e^{\lambda_1}))}{1+e^{\lambda_2}}, \frac{-\text{ST}(z_j, e^{\lambda_1})e^{\lambda_2}}{(1+e^{\lambda_2})^2}\right)$
$\iota_{[0,e^\lambda]}(\beta_j)$	$\max(0, \min(z_j, e^\lambda))$	$\mathbb{1}_{]0,e^\lambda[}(z_j)$	$e^\lambda \mathbb{1}_{z_j > e^\lambda}$

Implicit differentiation. We recall how the *implicit differentiation*[3] formula of the gradient $\nabla_\lambda \mathcal{L}(\lambda)$ is obtained for smooth inner optimization problems. We will provide a generalization to nonsmooth optimization problems in Section 6.3.2.

Theorem 6.1 (Bengio 2000)**.** *Let $\hat{\beta}^{(\lambda)} \in \arg\min_{\beta \in \mathbb{R}^p} \Phi(\beta, \lambda)$ be a solution of Problem (6.1). Assume that for all $\lambda > 0$, $\Phi(\cdot, \lambda)$ is a convex smooth function, $\nabla^2_\beta \Phi(\hat{\beta}^{(\lambda)}, \lambda) \succ 0$, and that for all $\beta \in \mathbb{R}^p$, $\Phi(\beta, \cdot)$ is differentiable over $]0, +\infty[$. Then the hypergradient $\nabla_\lambda \mathcal{L}(\lambda)$ reads:*

$$\underbrace{\nabla_\lambda \mathcal{L}(\lambda)}_{\in \mathbb{R}^r} = \underbrace{-\nabla^2_{\beta,\lambda} \Phi(\hat{\beta}^{(\lambda)}, \lambda)}_{\in \mathbb{R}^{r \times p}} \underbrace{\left(\nabla^2_\beta \Phi(\hat{\beta}^{(\lambda)}, \lambda)\right)^{-1}}_{\in \mathbb{R}^{p \times p}} \underbrace{\nabla \mathcal{C}(\hat{\beta}^{(\lambda)})}_{\in \mathbb{R}^p} \ . \tag{6.5}$$

Proof For a smooth convex function $\beta \mapsto \Phi(\beta, \lambda)$ the first-order condition writes:

$$\nabla_\beta \Phi(\hat{\beta}^{(\lambda)}, \lambda) = 0 \ , \tag{6.6}$$

for any $\hat{\beta}^{(\lambda)}$ solution of the inner problem. Moreover, if $\lambda \mapsto \nabla_\beta \Phi(\hat{\beta}^{(\lambda)}, \lambda)$ is differentiable, differentiating Equation (6.6) with respect to λ leads to:

$$\nabla^2_{\beta,\lambda} \Phi(\hat{\beta}^{(\lambda)}, \lambda) + \hat{\mathcal{J}}^\top_{(\lambda)} \nabla^2_\beta \Phi(\hat{\beta}^{(\lambda)}, \lambda) = 0 \ . \tag{6.7}$$

The Jacobian $\hat{\mathcal{J}}^\top_{(\lambda)}$ is computed by solving the following linear system:

$$\hat{\mathcal{J}}^\top_{(\lambda)} = -\underbrace{\nabla^2_{\beta,\lambda} \Phi(\hat{\beta}^{(\lambda)}, \lambda)}_{\in \mathbb{R}^{r \times p}} \underbrace{\left(\nabla^2_\beta \Phi(\hat{\beta}^{(\lambda)}, \lambda)\right)^{-1}}_{\in \mathbb{R}^{p \times p}} \ . \tag{6.8}$$

Plugging Equation (6.8) into Equation (6.3) yields the desired result. ■

The computation of the gradient via implicit differentiation (Equation (6.5)) involves the resolution of a $p \times p$ linear system (Bengio, 2000, Sec. 4). This potentially large linear system can be solved using different algorithms such as conjugate gradient (Hestenes and Stiefel 1952, as in Pedregosa 2016) or fixed point methods (Lions and Mercier 1979; Tseng and Yun 2009a, as in Grazzi et al. 2020). Implicit differentiation has been used for model selection of multiple estimators with smooth regularization term: kernel-based models (Chapelle et al., 2002; Seeger, 2008), weighted Ridge estimator (Foo et al., 2008), neural networks (Lorraine et al., 2019) or meta-learning (Franceschi et al., 2018;

[3] Note that *implicit* refers to the implicit function theorem, but leads to an *explicit* formula for the gradient.

Rajeswaran et al., 2019). In addition to hyperparameter selection, it has been applied successfully in natural language processing (Bai et al., 2019) and computer vision (Bai et al., 2020).

Problem (6.1) is typically solved using iterative solvers. In practice, the number of iterations is limited to reduce computation time, and also since very precise solutions are generally not necessary for machine learning tasks. Thus, Equation (6.6) is not exactly satisfied at machine precision, and consequently the linear system to solve Equation (6.5) does not lead to the exact gradient $\nabla_\lambda \mathcal{L}(\lambda)$, see Ablin et al. (2020) for quantitative convergence results. However, Pedregosa (2016) showed that one can resort to *approximate gradients* when the inner problem is smooth, justifying that implicit differentiation can be applied using an approximation of $\hat{\beta}$. Interestingly, this approximation scheme was shown to yield significant practical speedups when solving Problem (6.2), while preserving theoretical properties of convergence toward the optimum.

Iterative differentiation. Iterative differentiation computes the gradient $\nabla_\lambda \mathcal{L}(\lambda)$ by differentiating through the iterates of the algorithm used to solve Problem (6.1). Iterative differentiation can be applied using the forward-mode (Wengert 1964; Deledalle et al. 2014; Franceschi et al. 2017) or the reverse-mode (Linnainmaa 1970; LeCun et al. 1998; Domke 2012). Both rely on the chain rule, the gradient being decomposed as a large product of matrices, computed either in a forward or backward way. Note that forward and reverse modes are algorithm-dependent: in this section we illustrate iterative differentiation for proximal gradient descent (PGD, Lions and Mercier 1979; Combettes and Wajs 2005), using the forward-mode (Algorithm 6.1), and the reverse-mode (Algorithm 6.2).

The most popular method in automatic differentiation is the reverse-mode, a cornerstone of deep learning (Goodfellow et al., 2016, Chap. 8). Iterative differentiation for hyperparameter optimization can be traced back to Domke (2012), who derived (for smooth loss functions) a reverse-mode with gradient descent, heavy ball and L-BFGS algorithms. It first computes the solution of the optimization Problem (6.1) using an iterative solver, but requires storing the iterates along the computation for a backward evaluation of the hypergradient (Algorithm 6.2). Alternatively, the forward-mode computes jointly the solution along with the gradient $\nabla_\lambda \mathcal{L}(\lambda)$. It is memory efficient (no iterates storage) but more computationally expensive when the number of hyperparameters (r) is large; see Baydin et al. (2018) for a survey.

Resolution of the bilevel Problem (6.2). From a theoretical point of view, solving Problem (6.2) using gradient-based methods is also challenging, and results in the literature are quite scarce. Kunisch and Pock (2013) studied the convergence of a semi-Newton algorithm where both the outer and inner problems are smooth. Franceschi et al. (2018) gave similar results with weaker assumptions to unify hyperparameter optimization and meta-learning with a bilevel point of view. They required the inner problem to have a unique solution for all $\lambda > 0$ but do not have second-order assumptions on Φ. Recent results (Ghadimi and Wang, 2018; Ji et al., 2020; Mehmood and Ochs, 2021) have provided quantitative convergence toward a global solution of Problem (6.2), but under global joint convexity assumption and exact knowledge of the gradient Lipschitz constant.

Algorithm 6.1 FORWARD-MODE PGD

input : $\lambda \in \mathbb{R}^r, \gamma > 0, n_{\text{iter}} \in \mathbb{N}, \beta^{(0)} \in \mathbb{R}^p, \mathcal{J}^{(0)} \in \mathbb{R}^{p \times r}$

// jointly compute coef. & Jacobian

for $k = 1, \dots, n_{\text{iter}}$ **do**

  // update the regression coefficients

  $z^{(k)} = \beta^{(k-1)} - \gamma \nabla f(\beta^{(k-1)})$ // GD step

  $\mathrm{d}z^{(k)} = \mathcal{J}^{(k-1)} - \gamma \nabla^2 f(\beta^{(k-1)}) \mathcal{J}^{(k-1)}$

  $\beta^{(k)} = \operatorname{prox}_{\gamma g(\cdot, \lambda)}(z^{(k)})$ // prox. step

  // update the Jacobian

  $\mathcal{J}^{(k)} = \partial_z \operatorname{prox}_{\gamma g(\cdot, \lambda)}(z^{(k)}) \odot \mathrm{d}z^{(k)}$

  $\mathcal{J}^{(k)} \mathrel{+}= \partial_\lambda \operatorname{prox}_{\gamma g(\cdot, \lambda)}(z^{(k)})$ // $\mathcal{O}(pr)$

$v = \nabla \mathcal{C}(\beta^{n_{\text{iter}}})$

return $\beta^{n_{\text{iter}}}, \mathcal{J}^{n_{\text{iter}}\top} v$

Algorithm 6.2 REVERSE-MODE PGD

input : $\lambda \in \mathbb{R}^r, \gamma > 0, n_{\text{iter}} \in \mathbb{N}, \beta^{(0)} \in \mathbb{R}^p$

// computation of $\hat{\beta}$

for $k = 1, \dots, n_{\text{iter}}$ **do**

  $z^{(k)} = \beta^{(k-1)} - \gamma \nabla f(\beta^{(k-1)})$ // GD step

  $\beta^{(k)} = \operatorname{prox}_{\gamma g(\cdot, \lambda)}\left(z^{(k)}\right)$ // prox. step

// backward computation of the gradient g

$v = \nabla \mathcal{C}(\beta^{(n_{\text{iter}})})$, $h = 0_{\mathbb{R}^r}$

for $k = n_{\text{iter}}, n_{\text{iter}} - 1, \dots, 1$ **do**

  $h \mathrel{+}= v^\top \partial_\lambda \operatorname{prox}_{\gamma g(\cdot, \lambda)}(z^{(k)})$ // $\mathcal{O}(pr)$

  $v \leftarrow \partial_z \operatorname{prox}_{\gamma g(\cdot, \lambda)}(z^{(k)}) \odot v$ // $\mathcal{O}(p)$

  $v \leftarrow (\mathrm{Id} - \gamma \nabla^2 f(\beta^{(k)}))v$ // $\mathcal{O}(np)$

return $\beta^{n_{\text{iter}}}, h$

6.3 Bilevel optimization with nonsmooth inner problems

We recalled above how to compute hypergradients when the inner optimization problem is smooth. In this section we tackle the bilevel optimization Problem (6.2) with nonsmooth inner optimization Problem (6.1). Handling nonsmooth inner problems requires specific tools detailed in Section 6.3.1. We then show how to compute gradients with nonsmooth inner problems using implicit differentiation (Section 6.3.2) or iterative differentiation (Section 6.3.3). In Section 6.3.4 we tackle the problem of approximate gradient for a nonsmooth inner optimization problem. Finally, we propose in Section 6.3.6 an algorithm to solve the bilevel optimization Problem (6.2).

6.3.1 Theoretical framework

Differentiability of the regularization path. Before applying first-order methods to tackle Problem (6.2), one must ensure that the regularization path $\lambda \mapsto \hat{\beta}^{(\lambda)}$ is almost everywhere differentiable (as in Figure 6.1). This is the case for the Lasso (Mairal and Yu, 2012) and the SVM (Hastie et al., 2004; Rosset and Zhu, 2007) since solution paths are piecewise differentiable (see Figure 6.1). Results for nonquadratic data fitting terms are scarcer: Friedman et al. (2010) address the practical resolution of sparse logistic regression, but stay evasive regarding the differentiability of the regularization path. In the general case for problems of the form Problem (6.1), we believe it is an open question and leave it for future work.

Differentiability of proximal operators. The key point to obtain an implicit differentiation formula for nonsmooth inner problems is to differentiate the fixed point equation of proximal gradient descent. From a theoretical point of view, ensuring this differentiability at the optimum is non-trivial: Poliquin and Rockafellar (1996b, Thm. 3.8) showed that under a *twice epi-differentiability* condition the proximal operator is differentiable at optimum. For the convergence of forward and reverse modes in the nonsmooth case, one has to ensure that, after enough iterations, the updates of the algorithms become differentiable. Deledalle et al. (2014) justified (weak) differentiability of proximal operators as they are non-expansive. However this may not be a sufficient condition, see Bolte and Pauwels (2020a,b). In our case, we show differentiability after

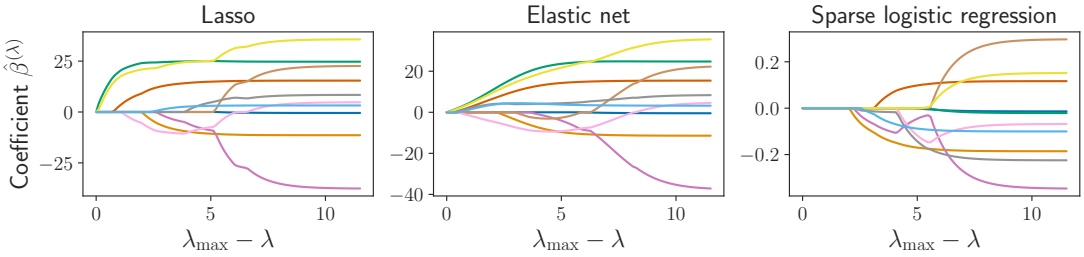


Figure 6.1 – **Regularization paths** (coefficient values as a function of λ), on the *diabetes* and *breast cancer* datasets for the Lasso, the elastic net and sparse logistic regression. This illustrates the weak differentiability of the paths. We used *diabetes* for the Lasso and the elastic net, and the 10 first features of *breast cancer* for the sparse logistic regression.

support identification of the algorithms: active constraints are identified after a finite number of iterations by proximal gradient descent (Liang et al., 2014; Vaiter et al., 2018) and proximal coordinate descent, see Nutini (2018, Sec. 6.2) or Klopfenstein et al. (2020). Once these constraints have been identified convergence is linear towards the Jacobian (see Theorem 6.12 and Figures 6.2 to 6.4).

For the rest of this chapter, we consider the bilevel optimization Problem (6.2) with the following assumptions on the inner Problem (6.1).

Assumption 6.2 (Smoothness)**.** *The function $f : \mathbb{R}^p \to \mathbb{R}$ is a convex, differentiable function, with a L-Lipschitz gradient.*

Assumption 6.3 (Proper, closed, convex)**.** *For all $\lambda \in \mathbb{R}^r$, for any $j \in [p]$, the function $g_j(\cdot, \lambda) : \mathbb{R} \to \mathbb{R}$ is proper, closed and convex.*

Assumption 6.4 (Non-degeneracy)**.** *The problem admits at least one solution:*

$$\arg\min_{\beta \in \mathbb{R}^p} \Phi(\beta, \lambda) \neq \emptyset \ ,$$

and, for any $\hat{\beta}$ solution of Problem (6.1), we have

$$-\nabla f(\hat{\beta}) \in \operatorname{ri}\left(\partial_\beta g(\hat{\beta}, \lambda)\right) \ .$$

To be able to extend iterative and implicit differentiation to the nonsmooth case, we need to introduce the notion of generalized support.

Definition 6.5 (Generalized support, Nutini et al. 2019, Def. 1)**.** *For a solution $\hat{\beta} \in \arg\min_{\beta \in \mathbb{R}^p} \Phi(\beta, \lambda)$, its* generalized support *$\hat{S} \subseteq [p]$ is the set of indices $j \in [p]$ such that g_j is differentiable at $\hat{\beta}_j$:*

$$\hat{S} \triangleq \{j \in [p] : \partial_\beta g_j(\hat{\beta}_j, \lambda) \text{ is a singleton}\} \ .$$

An iterative algorithm is said to achieve ***finite support identification*** *if its iterates $\beta^{(k)}$ converge to $\hat{\beta}$, and there exists $K \geq 0$ such that for all $j \notin \hat{S}$, for all $k \geq K, \beta_j^{(k)} = \hat{\beta}_j$.*

Examples. For the ℓ_1 norm (promoting sparsity), $g_j(\hat{\beta}_j, \lambda) = e^\lambda |\hat{\beta}_j|$, the generalized support is $\hat{S} \triangleq \{j \in [p] : \hat{\beta}_j \neq 0\}$. This set corresponds to the indices of the non-zero coefficients, which is the usual support definition. For the SVM estimator, $g_j(\hat{\beta}_j, \lambda) = \iota_{[0,e^\lambda]}(\hat{\beta}_j)$. This function is non-differentiable at 0 and at e^λ. The generalized support for the SVM estimator then corresponds to the set of indices such that $\hat{\beta}_j \in]0, e^\lambda[$.

Finally, to prove local linear convergence of the Jacobian we assume regularity and strong convexity on the generalized support.

Assumption 6.6 (Locally $\mathcal{C}^2$ and $\mathcal{C}^3$)**.** *The map $\beta \mapsto f(\beta)$ is locally $\mathcal{C}^3$ around $\hat{\beta}$. For all $\lambda \in \mathbb{R}^r$, for all $j \in \hat{S}$ the map $g_j(\cdot, \lambda)$ is locally $\mathcal{C}^2$ around $\hat{\beta}_j$.*

Assumption 6.7 (Restricted injectivity)**.** *Let $\hat{\beta}$ be a solution of Problem (6.1) and $\hat{S}$ its generalized support. The solution $\hat{\beta}$ satisfies the following restricted injectivity condition:*

$$\nabla^2_{\hat{S},\hat{S}} f(\hat{\beta}) \succ 0 \ .$$

Assumptions 6.2 and 6.3 are classical to ensure inner problems can be solved using proximal algorithms. Assumption 6.4 can be seen as a generalization of constraint qualifications (Hare and Lewis, 2007, Sec. 1) and is crucial to ensure *support identification*. Assumptions 6.6 and 6.7 are classical for the analysis (Liang et al., 2017) and sufficient to derive rates of convergence for the Jacobian of the inner problem once the generalized support has been identified.

The next lemma guarantees uniqueness of Problem (6.1) under Assumptions 6.4 and 6.7.

Lemma 6.8 (Liang et al. 2017, Prop. 4.1)**.** *Assume that there exists a neighborhood Λ of λ such that Assumptions 6.4 and 6.7 are satisfied for every $\lambda \in \Lambda$. Then for every $\lambda \in \Lambda$, Problem (6.1) has a unique solution, and the map $\lambda \mapsto \hat{\beta}^{(\lambda)}$ is well-defined on Λ.*

We first show how implicit and iterative differentiation can be used with a nonsmooth inner problem. Peyré and Fadili (2011) proposed to smooth the inner optimization problem, Ochs et al. (2015); Frecon et al. (2018) relied on the forward-mode combined with Bregman iterations to get differentiable steps. For nonsmooth optimization problems, implicit differentiation has been considered for (constrained) convex optimization problems (Gould et al., 2016; Amos and Kolter, 2017; Agrawal et al., 2019), Lasso-type problems (Mairal et al., 2012; **Bertrand** et al., 2020), total variation penalties (Cherkaoui et al., 2020), dictionary learning (Malézieux et al., 2021), and generalized to strongly monotone operators (Winston and Kolter, 2020).

6.3.2 Hypergradient computation: implicit differentiation

The exact proof of Theorem 6.1 cannot be applied when $\beta \mapsto \Phi(\beta, \lambda)$ is nonsmooth, as Equations (6.6) and (6.7) no longer hold. Nevertheless, instead of the optimality condition of smooth optimization, Equation (6.6), one can leverage the fixed point iteration of proximal gradient descent, which we will see in Equation (6.11). The main theoretical challenge is to show the differentiability of the function $\beta \mapsto \text{prox}_{\gamma g}(\beta - \gamma \nabla f(\beta))$. Besides, taking advantage of the generalized sparsity of the regression coefficients $\hat{\beta}^{(\lambda)}$, one can show that the Jacobian $\hat{\mathcal{J}}$ is row-sparse, leading to substantial computational benefits when computing the hypergradient $\nabla_\lambda \mathcal{L}(\lambda))$ for Problem (6.1),

Theorem 6.9 (nonsmooth implicit formula)**.** *Suppose Assumptions 6.2, 6.3 and 6.6 hold. Let $0 < \gamma \leq 1/L$, where L is the Lipschitz constant of ∇f. Let $\lambda \in \mathbb{R}^r$, Λ be a neighborhood of λ, and $\Gamma^\Lambda \triangleq \left\{\hat{\beta}^{(\lambda)} - \gamma \nabla f(\hat{\beta}^{(\lambda)}) \,:\, \lambda \in \Lambda\right\}$. In addition,*

(H1) Suppose Assumptions 6.4 and 6.7 hold on Λ.

(H2) Suppose $\lambda \mapsto \hat{\beta}^{(\lambda)}$ is continuously differentiable on Λ.

(H3) Suppose for all $z \in \Gamma^\Lambda$, $\lambda \mapsto \operatorname{prox}_{\gamma g(\cdot,\lambda)}(z)$ is continuously differentiable on Λ.

(H4) Suppose $\partial_z \operatorname{prox}_{\gamma g(\cdot,\lambda)}$ and $\partial_\lambda \operatorname{prox}_{\gamma g(\cdot,\lambda)}$ are Lipschitz continuous on $\Gamma^\Lambda \times \Lambda$.

Let $\hat{\beta} \triangleq \hat{\beta}^{(\lambda)}$ be the solution of Problem (6.1), $\hat{S}$ its generalized support of cardinality $\hat{s}$. Then the Jacobian $\hat{\mathcal{J}}$ of the inner Problem (6.1) is given by the following formula,

$\hat{z} = \hat{\beta} - \gamma \nabla f(\hat{\beta})$, and $A \triangleq \mathrm{Id}_{\hat{s}} - \partial_z \operatorname{prox}_{\gamma g(\cdot,\lambda)}(\hat{z})_{\hat{S}} \odot \left(\mathrm{Id}_{\hat{s}} - \gamma \nabla^2_{\hat{S},\hat{S}} f(\hat{\beta})\right)$:

$$\hat{\mathcal{J}}_{\hat{S}^c:} = \partial_\lambda \operatorname{prox}_{\gamma g(\cdot,\lambda)}\left(\hat{z}\right)_{\hat{S}^c} \quad , \tag{6.9}$$

$$\hat{\mathcal{J}}_{\hat{S}:} = A^{-1}\left(\partial_\lambda \operatorname{prox}_{\gamma g(\cdot,\lambda)}(\hat{z})_{\hat{S}} - \gamma \partial_z \operatorname{prox}_{\gamma g(\cdot,\lambda)}(\hat{z})_{\hat{S}} \odot \nabla^2_{\hat{S},\hat{S}^c} f(\hat{\beta}) \hat{\mathcal{J}}_{\hat{S}^c}\right) \quad . \tag{6.10}$$

Proof According to Lemma 6.8, Assumptions 6.4 and 6.7 ensure Problem (6.1) has a unique minimizer and $\lambda \mapsto \hat{\beta}^{(\lambda)}$ is well-defined on Λ. We consider the proximal gradient descent fixed point equation:

$$\hat{\beta}^{(\lambda)} = \operatorname{prox}_{\gamma g_{(\cdot,\lambda)}}\left(\hat{\beta}^{(\lambda)} - \gamma \nabla f(\hat{\beta}^{(\lambda)})\right) \quad . \tag{6.11}$$

Together with the conclusion of Lemma 6.8, Assumptions 6.2 and 6.6, and given (H2), (H3) and (H4), we have that $\lambda \mapsto \psi\left(\beta^{(\lambda)} - \gamma \nabla f(\hat{\beta}^{(\lambda)}), \lambda\right) \triangleq \operatorname{prox}_{\gamma g(\cdot,\lambda)}\left(\hat{\beta}^{(\lambda)} - \gamma \nabla f(\hat{\beta}^{(\lambda)})\right)$ is differentiable at λ. One can thus differentiate Equation (6.11) with respect to λ, which leads to:

$$\hat{\mathcal{J}} = \partial_z \operatorname{prox}_{\gamma g(\cdot,\lambda)}(\hat{z}) \odot \left(\mathrm{Id} - \gamma \nabla^2 f(\hat{\beta})\right) \hat{\mathcal{J}} + \partial_\lambda \operatorname{prox}_{\gamma g(\cdot,\lambda)}\left(\hat{z}\right) \quad , \tag{6.12}$$

with $\hat{z} = \hat{\beta} - \gamma \nabla f(\hat{\beta})$. In addition to $0 < \gamma < 1/L \leq 1/L_j$, the separability of g and Assumptions 6.2 to 6.4 and 6.6 ensure (see Lemma 2.19) that for any $j \in \hat{S}^c$,

$$\partial_z \operatorname{prox}_{\gamma g_j(\cdot,\lambda)}\left(\hat{\beta}_j - \gamma \nabla_j f(\hat{\beta})\right) = 0 \quad . \tag{6.13}$$

Plugging Equation (6.13) into Equation (6.12) ensures Equation (6.9) for all $j \in \hat{S}^c$:

$$\hat{\mathcal{J}}_{j:} = \partial_\lambda \operatorname{prox}_{\gamma g_j(\cdot,\lambda)}\left(\hat{\beta}_j - \gamma \nabla_j f(\hat{\beta})\right) \quad . \tag{6.14}$$

Plugging Equations (6.13) and (6.14) into Equation (6.12) shows that the Jacobian restricted on the generalized support $\hat{S}$ satisfies the following linear system:

$$\left(\mathrm{Id}_{\hat{s}} - \partial_z \operatorname{prox}_{\gamma g(\cdot,\lambda)}\left(\hat{z}\right)_{\hat{S}} \odot (\mathrm{Id}_{\hat{s}} - \gamma \nabla^2_{\hat{S},\hat{S}} f(\hat{\beta}))\right) \hat{\mathcal{J}}_{\hat{S}:} = -\gamma \partial_z \operatorname{prox}_{\gamma g(\cdot,\lambda)}(\hat{z})_{\hat{S}} \odot \nabla^2_{\hat{S},\hat{S}^c} f(\hat{\beta}) \hat{\mathcal{J}}_{\hat{S}^c:} + \partial_\lambda \operatorname{prox}_g(\hat{z})_{\hat{S}:} \quad .$$

Since $0 < \gamma \leq 1/L$,

$$\|\partial_z \operatorname{prox}_{\gamma g(\cdot,\lambda)}(\hat{z})_{\hat{S}} \odot (\mathrm{Id}_{\hat{s}} - \gamma \nabla^2_{\hat{S},\hat{S}} f(\hat{\beta}))\|_2 \leq \|\partial_z \operatorname{prox}_{\gamma g(\cdot,\lambda)}(\hat{z})_{\hat{S}}\| \cdot \|\mathrm{Id}_{\hat{s}} - \gamma \nabla^2_{\hat{S},\hat{S}} f(\hat{\beta})\|_2 < 1 \ . \tag{6.15}$$

Since Equation (6.15) holds, $A \triangleq \mathrm{Id}_{\hat{s}} - \partial_z \operatorname{prox}_{\gamma g(\cdot,\lambda)}(\hat{z})_{\hat{S}} \odot (\mathrm{Id}_{\hat{s}} - \gamma \nabla^2_{\hat{S},\hat{S}} f(\hat{\beta}))$ is invertible, which leads to Equation (6.10). ■

Remark 6.10. *In the smooth case a $p \times p$ linear system is needed to compute the Jacobian in Equation (6.8). For nonsmooth problems this is reduced to an $\hat{s} \times \hat{s}$ linear system ($\hat{s} \leq p$ being the size of the generalized support,* e.g., *the number of non-zero coefficients for the Lasso). This leads to significant speedups in practice, especially for very sparse vector $\hat{\beta}^{(\lambda)}$.*

Remark 6.11. *To obtain Theorem 6.9 we differentiated the fixed point equation of proximal gradient descent, though one could differentiate other fixed point equations (such as the one from proximal coordinate descent). The value of the Jacobian $\hat{\mathcal{J}}$ obtained with different fixed point equations would be the same, yet the associated systems could have different numerical stability properties. We leave this analysis to future work.*

6.3.3 Hypergradient computation: iterative differentiation

Instead of implicit differentiation, it is also possible to use iterative differentiation on proximal solvers. In section Section 6.2 we presented forward and reverse modes differentiation of proximal gradient descent (Algorithms 6.1 and 6.2). In this section we study the iterative differentiation of proximal coordinate descent (Algorithms 6.3 and 6.4). To instantiate algorithms easily on problems such as the Lasso, partial derivatives of usual proximal operators can be found in Table 6.3.

For coordinate descent, the computation of the iterative Jacobian in a forward way involves differentiating the following update:

$$\begin{aligned} z_j &\leftarrow \beta_j - \gamma_j \nabla_j f(\beta) \\ \beta_j &\leftarrow \operatorname{prox}_{\gamma_j g_j}\Big(\beta_j - \gamma_j \nabla_j f(\beta)\Big) \\ \mathcal{J}_{j:} &\leftarrow \underbrace{\partial_z \operatorname{prox}_{\gamma_j g_j(\cdot,\lambda)}(z_j)}_{\in \mathbb{R}} \underbrace{\Big(\mathcal{J}_{j:} - \gamma_j \nabla^2_{j:} f(\beta)\, \mathcal{J}\Big)}_{\in \mathbb{R}^p} + \underbrace{\partial_\lambda \operatorname{prox}_{\gamma_j g_j(\cdot,\lambda)}(z_j)}_{\in \mathbb{R}^p} \ . \end{aligned}$$

We address now the convergence of the iterative Jacobian scheme, a question which remained open in Deledalle et al. (2014, Section 4.1). We show next that the forward-mode converges to the Jacobian in the nonsmooth separable setting of this chapter. Moreover, we prove that the iterative Jacobian convergence is locally linear after support identification.

Theorem 6.12 (Local linear convergence of the Jacobian)**.** *Let $0 < \gamma \leq 1/L$. Suppose Assumptions 6.2, 6.3 and 6.6 hold. Let $\lambda \in \mathbb{R}^r$, Λ be a neighborhood of λ, and $\Gamma^\Lambda \triangleq \left\{\hat{\beta}^{(\lambda)} - \gamma \nabla f(\hat{\beta}^{(\lambda)}) \ : \ \lambda \in \Lambda\right\}$. In addition, suppose hypotheses (H1) to (H4) from Theorem 6.9 are satisfied and the sequence $(\beta^{(k)})_{k\in\mathbb{N}}$ generated by Algorithm 6.1 (respectively by Algorithm 6.3) converges toward $\hat{\beta}$.*

Algorithm 6.3 FORWARD-MODE PCD

input : $X \in \mathbb{R}^{n \times p}, y \in \mathbb{R}^n, \lambda \in \mathbb{R}^r,$
$n_{\text{iter}} \in \mathbb{N}, \beta \in \mathbb{R}^p$
$\mathcal{J} \in \mathbb{R}^{p \times r}, \gamma_1, \dots, \gamma_p$

// jointly compute coef. & Jacobian
for $k = 1, \dots, n_{\text{iter}}$ **do**
  for $j = 1, \dots, p$ **do**
    // update the regression coefficients
    $z_j \leftarrow \beta_j - \gamma_j \nabla_j f(\beta)$ // CD step
    $\mathrm{d}z_j \leftarrow \mathcal{J}_{j:} - \gamma_j \nabla^2_{j:} f(\beta)\, \mathcal{J}$
    $\beta_j \leftarrow \operatorname{prox}_{\gamma_j g_j(\cdot, \lambda)}(z_j)$ // prox. step
    // update the Jacobian
    // diff. with respect to λ
    $\mathcal{J}_{j:} \leftarrow \partial_z \operatorname{prox}_{\gamma_j g_j(\cdot, \lambda)}(z_j) \mathrm{d}z_j$
    $\mathcal{J}_{j:} \mathrel{+}= \partial_\lambda \operatorname{prox}_{\gamma_j g_j(\cdot, \lambda)}(z_j)$
  $\beta^{(k)} = \beta$
  $\mathcal{J}^{(k)} = \mathcal{J}$
$v = \nabla C(\beta)$
return $\beta^{n_{\text{iter}}}, \mathcal{J}^\top v$

Algorithm 6.4 REVERSE-MODE PCD

input : $X \in \mathbb{R}^{n \times p}, y \in \mathbb{R}^n, \lambda \in \mathbb{R}^r, n_{\text{iter}} \in \mathbb{N}, \beta \in \mathbb{R}^p, \gamma_1, \dots, \gamma_p > 0$

// compute coef.
for $k = 1, \dots, n_{\text{iter}}$ **do**
  for $j = 1, \dots, p$ **do**
    // update the regression coefficients
    $z_j \leftarrow \beta_j - \gamma_j \nabla_j f(\beta)$ // CD step
    $\beta_j \leftarrow \operatorname{prox}_{\gamma_j g_j(\cdot, \lambda)}(z_j)$ // prox. step
    $\beta^{(k,j)} = \beta; z_j^{(k)} = z_j$ // store iterates
// compute gradient g in a backward way
$v = \nabla C(\beta^{n_{\text{iter}}}), h = 0_{\mathbb{R}^r}$
for $k = n_{\text{iter}}, n_{\text{iter}} - 1, \dots, 1$ **do**
  for $j = p, \dots, 1$ **do**
    $h \mathrel{-}= \gamma_j v_j \partial_\lambda \operatorname{prox}_{\gamma_j g_j(\cdot, \lambda)}(z_j^{(k)})$
    $v_j \mathrel{*}= \partial_z \operatorname{prox}_{\gamma_j g_j(\cdot, \lambda)}(z_j^{(k)})$
    $v \mathrel{-}= \gamma_j v_j \nabla^2_{j:} f(\beta^{(k,j)})$ // $\mathcal{O}(np)$
return $\beta^{n_{\text{iter}}}, h$

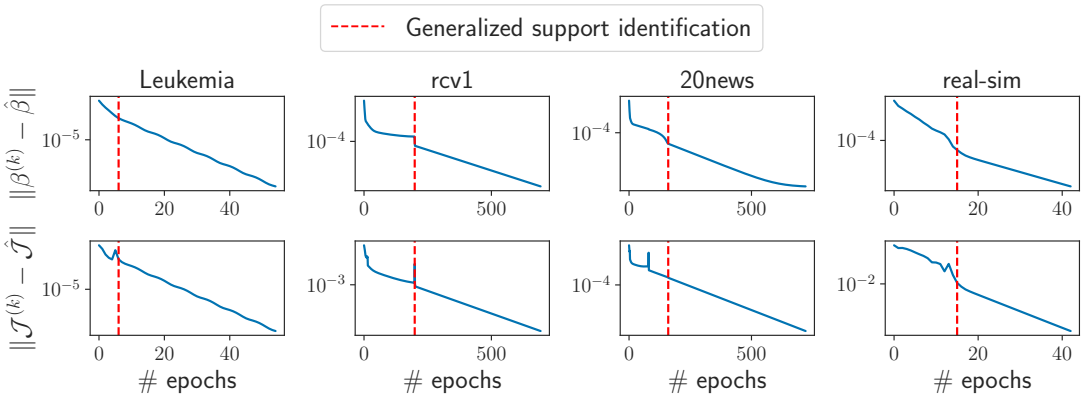


Figure 6.2 – **Local linear convergence of the Jacobian for the SVM.** Distance to optimum for the coefficients β (top) and the Jacobian $\mathcal{J}$ (bottom) of the forward-mode differentiation of proximal coordinate descent (Algorithm 6.3) on multiple datasets. One epoch corresponds to one pass over the data, *i.e.*, one iteration with proximal gradient descent.

Then, the sequence of Jacobians $(\mathcal{J}^{(k)})_{k \geq 0}$ *generated by the forward-mode differentiation of proximal gradient descent (Algorithm 6.1) (respectively by forward-mode differentiation of proximal coordinate descent, Algorithm 6.3) converges locally linearly towards* $\hat{\mathcal{J}}$.

Proof of Theorem 6.12 can be found in Section 6.A.1.

Figures 6.3 and 6.4 are the counterparts of Figure 6.2 for the Lasso and sparse logistic regression. It shows the local linear convergence of the Jacobian for the Lasso, obtained by the forward-mode differentiation of coordinate descent. The solvers used to determine

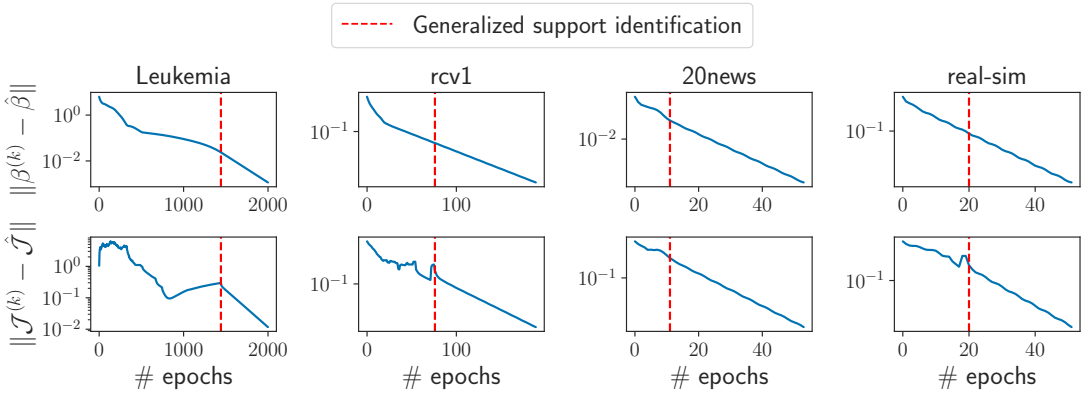


Figure 6.3 – **Local linear convergence of the Jacobian for the Lasso.** Distance to optimum for the coefficients β (top) and the Jacobian $\mathcal{J}$ (bottom) of the forward-mode differentiation of proximal coordinate descent (Algorithm 6.3) on multiple datasets.

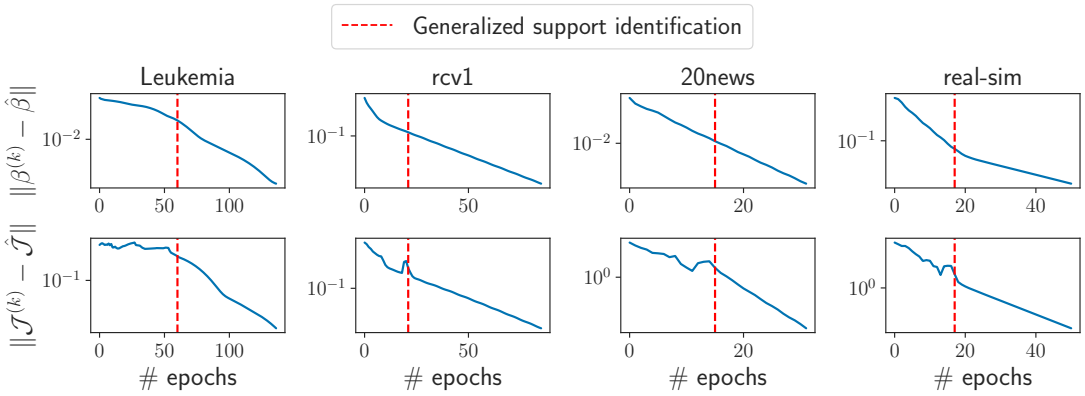


Figure 6.4 – **Local linear convergence of the Jacobian for sparse logistic regression.** Distance to optimum for the coefficients β (top) and the Jacobian $\mathcal{J}$ (bottom) of the forward-mode differentiation of proximal coordinate descent (Algorithm 6.3) on multiple datasets.

the exact solution up to machine precision are `Celer` (Massias et al., 2018b, 2020b) for the Lasso and `Blitz` (Johnson and Guestrin, 2015) for the sparse logistic regression. Table 6.1 summarizes the values of the hyperparameters λ used in Figures 6.2 to 6.4.

Comments on Figure 6.2. We illustrate the results of Theorem 6.12 on SVM (for the Lasso and sparse logistic regression, see Figures 6.3 and 6.4) for multiple datasets (*leukemia*, *rcv1*, *news20* and *real-sim*[4]). The values of the hyperparameters λ are summarized in Table 6.1. Regression coefficients $\hat{\beta}^{(\lambda)}$ were computed to machine precision (up to duality gap smaller than 10^{-16}) using a state-of-the-art coordinate descent solver implemented in `Lightning` (Blondel and Pedregosa, 2016). The exact Jacobian was computed via implicit differentiation (Equation (6.10)). Once these quantities were obtained, we used the forward-mode differentiation of proximal coordinate descent (Algorithm 6.3) and monitored the distance between the iterates of the regression coefficients $\beta^{(k)}$ and the exact solution $\hat{\beta}$. We also monitored the distance between the

[4]Data available on the `libsvm` website: https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

Table 6.1 – Dataset characteristics and regularization parameters used in Figures 6.2 to 6.4.

Datasets # samples # features	*leukemia* $n = 38$ $p = 7129$	*rcv1* $n = 20\,242$ $p = 19\,959$	*news20* $n = 19\,996$ $p = 632\,982$	*real-sim* $n = 72\,309$ $p = 20\,958$
Lasso	$e^\lambda = 0.01\, e^{\lambda_{\max}}$	$e^\lambda = 0.075\, e^{\lambda_{\max}}$	$e^\lambda = 0.3\, e^{\lambda_{\max}}$	$e^\lambda = 0.1\, e^{\lambda_{\max}}$
Logistic regression	$e^\lambda = 0.1\, e^{\lambda_{\max}}$	$e^\lambda = 0.25\, e^{\lambda_{\max}}$	$e^\lambda = 0.8\, e^{\lambda_{\max}}$	$e^\lambda = 0.15\, e^{\lambda_{\max}}$
SVM	$e^\lambda = 10^{-5}$	$e^\lambda = 3 \times 10^{-2}$	$e^\lambda = 10^{-3}$	$e^\lambda = 5 \times 10^{-2}$

iterates of the Jacobian $\mathcal{J}^{(k)}$ and the exact Jacobian $\hat{\mathcal{J}}$. The red vertical dashed line represents the iteration number where support identification happens. Once the support is identified, Figures 6.2 to 6.4 illustrate the linear convergence of the Jacobian. However, the behavior of the iterative Jacobian before support identification is more erratic and not even monotone.

6.3.4 Hypergradient computation with approximate gradients

As mentioned in Section 6.2, relying on iterative algorithms to solve Problem (6.1), one only has access to an approximation of $\hat{\beta}^{(\lambda)}$: this may lead to numerical errors when computing the gradient in Theorem 6.9. Extending the result of Pedregosa (2016, Thm. 1), which states that hypergradients can be computed approximately, we give a stability result for the computation of approximate hypergradients in the case of nonsmooth inner problems. For this purpose we need to add several assumptions to the previous framework.

Theorem 6.13 (Bound on the error of approximate hypergradient)**.** *For $\lambda \in \mathbb{R}^r$, let $\hat{\beta}^{(\lambda)} \in \mathbb{R}^p$ be the exact solution of the inner Problem (6.1), and $\hat{S}$ its generalized support. Suppose Assumptions 6.2, 6.3 and 6.6 hold. Let Λ be a neighborhood of λ, and $\Gamma^\Lambda \triangleq \left\{\hat{\beta}^{(\lambda)} - \gamma \nabla f(\hat{\beta}^{(\lambda)}) \,:\, \lambda \in \Lambda\right\}$. Suppose hypotheses (H1) to (H4) from Theorem 6.9 are satisfied. In addition suppose*

(H5) The application $\beta \mapsto \nabla^2 f(\beta)$ is Lipschitz continuous.

(H6) The criterion $\beta \mapsto \nabla \mathcal{C}(\beta)$ is Lipschitz continuous.

(H7) Both optimization problems in Algorithm 6.5 are solved up to precision ϵ with support identification: $\|\beta^{(\lambda)} - \hat{\beta}^{(\lambda)}\| \leq \epsilon$, $A^\top$ is invertible, and $\|A^{-1\top} \nabla_{\hat{S}} \mathcal{C}(\beta^{(\lambda)}) - v\| \leq \epsilon$.

Then the error on the approximate hypergradient h returned by Algorithm 6.5 is of the order of magnitude of the error ϵ on $\beta^{(\lambda)}$ and v:

$$\|\nabla \mathcal{L}(\lambda) - h\| = \mathcal{O}(\epsilon) \ .$$

Proof of Theorem 6.13 can be found in Section 6.B. Following the analysis of Pedregosa (2016), two sources of approximation errors arise when computing the hypergradient: one from the inexact computation of $\hat{\beta}$, and another from the approximate resolution of the linear system. Theorem 6.13 states that if the inner optimization problem and the linear system are solved up to precision ϵ, *i.e.*, $\|\hat{\beta}^{(\lambda)} - \beta^{(\lambda)}\| \leq \epsilon$ and

$\|A^{-1\top}\nabla_S \mathcal{C}(\beta^{(\lambda)}) - v\| \leq \epsilon$, then the approximation on the hypergradient is also of the order of ϵ.

Remark 6.14. *The Lipschitz continuity of the proximity operator with respect to λ (H4) is satisfied for usual proximal operators, in particular all the operators in Table 6.3. The Lipschitz continuity of the Hessian and the criterion, hypotheses (H5) and (H6), are satisfied for usual machine learning loss functions and criteria, such as the least squares and the logistic loss.*

Remark 6.15. *To simplify the analysis, we used the same tolerance for the resolution of the inner Problem (6.1) and the resolution of the linear system. Theorem 6.13 gives intuition on the fact that the inner problem does not need to be solved at high precision to lead to good hypergradients estimation. Note that in practice one does not easily control the distance between the approximate solution and the exact one $\|\beta^{(k)} - \hat{\beta}\|$: most softwares provide a solution up to a given duality gap (sometimes even other criteria), not $\|\beta^{(k)} - \hat{\beta}\|$.*

6.3.5 Proposed method for hypergradient computation

We now describe our proposed method to compute the hypergradient of Problem (6.2). In order to take advantage of the sparsity induced by the generalized support, we propose an implicit differentiation algorithm for nonsmooth inner problem that can be found in Algorithm 6.5. First, we compute a solution of the inner Problem (6.1) using a solver identifying the generalized support (Liang et al., 2014; Klopfenstein et al., 2020). Then, the hypergradient is computed by solving the linear system in Equation (6.10). This linear system, as mentioned in Section 6.2, can be solved using multiple algorithms, including conjugate gradient or fixed point methods. Table 6.2 summarizes the computational complexity in space and time of the described algorithms.

Table 6.2 – Cost in time and space for each method: p is the number of features, n the number of samples, r the number of hyperparameters, and $\hat{s}$ is the size of the generalized support (Definition 6.5, $\hat{s} \leq p$ and usually $\hat{s} \ll p$). The number of iterations of the inner solver is noted n_{iter}, the number of iterations of the solver of the linear system is noted n_{sys}.

Differentiation	Algorithm	Space	Time
Forward-mode PGD	Algorithm 6.1	$\mathcal{O}(p\,r)$	$\mathcal{O}(n\,p\,r\,n_{\text{iter}})$
Reverse-mode PGD	Algorithm 6.2	$\mathcal{O}(p\,n_{\text{iter}})$	$\mathcal{O}(n\,p\,n_{\text{iter}} + n\,p\,n_{\text{iter}})$
Forward-mode PCD	Algorithm 6.3	$\mathcal{O}(p\,r)$	$\mathcal{O}(n\,p\,r\,n_{\text{iter}})$
Reverse-mode PCD	Algorithm 6.4	$\mathcal{O}(p\,n_{\text{iter}})$	$\mathcal{O}(n\,p\,n_{\text{iter}} + n\,p^2\,n_{\text{iter}})$
Implicit differentiation	Algorithm 6.5	$\mathcal{O}(p + \hat{s})$	$\mathcal{O}(n\,p\,n_{\text{iter}} + n\,\hat{s}\,n_{\text{sys}})$

6.3.6 Resolution of the bilevel optimization Problem (6.2)

From a practical point of view, once the hypergradient has been computed, first-order methods require the definition of a step size to solve the non-convex Problem (6.2). As the Lipschitz constant is not available for the outer problem, first-order methods need to rely on other strategies, such as:

- Gradient descent with manually adjusted fixed step sizes (Frecon et al., 2018; Ji et al., 2020). The main disadvantage of this technique is that it requires a careful tuning of the step size for each experiment. In addition to being potentially tedious, it does not lead to an automatic procedure.
- L-BFGS (as in Deledalle et al. 2014). L-BFGS is a quasi-Newton algorithm that exploits past iterates to approximate the Hessian and propose a better descent direction, which is combined with some line search (Nocedal and Wright, 2006). Yet, due to the approximate gradient computation, we observed that L-BFGS did not always converge.
- ADAM (Kingma and Ba, 2014). It turned out to be inappropriate to the present setting. ADAM was very sensitive to the initial step size and required a careful tuning for each experiment.
- Iteration specific step sizes obtained by line search (Pedregosa, 2016). While the approach from Pedregosa (2016) requires no tuning, we observed that it could diverge when close to the optimum. The adaptive step size strategy proposed in Algorithm 6.6, used in all the experiments, turned out to be robust and efficient across problems and datasets.

Remark 6.16 (Uniqueness)**.** *The solution of Problem (6.1) may be non-unique, leading to a multi-valued regularization path* $\lambda \mapsto \hat{\beta}^{(\lambda)}$ *(Liu et al., 2020) and requiring tools such as* optimistic gradient *(Dempe et al., 2015, Chap. 3.8). Though it is not possible to ensure uniqueness in practice, we did not face experimental issues due to potential non-uniqueness. For the Lasso, this experimental observation can be theoretically justified (Tibshirani, 2013): when the design matrix is sampled from a continuous distribution, the solution of the Lasso is almost surely unique.*

Remark 6.17 (Initialization)**.** *One advantage of the nonsmooth case with the* ℓ_1 *norm is that one can find a good initialization point: there exists a value* $\lambda_{\max}$ *(see Table 6.1) such that the solution of Problem (6.1) vanishes for* $\lambda \geq \lambda_{\max}$*. Hence, a convenient and robust initialization value can be chosen as* $e^{\lambda} = e^{\lambda_{\max}}/100$*. This is in contrast with the smooth case, where finding a good initialization heuristic is hard: starting in flat zones can lead to poor performance for gradient-based methods (Pedregosa, 2016).*

6.4 Experiments

In this section, we illustrate the benefits of our proposed Algorithm 6.5 to compute hypergradients and Algorithm 6.6 to solve Problem (6.2). Our package, `sparse-ho`, is implemented in `Python`. It relies on `Numpy` (Harris et al., 2020), `Numba` (Lam et al., 2015) and `SciPy` (Virtanen et al., 2020). Figures were plotted using `matplotlib` (Hunter, 2007). The package is available under BSD3 license at https://github.com/qb3/sparse-ho, with documentation and examples available at https://qb3.github.io/sparse-ho/. Online code includes scripts to reproduce all figures and experiments of the chapter.

6.4.1 Hypergradient computation

Comparison with alternative approaches (Figure 6.1). First, we compare different methods to compute the hypergradient:

Algorithm 6.5 IMPLICIT DIFFERENTIATION

input : $\lambda \in \mathbb{R}, \epsilon > 0$
init : $\gamma > 0$
// compute the solution of inner problem
Find β such that: $\Phi(\beta, \lambda) - \Phi(\hat{\beta}, \lambda) \leq \epsilon$
// compute the gradient
Compute the generalized support S of β,
$z = \beta - \gamma \nabla f(\beta)$
$\mathcal{J}_{S^c:} = \partial_\lambda \operatorname{prox}_{\gamma g(\cdot,\lambda)}(z)_{S^c}$
$s = |S|$
$A = \mathrm{Id}_s - \partial_z \operatorname{prox}_{\gamma g(\cdot,\lambda)}(z)_S \odot (\mathrm{Id}_s - \gamma \nabla^2_{S,S} f(\beta))$
Find $v \in \mathbb{R}^s$ s.t. $\|A^{-1\top} \nabla_S \mathcal{C}(\beta) - v\| \leq \epsilon$
$B = \partial_\lambda \operatorname{prox}_{\gamma g(\cdot,\lambda)}(z)_S - \gamma \partial_z \operatorname{prox}_{\gamma g(\cdot,\lambda)}(z)_S \odot \nabla^2_{S,S^c} f(\beta)\, \mathcal{J}_{S^c}$
$\nabla \mathcal{L}(\lambda) = \mathcal{J}^\top_{S^c:} \nabla_{S^c} \mathcal{C}(\beta) + v^\top B$
return $\mathcal{L}(\lambda) \triangleq \mathcal{C}(\beta), \nabla \mathcal{L}(\lambda)$

Algorithm 6.6 GRADIENT DESCENT WITH APPROXIMATE GRADIENT

input : $\lambda \in \mathbb{R}^r, (\epsilon_i)$
init : use_adaptive_step_size = True
for $i = 1, \ldots,$ iter **do**
  $\lambda^{\text{old}} \leftarrow \lambda$
  // compute the value and the gradient
  $\mathcal{L}(\lambda), \nabla \mathcal{L}(\lambda) \leftarrow$ Algorithm 6.5$(X, y, \lambda, \epsilon_i)$
  if use_adaptive_step_size **then**
    $\alpha = 1/\|\nabla \mathcal{L}(\lambda)\|$
  $\lambda \mathrel{-}= \alpha \nabla \mathcal{L}(\lambda)$ // gradient step
  if $\mathcal{L}(\lambda) > \mathcal{L}(\lambda^{\text{old}})$ **then**
    use_adaptive_step_size = False
    $\alpha \mathrel{/}= 10$
return λ

Table 6.1 – Characteristics of the datasets used for the experiments.

name	# samples n	# features p	# classes q	density
breast cancer	569	30	–	1
diabetes	442	10	–	1
leukemia	72	7129	–	1
gina agnostic	3468	970	–	1
rcv1	20 242	19 960	–	3.7×10^{-3}
real-sim	72 309	20 958	–	2.4×10^{-3}
news20	19 996	632 983	–	6.1×10^{-4}
mnist	60.000	683	10	2.2×10^{-1}
usps	7291	256	10	1
rcv1 (multiclass)	15 564	16 245	53	4.0×10^{-3}
aloi	108 000	128	1000	2.4×10^{-1}

- Forward-mode differentiation of proximal coordinate descent (Algorithm 6.3).
- Reverse-mode differentiation of proximal coordinate descent (Algorithm 6.4).
- `cvxpylayers` (Agrawal et al., 2019), a software based on `cvxpy` (Diamond and Boyd, 2016), solving *disciplined parametrized programming* and providing derivatives with respect to the parameters of the program. It is thus possible to use `cvxpylayers` to compute gradients with respect to the regularization parameters.

Figure 6.1 compares the time taken by multiple methods to compute a single hypergradient $\nabla \mathcal{L}(\lambda)$ for the Lasso (see Table 6.1), for multiple values of λ. It shows the time taken to compute the regression coefficients and the hypergradient, as a function of the number of columns, sampled from the design matrix from the *gina* dataset. The columns were selected at random and 10 repetitions were performed for each point of the curves. In order to aim for good numerical precision, problems were solved up to a

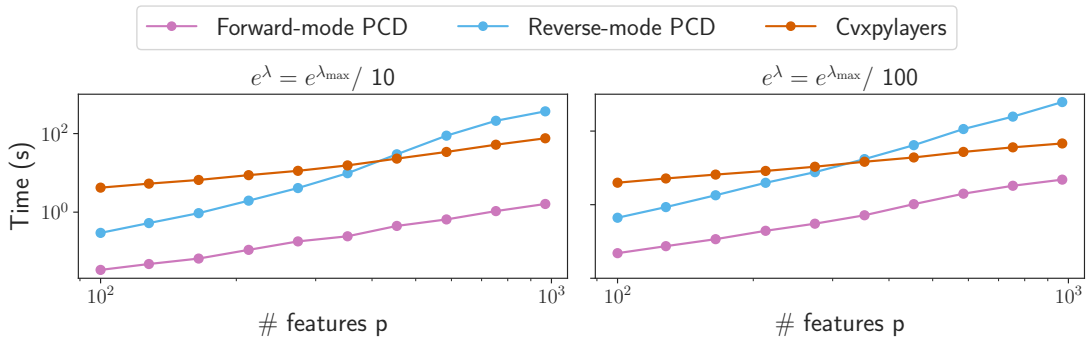


Figure 6.1 – **Lasso with hold-out criterion:** time comparison on the gina dataset to compute a single hypergradient as a function of the number of features, for two values of λ, $e^\lambda = e^{\lambda_{\max}}/10$ (left) and $e^\lambda = e^{\lambda_{\max}}/100$ (right).

duality gap of 10^{-6} for the forward-mode and the reverse-mode. `cvxpylayers` relies on `cvxpy`, solving Problem (6.1) using a splitting conic solver (O'Donoghue et al., 2019). Since the termination criterion of the splitting conic solver is not exactly the duality gap (O'Donoghue et al., 2016, Sec. 3.5), we used the default tolerance of 10^{-4}. The hypergradient $\nabla\mathcal{L}(\lambda)$ was computed for hold-out mean squared error (see Table 6.2).

The forward-mode differentiation of proximal coordinate descent is one order of magnitude faster than `cvxpylayers` and two orders of magnitude faster than the reverse-mode differentiation of proximal coordinate descent. The larger the value of λ, the sparser the coefficients β are, leading to significant speedups in this regime. This performance is in accordance with the lower time cost of the forward mode in Table 6.2.

Combining implicit differentiation with state-of-the art solvers (Figures 6.2 and 6.3). We now compare the different approaches described in Section 6.3:

- Forward-mode differentiation of proximal coordinate descent (Algorithm 6.3).
- Implicit differentiation (Algorithm 6.5) with proximal coordinate descent to solve the inner problem. For efficiency, this solver was coded in `Numba` (Lam et al., 2015).
- Implicit differentiation (Algorithm 6.5) with state-of-the-art algorithm to solve the inner problem: we used `Celer` (Massias et al., 2020b) for the Lasso, and `Lightning` (Blondel and Pedregosa, 2016) for the SVM.

Figure 6.2 shows for three datasets and two values of regularization parameters the absolute difference between the exact hypergradient and the approximate hypergradient obtained via multiple algorithms as a function of time. Figure 6.3 reports similar results for the SVM, on the same datasets, except *news20*, which is not well suited for SVM, due to limited number of samples.

First, it demonstrates that implicit differentiation methods are faster than the forward-mode of proximal coordinate descent (pink). This illustrates the benefits of restricting the gradient computation to the support of the Jacobian, as described in Section 6.3.5. Second, thanks to the flexibility of our approach, we obtain additional speed-ups by combining implicit differentiation with a state-of-the-art solver, `Celer`. The resulting method (orange) significantly improves over implicit differentiation using a vanilla proximal coordinate descent (green).

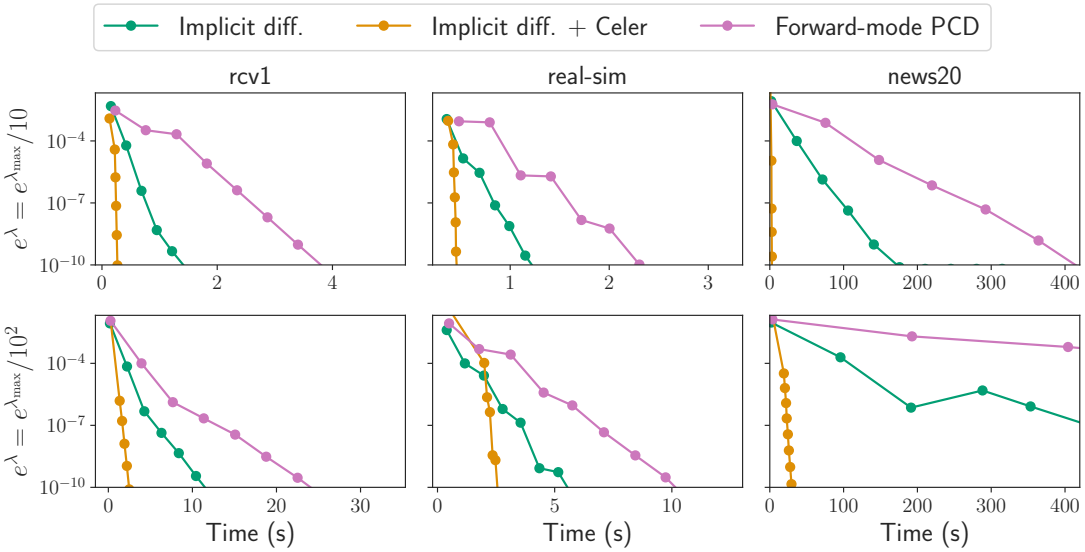


Figure 6.2 – **Lasso with hold-out criterion:** absolute difference between the exact hypergradient (using $\hat{\beta}$) and the iterate hypergradient (using $\beta^{(k)}$) of the Lasso as a function of time. Results are for three datasets and two different regularization parameters. "Implicit diff. + `Celer`)" uses `Celer` (Massias et al., 2020b) instead of our proximal coordinate descent implementation.

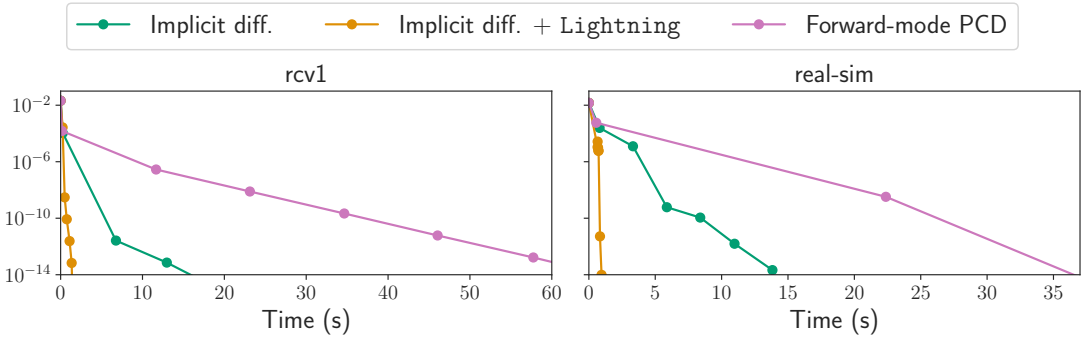


Figure 6.3 – **SVM with hold-out criterion:** absolute difference between the exact hypergradient (using $\hat{\beta}$) and the iterate hypergradient (using $\beta^{(k)}$) of the SVM as a function of time. "Implicit diff. + `Lightning`" uses `Lightning` (Blondel and Pedregosa, 2016), instead of our proximal coordinate descent implementation.

6.4.2 Resolution of the bilevel optimization problem

In this section we compare multiple methods to find the optimal hyperparameters for the Lasso, elastic net and multiclass sparse logistic regression. The following methods are compared:

- **Grid-search**: for the Lasso and the elastic net, the number of hyperparameters is small, and grid-search is tractable. For the Lasso we chose a grid of 100 hyperparameters λ, uniformly spaced between $\lambda_{\max} - \ln(10^4)$ and $\lambda_{\max}$. For the elastic net we chose for each of the two hyperparameters a grid of 10 values uniformly spaced between $\lambda_{\max}$ and $\lambda_{\max} - \ln(10^4)$. The product grid thus has 10^2 points.

- **Random-search**: we chose 30 values of λ sampled uniformly between $\lambda_{\max}$ and $\lambda_{\max} - \ln(10^4)$ for each hyperparameter. For the elastic net we chose 30 points sampled uniformly in $[\lambda_{\max} - \ln(10^4), \lambda_{\max}] \times [\lambda_{\max} - \ln(10^4), \lambda_{\max}]$.
- **SMBO**: this algorithm is SMBO using as criterion expected improvement (EI) and the Tree-structured Parzen Estimator (TPE) as model. First it evaluates $\mathcal{L}$ using 5 values of λ, chosen uniformly at random between $\lambda_{\max}$ and $\lambda_{\max} - \ln(10^4)$. Then a TPE model is fitted on the data points $(\lambda^{(1)}, \mathcal{L}(\lambda^{(1)})), \dots, (\lambda^{(5)}, \mathcal{L}(\lambda^{(5)}))$. Iteratively, the EI is used to choose the next point to evaluate $\mathcal{L}$ at, and this value is used to update the model. We used the `hyperopt` implementation (Bergstra et al., 2013).
- **1st order**: first-order method with exact gradient (Algorithm 6.6 with constant tolerances $\epsilon_i = 10^{-6}$), with $\lambda_{\max} - \ln(10^2)$ as a starting point.
- **1st order approx**: a first-order method using approximate gradient (Algorithm 6.6 with tolerances ϵ_i, geometrically decreasing from 10^{-2} to 10^{-6}), with $\lambda_{\max} - \ln(10^2)$ as a starting point.

Outer criterion. In the Lasso and elastic net experiments, we pick a K-fold CV loss as outer criterion[5]. Hence, the dataset (X, y) is partitioned into K hold-out datasets $(X^{\text{train}_k}, y^{\text{train}_k}), (X^{\text{val}_k}, y^{\text{val}_k})$. The bilevel optimization problems then write:

$$\begin{aligned} &\underset{\lambda=(\lambda_1,\lambda_2)\in\mathbb{R}^2}{\arg\min}\ \mathcal{L}(\lambda) = \frac{1}{K}\sum_{k=1}^{K} \|y^{\text{val}_k} - X^{\text{val}_k}\hat{\beta}^{(\lambda,k)}\|_2^2 \\ &\text{s.t. } \hat{\beta}^{(\lambda,k)} \in \underset{\beta\in\mathbb{R}^p}{\arg\min}\ \frac{1}{2n}\left\|y^{\text{train}_k} - X^{\text{train}_k}\beta\right\|_2^2 + e^{\lambda_1}\|\beta\|_1 + \frac{e^{\lambda_2}}{2}\|\beta\|_2^2, \quad \forall k \in [K]\ , \end{aligned} \tag{6.16}$$

while Lasso CV is obtained taking $\lambda_2 \to -\infty$ in the former. By considering an extended variable $\beta \in \mathbb{R}^{K\times p}$, cross-validation can be cast as an instance of Problem (6.2).

Figure 6.4 represents the cross-validation loss in Lasso CV as a function of the regularization parameter λ (black curve, three top rows) and as a function of time (bottom). Each point corresponds to the evaluation of the cross-validation criterion for one λ value. The top rows show cross-validation loss as a function of λ, for the grid-search, the SMBO optimizer and the first-order method. The lightest crosses correspond to the first iterations of the algorithm and the darkest, to the last ones. For instance, Lasso grid-search starts to evaluate the cross-validation function with $\lambda = \lambda_{\max}$ and then decreases to $\lambda = \lambda_{\max} - \ln(10^4)$. On all the datasets, first-order methods are faster to find the optimal regularization parameter, requiring only 5 iterations.

Figure 6.5 represents the level sets of the cross-validation loss for the elastic net (three top rows) and the cross-validation loss as a function of time (bottom). One can see that after 5 iterations the SMBO algorithm (blue crosses) suddenly slows down (bottom) as the hyperparameter suggested by the algorithm leads to a costly optimization problem to solve, while first-order methods converge quickly as for Lasso CV. In the present context, inner problems are slower to solve for low values of the regularization parameters.

Multiclass sparse logistic regression (# classes hyperparameters, Figure 6.6). We consider a multiclass classification problem with q classes. The design matrix is noted $X \in \mathbb{R}^{n\times p}$, and the target variable $y \in \{1, \dots, q\}^n$. We chose to use a one-versus-

[5] In our experiments the default choice is $K = 5$.

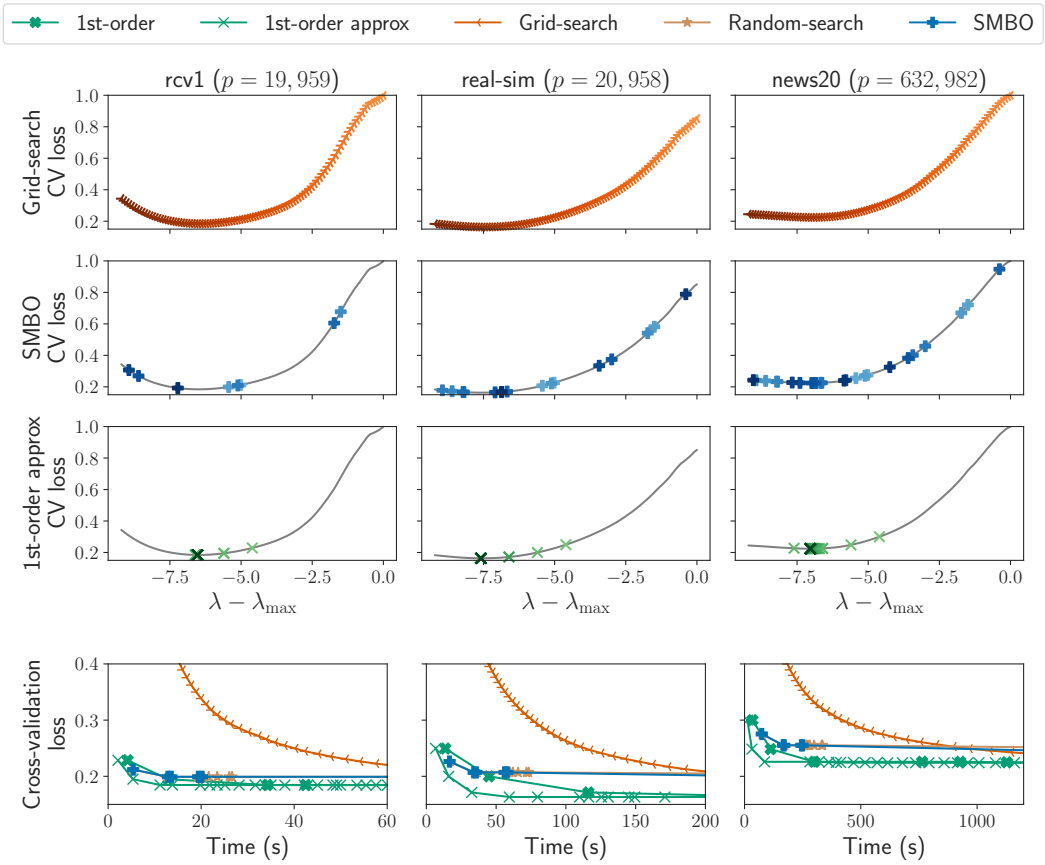


Figure 6.4 – **Lasso with cross-validation criterion:** cross-validation loss as a function of λ (black line, top) and as a function of time (bottom). Lighter markers correspond to earlier iterations of the algorithm.

all model with q regularization parameters. We use a binary cross-entropy for the inner loss:

$$\psi^k(\beta, \lambda_k; X, y) \triangleq -\frac{1}{n}\sum_{i=1}^{n}\Big(\mathbb{1}_{y_i=k}\ln(\sigma(X_{i:}\beta)) + (1-\mathbb{1}_{y_i=k})\ln(1-\sigma(X_{i:}\beta))\Big) + e^{\lambda_k}\|\beta\|_1 \ ,$$

and a multiclass cross-entropy for the outer criterion:

$$\mathcal{C}\left(\hat{\beta}^{(\lambda_1)}, \dots, \hat{\beta}^{(\lambda_q)}; X, y\right) \triangleq -\sum_{i=1}^{n}\sum_{k=1}^{q}\ln\left(\frac{e^{X_{i:}\hat{\beta}^{(\lambda_k)}}}{\sum_{l=1}^{q} e^{X_{i:}\hat{\beta}^{(\lambda_l)}}}\right)\mathbb{1}_{y_i=k} \ . \tag{6.17}$$

With a single train/test split, the bilevel problem to solve writes:

$$\begin{aligned} &\underset{\lambda \triangleq (\lambda_1,\dots,\lambda_q)\in\mathbb{R}^q}{\arg\min} \quad \mathcal{C}\left(\hat{\beta}^{(\lambda_1)}, \dots, \hat{\beta}^{(\lambda_q)}; X^{\text{test}}, y^{\text{test}}\right) \\ &\qquad \text{s.t. } \hat{\beta}^{(\lambda_k)} \in \underset{\beta\in\mathbb{R}^p}{\arg\min}\, \psi^k(\beta, \lambda_k; X^{\text{train}}, y^{\text{train}}) \quad \forall k \in [q] \ . \end{aligned} \tag{6.18}$$

Figure 6.6 represents the multiclass cross-entropy (top), the accuracy on the validation set (middle) and the accuracy on the test set (unseen data, bottom). When the number of hyperparameter is moderate ($q = 10$, on *mnist* and *usps*), the multiclass cross-entropy

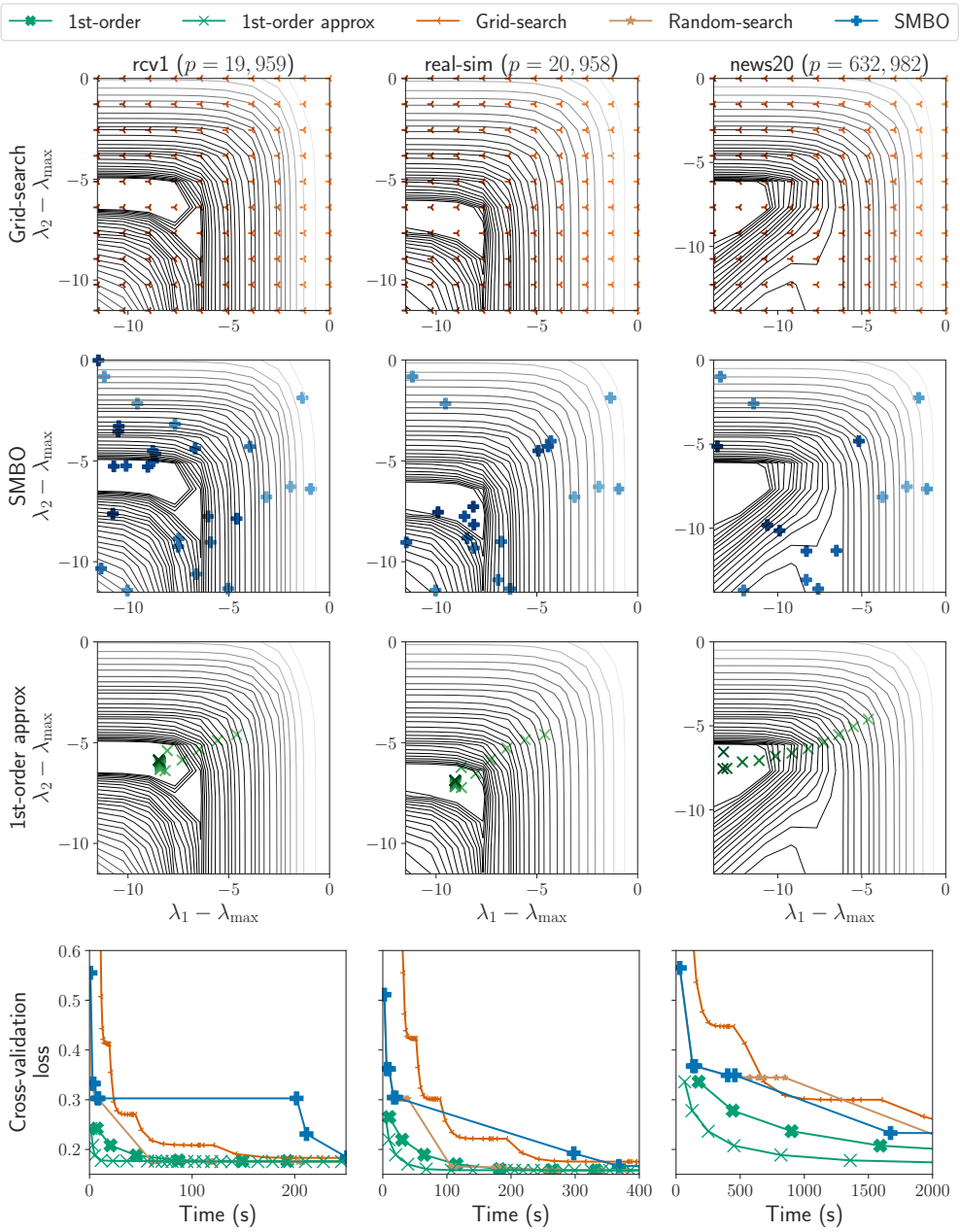


Figure 6.5 – **Elastic net cross-validation, time comparison (2 hyperparameters).** Level sets of the cross-validation loss (black lines, top) and cross-validation loss as a function of time (bottom) on *rcv1*, *real-sim* and *news20* datasets.

reached by SMBO and random techniques is as good as first-order techniques. This is expected and follows the same conclusion as Bergstra and Bengio (2012); Frazier (2018): when the number of hyperparameters is moderate, SMBO and random techniques can be used efficiently. However, when the number of hyperparameters increases (*rcv1*, $q = 53$ and *aloi*, $q = 1000$), the hyperparameter space is too large: zero-order solvers simply fail. On the contrary, first-order techniques manage to find hyperparameters leading to significantly better accuracy.

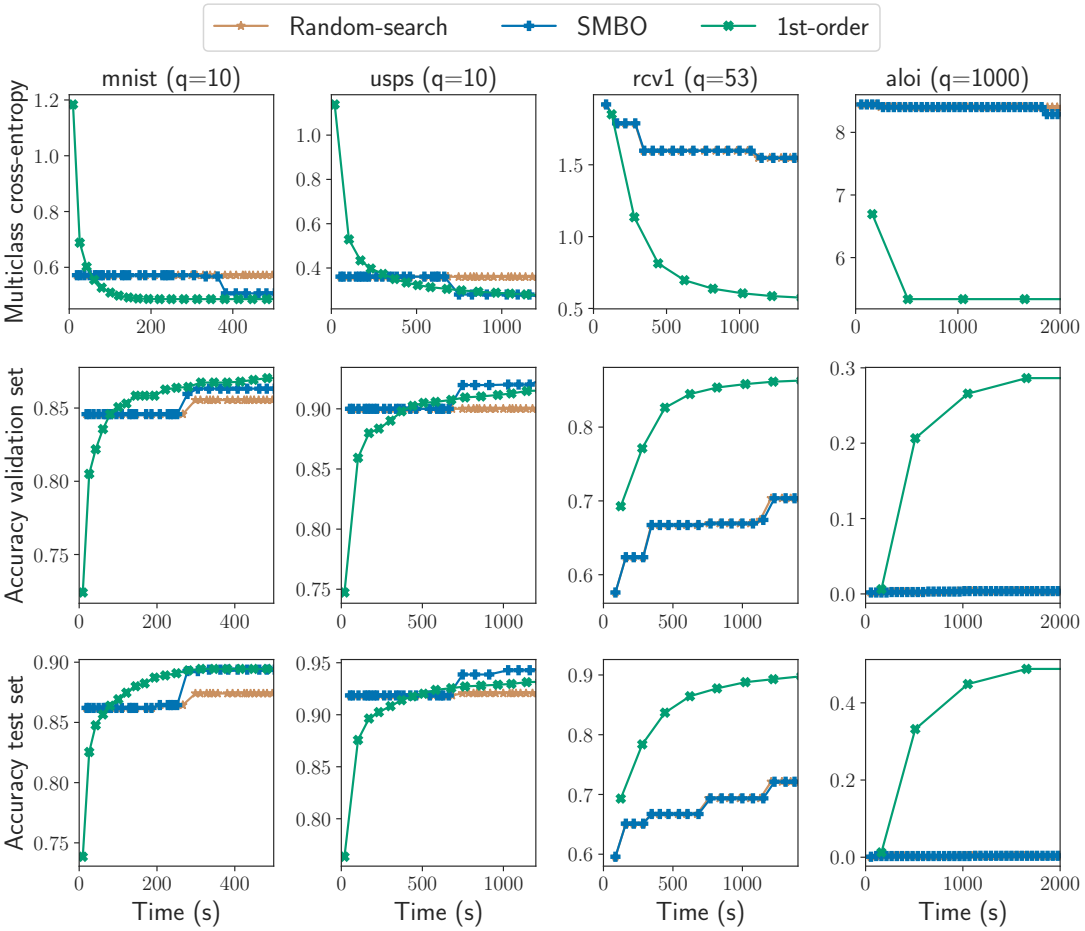


Figure 6.6 – **Multiclass sparse logistic regression hold-out, time comparison (# classes hyperparameters).** Multiclass cross-entropy (top), accuracy on the validation set (middle), and accuracy on the test set (bottom) as a function of time on *mnist*, *usps* ($q = 10$ classes), *rcv1* ($q = 53$ classes), *aloi* ($q = 1000$ classes).

6.5 Conclusion

In this work we considered the problem of hyperparameter optimization to select the regularization parameter of linear models with nonsmooth objective. Casting this problem as a bilevel optimization problem, we proposed to use first-order methods. We showed that the usual automatic differentiation techniques, implicit differentiation, forward and reverse modes, can be used to compute the hypergradient, despite the non-smoothness of the inner problem. Experimentally, we showed the interest of first-order techniques to solve bilevel optimization on a wide range of estimators (ℓ_1 penalized methods, SVM, etc.) and datasets. The presented techniques could also be extended to other criteria (Lounici et al., 2021) or to more general bilevel optimization problems, in particular implicit differentiation could be well suited for meta-learning problems, with a potentially large number of hyperparameters (Franceschi et al., 2018).

Appendix

6.A Proof of the local linear convergence

6.A.1 Local linear convergence

We now detail the following result: an asymptotic vector autoregressive sequence, with an error term vanishing linearly to 0, converges linearly to its limit. In a more formal way:

Lemma 6.18. *Let $A \in \mathbb{R}^{p\times p}, b \in \mathbb{R}$ with $\rho(A) < 1$. Let $(\mathcal{J}^{(k)})_{k\in\mathbb{N}}$ be a sequence of $\mathbb{R}^p$ such that:*

$$\mathcal{J}^{(k+1)} = A\,\mathcal{J}^{(k)} + b + \epsilon^{(k)} \ , \tag{6.19}$$

with $(\epsilon^{(k)})_{k\in\mathbb{N}}$ a sequence which converges linearly to 0, then $(\mathcal{J}^{(k)})_{k\in\mathbb{N}}$ converges linearly to its limit $\hat{\mathcal{J}} \triangleq (\mathrm{Id} - A)^{-1}b$.

Proof Assume $(\epsilon^{(k)})_{k\in\mathbb{N}}$ converges linearly. Then, there exists $c_1 > 0, 0 < \nu < 1$ such that:

$$\|\epsilon^{(k)}\| \leq c_1\nu^k \ .$$

Applying a standard result on spectral norms (see Polyak 1987, Chapter 2, Lemma 1) yields a bound on $\|A^k\|_2$. More precisely, for every $\delta > 0$ there is a constant $c_2(\delta) = c_2$ such that

$$\|A^k\|_2 \leq c_2(\rho(A) + \delta)^k \ .$$

Without loss of generality, we consider from now on a choice of δ such that $\rho(A)+\delta < 1$. Since $\hat{\mathcal{J}} = (\mathrm{Id} - A)^{-1}b$ the limit $\hat{\mathcal{J}}$ of the sequence satisfies:

$$\hat{\mathcal{J}} = A\hat{\mathcal{J}} + b \ . \tag{6.20}$$

Taking the difference between Equations (6.19) and (6.20) yields:

$$\mathcal{J}^{(k+1)} - \hat{\mathcal{J}} = A(\mathcal{J}^{(k)} - \hat{\mathcal{J}}) + \epsilon^{(k)} \ . \tag{6.21}$$

Unrolling Equation (6.21) yields $\mathcal{J}^{(k+1)} - \hat{\mathcal{J}} = A^{k+1}(\mathcal{J}^{(0)} - \mathcal{J}) + \sum_{k'=0}^{k} A^{k'}\epsilon^{(k-k')}$. Taking the norm on both sides and using the triangle inequality leads to

$$\begin{aligned}
\|\mathcal{J}^{(k+1)} - \hat{\mathcal{J}}\|_2 &\leq \|A^{k+1}(\mathcal{J}^{(0)} - \mathcal{J})\|_2 + \sum_{k'=0}^{k}\|A^{k'}\|_2\|\epsilon^{(k-k')}\| \\
&\leq \|A^{k+1}\|_2 \cdot \|\mathcal{J}^{(0)} - \hat{\mathcal{J}}\|_2 + c_1\sum_{k'=0}^{k}\|A^{k'}\|_2 \cdot \nu^{k-k'} \\
&\leq c_2(\rho(A) + \delta)^{k+1} \cdot \|\mathcal{J}^{(0)} - \hat{\mathcal{J}}\|_2 + c_1\sum_{k'=0}^{k} c_2(\rho(A) + \delta)^{k'}\nu^{k-k'}
\end{aligned}$$

We can now split the last summand in two parts and obtain the following bound, reminding that $\rho(A) + \delta < 1$:

$$\begin{aligned}\|\mathcal{J}^{(k+1)} - \hat{\mathcal{J}}\|_2 &\leq c_2(\rho(A)+\delta)^{k+1} \cdot \|\mathcal{J}^{(0)} - \hat{\mathcal{J}}\|_2 \\ &\quad + c_1 c_2 \left(\sum_{k'=0}^{k/2} (\rho(A)+\delta)^{k'} \nu^{k-k'} + \sum_{k'=k/2}^{k} (\rho(A)+\delta)^{k'} \nu^{k-k'} \right) \\ &\leq c_2(\rho(A)+\delta)^{k+1} \cdot \|\mathcal{J}^{(0)} - \hat{\mathcal{J}}\|_2 + \frac{c_1 c_2 (\rho(A)+\delta)}{1-\rho(A)-\delta} \sqrt{\nu}^{k} \\ &\quad + \frac{c_1 c_2 \nu}{1-\nu} \sqrt{(\rho(A)+\delta)}^{k} \ .\end{aligned}$$

Thus, $(\mathcal{J}^{(k)})_{k\in\mathbb{N}}$ converges linearly towards its limit $\hat{\mathcal{J}}$. ■

Theorem 6.12 (Local linear convergence of the Jacobian)**.** *Let $0 < \gamma \leq 1/L$. Suppose Assumptions 6.2, 6.3 and 6.6 hold. Let $\lambda \in \mathbb{R}^r$, Λ be a neighborhood of λ, and $\Gamma^\Lambda \triangleq \left\{ \hat{\beta}^{(\lambda)} - \gamma \nabla f(\hat{\beta}^{(\lambda)}) \ : \ \lambda \in \Lambda \right\}$. In addition, suppose hypotheses (H1) to (H4) from Theorem 6.9 are satisfied and the sequence $(\beta^{(k)})_{k\in\mathbb{N}}$ generated by Algorithm 6.1 (respectively by Algorithm 6.3) converges toward $\hat{\beta}$.*

Then, the sequence of Jacobians $(\mathcal{J}^{(k)})_{k\geq 0}$ generated by the forward-mode differentiation of proximal gradient descent (Algorithm 6.1) (respectively by forward-mode differentiation of proximal coordinate descent, Algorithm 6.3) converges locally linearly towards $\hat{\mathcal{J}}$.

Proof We first prove Theorem 6.12 for proximal gradient descent.

Proximal gradient descent case. Solving Problem (6.1) with proximal gradient descent leads to the following updates:

$$\beta^{(k+1)} = \operatorname{prox}_{\gamma g(\cdot,\lambda)}(\underbrace{\beta^{(k)} - \gamma \nabla f(\beta^{(k)})}_{z^{(k)}}) \ . \tag{6.22}$$

Consider the following sequence $(\mathcal{J}^{(k)})_{k\in\mathbb{N}}$ defined by:

$$\mathcal{J}^{(k+1)} = \partial_z \operatorname{prox}_{\gamma g(\cdot,\lambda)}(z^{(k)}) \odot \left(\mathrm{Id} - \gamma \nabla^2 f(\beta^{(k)}) \right) \mathcal{J}^{(k)} + \partial_\lambda \operatorname{prox}_{\gamma g(\cdot,\lambda)}(z^{(k)}) \ . \tag{6.23}$$

Note that if $\operatorname{prox}_{\gamma g(\cdot,\lambda)}$ is not differentiable with respect to the first variable at $z^{(k)}$ (respectively with respect to the second variable λ), any weak Jacobian can be used. When (H3) holds, differentiating Equation (6.22) with respect to λ yields exactly Equation (6.23).

Assumptions 6.2 to 6.4 and 6.6 and the convergence of $(\beta^{(k)})$ toward $\hat{\beta}$ ensure proximal gradient descent algorithm has finite identification property (Liang et al., 2014, Thm. 3.1): we note K the iteration when identification is achieved. As before, the separability of g, Assumptions 6.2 to 6.4 and 6.6 ensure (see Theorems 2.13 and 2.19) $\partial_z \operatorname{prox}_{\gamma g(\cdot,\lambda)}(z^k)_{\hat{S}^c} = 0$, for all $k \geq K$. Thus, for all $k \geq K$,

$$\mathcal{J}^{(k)}_{\hat{S}^c:} = \hat{\mathcal{J}}_{\hat{S}^c:} = \partial_\lambda \operatorname{prox}_{\gamma g(\cdot,\lambda)}(z^{(k)})_{\hat{S}^c:} \ .$$

The updates of the Jacobian then become:

$$\mathcal{J}^{(k+1)}_{\hat{S}:} = \partial_z \operatorname{prox}_{\gamma g(\cdot,\lambda)}(z^{(k)})_{\hat{S}} \odot \left(\operatorname{Id} - \gamma \nabla^2_{\hat{S},\hat{S}} f(\beta^{(k)})\right) \mathcal{J}^{(k)}_{\hat{S}:} + \partial_\lambda \operatorname{prox}_{\gamma g(\cdot,\lambda)}(z^{(k)})_{\hat{S}:} \ .$$

From Assumption 6.6, we have that f is locally $\mathcal{C}^3$ at $\hat{\beta}$, $g(\cdot,\lambda)$ is locally $\mathcal{C}^2$ at $\hat{\beta}$ hence $\operatorname{prox}_{g(\cdot,\lambda)}$ is locally $\mathcal{C}^2$. The function $\beta \mapsto \partial_z \operatorname{prox}_{\gamma g(\cdot,\lambda)}(\beta - \gamma \nabla f(\beta))_{\hat{S}} \odot (\operatorname{Id} - \gamma \nabla^2_{\hat{S},\hat{S}} f(\beta))$ is differentiable at $\hat{\beta}$. Using (H4) we have that $\beta \mapsto \partial_\lambda \operatorname{prox}_{\gamma g(\cdot,\lambda)}(\beta - \gamma \nabla f(\beta))_{\hat{S}:}$ is also differentiable at $\hat{\beta}$. Using the Taylor expansion of the previous functions yields:

$$\mathcal{J}^{(k+1)}_{\hat{S}:} = \underbrace{\partial_z \operatorname{prox}_{\gamma g(\cdot,\lambda)}(\hat{z})_{\hat{S}} \odot \left(\operatorname{Id} - \gamma \nabla^2_{\hat{S},\hat{S}} f(\hat{\beta})\right)}_{A} \mathcal{J}^{(k)}_{\hat{S}:} + \underbrace{\partial_\lambda \operatorname{prox}_{\gamma g(\cdot,\lambda)}(\hat{z})_{\hat{S}:}}_{b} + \underbrace{o(\|\beta^{(k)} - \hat{\beta}\|)}_{\epsilon^{(k)}} \ . \tag{6.24}$$

Thus, for $0 < \gamma \leq 1/L$,

$$\rho(A) \leq \|A\|_2 \leq \underbrace{\|\partial_z \operatorname{prox}_{\gamma g(\cdot,\lambda)}(\hat{z})_{\hat{S}}\|}_{\leq 1 \text{ (non-expansiveness)}} \cdot \underbrace{\|\operatorname{Id} - \gamma \nabla^2_{\hat{S},\hat{S}} f(\hat{\beta})\|_2}_{<1 \text{ (Assumption 6.7 and } 0<\gamma\leq 1/L)} < 1 \ . \tag{6.25}$$

The inequality on the derivative of the proximal operator comes from the non-expansiveness of proximal operators. The second inequality comes from Assumption 6.7 and $0 < \gamma \leq 1/L$.

Assumptions 6.2 to 6.4, 6.6 and 6.7 and the convergence of $(\beta^{(k)})$ toward $\hat{\beta}$ ensure $(\beta^{(k)})_{k\in\mathbb{N}}$ converges locally linearly (Liang et al., 2014, Thm. 3.1). The asymptotic autoregressive sequence in Equation (6.24), $\rho(A) < 1$, and the local linear convergence of $(\epsilon^{(k)})_{k\in\mathbb{N}}$, yield our result using Lemma 6.18.

We now prove Theorem 6.12 for proximal coordinate descent.

Proximal coordinate descent. Compared to proximal gradient descent, the analysis of coordinate descent requires studying functions defined as a the composition of p applications, each of them only modifying one coordinate.

Coordinate descent updates read as follows:

$$\beta_j^{(k,j)} = \operatorname{prox}_{\gamma_j g_j(\cdot,\lambda)} \underbrace{\left(\beta_j^{(k,j-1)} - \gamma_j \nabla_j f(\beta^{(k,j-1)})\right)}_{\triangleq z_j^{(k,j-1)}} \ . \tag{6.26}$$

We consider the following sequence:

$$\begin{aligned} \mathcal{J}^{(k,j)}_{j:} = \ & \partial_z \operatorname{prox}_{\gamma_j g_j(\cdot,\lambda)}(z_j^{(k,j-1)}) \left(\mathcal{J}^{(k,j-1)}_{j:} - \gamma_j \nabla^2_{j:} f(\beta^{(k,j-1)})\, \mathcal{J}^{(k,j-1)}\right) \\ & + \partial_\lambda \operatorname{prox}_{\gamma_j g_j(\cdot,\lambda)}(z_j^{(k,j-1)}) \ . \end{aligned} \tag{6.27}$$

Note that if $\operatorname{prox}_{\gamma g(\cdot,\lambda)}$ is not differentiable with respect to the first variable at $z^{(k)}$ (respectively with respect to the second variable λ), any weak Jacobian can be used. When (H3) holds, differentiating Equation (6.26) with respect to λ yields exactly Equation (6.27).

Assumptions 6.2 to 6.4 and 6.6 and the convergence of $(\beta^{(k)})_{k\in\mathbb{N}}$ toward $\hat{\beta}$ ensure proximal coordinate descent has finite identification property (Klopfenstein et al., 2020, Thm. 1): we note K the iteration when identification is achieved. Once the generalized support $\hat{S}$ (of cardinality $\hat{s}$) has been identified, we have that for all $k \geq K$, $\beta^{(k)}_{\hat{S}^c} = \hat{\beta}_{\hat{S}}$ and for any $j \in \hat{S}^c$, $\partial_z \operatorname{prox}_{\gamma_j g_j(\cdot,\lambda)}(z_j^{(k,j-1)}) = 0$. Thus $\mathcal{J}_{j:}^{(k,j)} = \partial_\lambda \operatorname{prox}_{\gamma_j g_j(\cdot,\lambda)}(z_j^{(k,j-1)})$. Then, we have that for any $j \in \hat{S}$ and for all $k \geq K$:

$$\begin{aligned}\mathcal{J}_{j:}^{(k,j)} = \partial_z \operatorname{prox}_{\gamma_j g_j(\cdot,\lambda)}(z_j^{(k,j-1)}) \left(\mathcal{J}_{j:}^{(k,j-1)} - \gamma_j \nabla^2_{j,\hat{S}} f(\beta^{(k,j-1)})\, \mathcal{J}_{\hat{S}:}^{(k,j-1)}\right) \\ + \partial_\lambda \operatorname{prox}_{\gamma_j g_j(\cdot,\lambda)}(z_j^{(k,j-1)}) - \gamma_j \partial_z \operatorname{prox}_{\gamma_j g_j(\cdot,\lambda)}(z_j^{(k,j-1)}) \nabla^2_{j,\hat{S}^c} f(\beta^{(k,j-1)})\, \mathcal{J}_{\hat{S}^c:}^{(k,j-1)} \quad .\end{aligned}$$

Let $e_1, \dots, e_{\hat{s}}$ be the vectors of the canonical basis of $\mathbb{R}^{\hat{s}}$. We can consider the applications

$$\begin{aligned}&\mathbb{R}^p \to \mathbb{R}^{\hat{s}} \\ &\beta \mapsto \partial_z \operatorname{prox}_{\gamma_j g_j(\cdot,\lambda)}\left(\beta_j - \gamma_j \nabla_j f(\beta)\right)\left(e_j - \gamma_j \nabla^2_{j,\hat{S}} f(\beta)\right) \quad ,\end{aligned}$$

and

$$\begin{aligned}&\mathbb{R}^p \to \mathbb{R}^{\hat{s}\times r} \\ &\beta \mapsto \partial_\lambda \operatorname{prox}_{\gamma_j g_j(\cdot,\lambda)}\left(\beta_j - \gamma_j \nabla_j f(\beta)\right) - \gamma_j \partial_z \operatorname{prox}_{\gamma_j g_j(\cdot,\lambda)}\left(\beta_j - \gamma_j \nabla_j f(\beta)\right) \nabla^2_{j,\hat{S}^c} f(\beta) \hat{\mathcal{J}}_{\hat{S}^c:} \quad ,\end{aligned}$$

which are both differentiable at $\hat{\beta}$ using Assumption 6.6 and (H4). The Taylor expansion of the previous functions yields:

$$\begin{aligned}\mathcal{J}_{j:}^{(k,j)} = \partial_z \operatorname{prox}_{\gamma_j g_j(\cdot,\lambda)}\left(\hat{z}_j\right)\left(e_j - \gamma_j \nabla^2_{j,\hat{S}} f(\hat{\beta})\right) \mathcal{J}_{\hat{S}:}^{(k,j-1)} \\ + \partial_\lambda \operatorname{prox}_{\gamma_j g_j(\cdot,\lambda)}\left(\hat{z}_j\right) - \gamma_j \partial_z \operatorname{prox}_{\gamma_j g_j(\cdot,\lambda)}\left(\hat{z}_j\right) \nabla^2_{j,\hat{S}^c} f(\hat{\beta})\, \mathcal{J}_{\hat{S}^c:}^{(k,j-1)} \\ + o(\|\beta^{(k,j-1)} - \hat{\beta}\|) \quad .\end{aligned}$$

Let $j_1, \dots, j_{\hat{s}}$ be the indices of the generalized support of $\hat{\beta}$. When considering a full epoch of coordinate descent, the Jacobian is obtained as the product of matrices of the form

$$A_s^\top = \left(\; e_1 \;\middle|\; \dots \;\middle|\; e_{s-1} \;\middle|\; v_{j_s} \;\middle|\; e_{s+1} \;\middle|\; \dots \;\middle|\; e_{\hat{s}} \; \right) \in \mathbb{R}^{\hat{s}\times\hat{s}} \quad ,$$

where $v_{j_s} = \partial_z \operatorname{prox}_{\gamma_{j_s} g_{j_s}}\left(\hat{z}_{j_s}\right)\left(e_s - \gamma_{j_s} \nabla^2_{j_s,\hat{S}} f(\hat{\beta})\right) \in \mathbb{R}^{\hat{s}}$. A full epoch can then be written

$$\mathcal{J}_{\hat{S}:}^{(k+1)} = \underbrace{A_{\hat{s}} A_{\hat{s}-1} \dots A_1}_{A} \mathcal{J}_{\hat{S}:}^{(k)} + b + \epsilon^{(k)} \quad ,$$

for a certain $b \in \mathbb{R}^{\hat{s}}$.

The spectral radius of A is strictly bounded by 1 (Klopfenstein et al., 2020, Lemma 8): $\rho(A) < 1$. Assumptions 6.2 to 6.4 and 6.6 and the convergence of $(\beta^{(k)})_{k\in\mathbb{N}}$ toward $\hat{\beta}$ ensure local linear convergence of $(\beta^{(k)})_{k\in\mathbb{N}}$ (Klopfenstein et al., 2020, Thm. 2). Hence,

we can write the update for the Jacobian after an update of the coordinates from 1 to p:

$$\mathcal{J}_{\hat{S}:}^{(k+1)} = A\,\mathcal{J}_{\hat{S}:}^{(k)} + b + \epsilon^{(k)} \ , \tag{6.28}$$

with $(\epsilon^{(k)})_{k\in\mathbb{N}}$ converging linearly to 0.

Recalling $\rho(A) < 1$, Lemma 6.18 and the last display yield our result using. ■

6.B Proof of the approximate gradient theorem

Theorem 6.13 (Bound on the error of approximate hypergradient)**.** *For $\lambda \in \mathbb{R}^r$, let $\hat{\beta}^{(\lambda)} \in \mathbb{R}^p$ be the exact solution of the inner Problem (6.1), and $\hat{S}$ its generalized support. Suppose Assumptions 6.2, 6.3 and 6.6 hold. Let Λ be a neighborhood of λ, and $\Gamma^\Lambda \triangleq \left\{\hat{\beta}^{(\lambda)} - \gamma\nabla f(\hat{\beta}^{(\lambda)}) \,:\, \lambda \in \Lambda\right\}$. Suppose hypotheses (H1) to (H4) from Theorem 6.9 are satisfied. In addition suppose*

(H5) The application $\beta \mapsto \nabla^2 f(\beta)$ is Lipschitz continuous.

(H6) The criterion $\beta \mapsto \nabla\mathcal{C}(\beta)$ is Lipschitz continuous.

(H7) Both optimization problems in Algorithm 6.5 are solved up to precision ϵ with support identification: $\|\beta^{(\lambda)} - \hat{\beta}^{(\lambda)}\| \leq \epsilon$, $A^\top$ is invertible, and $\|A^{-1^\top}\nabla_{\hat{S}}\mathcal{C}(\beta^{(\lambda)}) - v\| \leq \epsilon$.

Then the error on the approximate hypergradient h returned by Algorithm 6.5 is of the order of magnitude of the error ϵ on $\beta^{(\lambda)}$ and v:

$$\|\nabla\mathcal{L}(\lambda) - h\| = \mathcal{O}(\epsilon) \ .$$

Proof

Overview of the proof. Our goal is to bound the error between the approximate hypergradient h returned by Algorithm 6.5 and the true hypergradient $\nabla\mathcal{L}(\lambda)$. Following the analysis of Pedregosa (2016), two sources of approximation errors arise when computing the hypergradient:

- Approximation errors from the inexact computation of $\hat{\beta}$. Dropping the dependency with respect to λ, we denote β the approximate solution and suppose the problem is solved to precision ϵ with support identification (H7):

$$\begin{cases} \beta_{\hat{S}^c} = \hat{\beta}_{\hat{S}^c} \\ \|\beta_{\hat{S}} - \hat{\beta}_{\hat{S}}\| \leq \epsilon \ . \end{cases}$$

- Approximation errors from the approximate resolution of the linear system, using (H7) yields:

$$\|A^{-1^\top}\nabla_{\hat{S}}\mathcal{C}(\beta) - v\| \leq \epsilon \ .$$

The exact solution of the exact linear system $\hat{v}$ satisfies:

$$\hat{v} = \hat{A}^{-1\top} \nabla_{\hat{S}} \mathcal{C}(\hat{\beta}) \ ,$$

with

$$A \triangleq \mathrm{Id}_{|\hat{S}|} - \underbrace{\partial_z \operatorname{prox}_{\gamma g(\cdot,\lambda)} \left(\beta - \gamma \nabla f(\beta)\right)_{\hat{S}}}_{\triangleq C} \underbrace{\left(\mathrm{Id}_{|\hat{S}|} - \gamma \nabla^2_{\hat{S},\hat{S}} f(\beta)\right)}_{\triangleq D} \ ,$$

$$\hat{A} \triangleq \mathrm{Id}_{|\hat{S}|} - \underbrace{\partial_z \operatorname{prox}_{\gamma g(\cdot,\lambda)} \left(\hat{\beta} - \gamma \nabla f(\hat{\beta})\right)_{\hat{S}}}_{\triangleq \hat{C}} \underbrace{\left(\mathrm{Id}_{|\hat{S}|} - \gamma \nabla^2_{\hat{S},\hat{S}} f(\hat{\beta})\right)}_{\triangleq \hat{D}} \ .$$

- Using the last two points, the goal is to bound the difference between the exact hypergradient and the approximate hypergradient, $\|\nabla \mathcal{L}(\lambda) - h\|$. Following Algorithm 6.5, the exact hypergradient reads

$$\nabla \mathcal{L}(\lambda) = \hat{B}\hat{v} + \hat{\mathcal{J}}^{\top}_{\hat{S}^c:} \nabla_{\hat{S}^c} \mathcal{C}(\hat{\beta}) \ ,$$

and similarly for the approximate versions:

$$h = Bv + \mathcal{J}^{\top}_{\hat{S}^c:} \nabla_{\hat{S}^c} \mathcal{C}(\beta) \ ,$$

with

$$B \triangleq \partial_\lambda \operatorname{prox}_{\gamma g(\cdot,\lambda)} \left(\beta - \gamma \nabla f(\beta)\right)_{\hat{S}:} - \gamma \partial_z \operatorname{prox}_{\gamma g(\cdot,\lambda)} \left(\beta - \gamma \nabla f(\beta)\right)_{\hat{S}} \odot \left(\nabla^2_{\hat{S},\hat{S}^c} f(\beta)\right) \mathcal{J}_{\hat{S}^c:}$$

$$\hat{B} \triangleq \partial_\lambda \operatorname{prox}_{\gamma g(\cdot,\lambda)} \left(\hat{\beta} - \gamma \nabla f(\hat{\beta})\right)_{\hat{S}:} - \gamma \partial_z \operatorname{prox}_{\gamma g(\cdot,\lambda)} \left(\hat{\beta} - \gamma \nabla f(\hat{\beta})\right)_{\hat{S}} \odot \left(\nabla^2_{\hat{S},\hat{S}^c} f(\hat{\beta})\right) \hat{\mathcal{J}}_{\hat{S}^c:} \ .$$

We can exploit these decompositions to bound the difference between the exact hypergradient and the approximate hypergradient:

$$\begin{aligned}
\|\nabla \mathcal{L}(\lambda) - h\| &= \|\hat{B}\hat{v} - Bv + \hat{\mathcal{J}}^{\top}_{\hat{S}^c:} \nabla_{\hat{S}^c} \mathcal{C}(\hat{\beta}) - \hat{\mathcal{J}}^{\top}_{\hat{S}^c:} \nabla_{\hat{S}^c} \mathcal{C}(\beta)\| \\
&\leq \|\hat{B}\hat{v} - Bv\| + \|\hat{\mathcal{J}}^{\top}_{\hat{S}^c:} \nabla_{\hat{S}^c} \mathcal{C}(\hat{\beta}) - \hat{\mathcal{J}}^{\top}_{\hat{S}^c:} \nabla_{\hat{S}^c} \mathcal{C}(\beta)\| \\
&\leq \|\hat{B}\hat{v} - B\hat{v} + B\hat{v} - Bv\| + \|\hat{\mathcal{J}}^{\top}_{\hat{S}^c:} (\nabla_{\hat{S}^c} \mathcal{C}(\hat{\beta}) - \nabla_{\hat{S}^c} \mathcal{C}(\beta))\| \\
&\leq \|\hat{v}\| \cdot \|\hat{B} - B\| + \|B\| \cdot \|\hat{v} - v\| + L_{\mathcal{C}} \|\hat{\mathcal{J}}^{\top}_{\hat{S}^c:}\| \cdot \|\beta - \hat{\beta}\| \ . \qquad (6.29)
\end{aligned}$$

Bounding $\|\hat{v} - v\|$ and $\|\hat{B} - B\|$ in Equation (6.29) yields the desired result which is bounding the difference between the exact hypergradient and the approximate hypergradient $\|\nabla \mathcal{L}(\lambda) - h\|$.

Bound on $\|\hat{v} - v\|$. We first prove that $\|A - \hat{A}\| = \mathcal{O}(\epsilon)$. Let L_H be the Lipschitz constant of the application $\beta \mapsto \nabla^2 f(\beta)$, then we have:

$$\begin{aligned}
\|A - \hat{A}\|_2 &= \|CD - \hat{C}\hat{D}\|_2 \\
&\leq \|CD - C\hat{D}\|_2 + \|C\hat{D} - \hat{C}\hat{D}\|_2 \\
&\leq \underbrace{\|C\|_2}_{\leq 1 \text{ (non-expansiveness)}} \underbrace{\|D - \hat{D}\|_2}_{\leq L_H \|\beta - \hat{\beta}\| \text{ using (H5)}} + \underbrace{\|\hat{D}\|_2}_{\leq 1} \underbrace{\|C - \hat{C}\|_2}_{\mathcal{O}(\|\beta - \hat{\beta}\|) \text{ using (H4)}} \\
&\leq L_H \|\beta - \hat{\beta}\| + \mathcal{O}(\|\beta - \hat{\beta}\|) \\
&= \mathcal{O}(\|\beta - \hat{\beta}\|) \ . \qquad (6.30)
\end{aligned}$$

Let $\tilde{v}$ be the exact solution of the approximate system $A^\top \tilde{v} \triangleq \nabla_{\hat{S}} \mathcal{C}(\beta)$. The following conditions are met:

- $\hat{v}$ is the exact solution of the exact linear system and $\tilde{v}$ is the exact solution of the approximate linear system

$$\begin{aligned} \hat{A}^\top \hat{v} &\triangleq \nabla_{\hat{S}} \mathcal{C}(\hat{\beta}) \\ A^\top \tilde{v} &\triangleq \nabla_{\hat{S}} \mathcal{C}(\beta) \ . \end{aligned}$$

- One can control the difference between the exact matrix in the linear system $\hat{A}$ and the approximate matrix A.

$$\|A - \hat{A}\|_2 \leq \delta \|\beta - \hat{\beta}\| \ ,$$

  for a certain $\delta > 0$ (Equation (6.30)).

- One can control the difference between the two right-hand side of the linear systems

$$\|\nabla_{\hat{S}} \mathcal{C}(\beta) - \nabla_{\hat{S}} \mathcal{C}(\hat{\beta})\| \leq L_{\mathcal{C}} \|\beta - \hat{\beta}\| \ ,$$

  since $\beta \mapsto \nabla \mathcal{C}(\beta)$ is $L_{\mathcal{C}}$-Lipschitz continuous (H6).

- One can control the product of the perturbations

$$\delta \cdot \|\beta - \hat{\beta}\| \cdot \|\hat{A}^{-1}\|_2 \leq \rho < 1 \ .$$

Conditions are met to apply the result by Higham (2002, Thm 7.2), which leads to

$$\begin{aligned} \|\tilde{v} - \hat{v}\| &\leq \frac{\epsilon}{1 - \epsilon \|\hat{A}^{-1}\| \delta} \left(L_{\mathcal{C}} \|\hat{A}^{-1}\| + \|\hat{v}\| \cdot \|\hat{A}^{-1}\| \delta \right) \\ &\leq \frac{\epsilon}{1 - \rho} \left(L_{\mathcal{C}} \|\hat{A}^{-1}\| + \|\hat{v}\| \cdot \|\hat{A}^{-1}\| \delta \right) \\ &= \mathcal{O}(\epsilon) \ . \end{aligned} \tag{6.31}$$

The bound on $\|\tilde{v} - \hat{v}\|$ finally yields a bound on the first quantity in Equation (6.3), $\|v - \hat{v}\|$:

$$\begin{aligned} \|v - \hat{v}\| &= \|v - \tilde{v} + \tilde{v} - \hat{v}\| \\ &\leq \|v - \tilde{v}\| + \|\tilde{v} - \hat{v}\| \\ &\leq \|A^{-1} A (v - \tilde{v})\| + \|\tilde{v} - \hat{v}\| \\ &\leq \|A^{-1}\|_2 \times \underbrace{\left\| A(v - \tilde{v}) \right\|}_{\leq \epsilon \text{ (H7)}} + \underbrace{\|\tilde{v} - \hat{v}\|}_{\mathcal{O}(\epsilon) \text{ (Equation (6.31))}} \\ &= \mathcal{O}(\epsilon) \ . \end{aligned} \tag{6.32}$$

Bound on $\|B - \hat{B}\|_2$. We now bound the second quantity in Equation (6.3) $\|B - \hat{B}\|_2$:

$$\begin{aligned}
\|B - \hat{B}\|_2 &\leq \|\partial_\lambda \operatorname{prox}_{\gamma g(\cdot,\lambda)}(\beta - \gamma \nabla f(\beta))_{\hat{S}:} - \partial_\lambda \operatorname{prox}_{\gamma g(\cdot,\lambda)}(\hat{\beta} - \gamma \nabla f(\hat{\beta}))_{\hat{S}:}\|_2 \\
&\quad + \gamma \|\partial_z \operatorname{prox}_{\gamma g(\cdot,\lambda)}(\hat{\beta} - \gamma \nabla f(\hat{\beta}))_{\hat{S}} \nabla^2_{\hat{S},\hat{S}^c} f(\hat{\beta}) \hat{\mathcal{J}}_{\hat{S}^c:} \qquad (6.33) \\
&\quad - \partial_z \operatorname{prox}_{\gamma g(\cdot,\lambda)}(\beta - \gamma \nabla f(\beta))_{\hat{S}} \nabla^2_{\hat{S},\hat{S}^c} f(\beta) \hat{\mathcal{J}}_{\hat{S}^c:}\|_2 \\
&\leq L_1 \|\beta - \gamma \nabla f(\beta)_{\hat{S}:} - \hat{\beta} + \gamma \nabla f(\hat{\beta})\| \text{ using (H4)} \\
&\quad + L_2 \|\hat{\beta} - \beta\| \cdot \|\hat{\mathcal{J}}_{\hat{S}^c:}\| \text{ using (H4) and Assumption 6.6} \\
&= \mathcal{O}(\|\hat{\beta} - \beta\|) \ . \qquad (6.34)
\end{aligned}$$

Plugging Equations (6.32) and (6.34) into Equation (6.3) yields the desired result: $\|\nabla \mathcal{L}(\lambda) - h\| = \mathcal{O}(\epsilon)$. ■

Conclusion and perspectives

«La chose la plus importante à toute la vie est le choix du métier : le hasard en dispose.»

In this thesis we first investigated some theoretical properties of coordinate descent. We showed finite time model identification and local linear convergence. Relying on these two properties, we proposed an Anderson accelerated version of cyclic proximal coordinate descent: `andersoncd`. It has been implemented in the largest python brain signal processing package, and is now the default solver for sparse signal estimation.

Then we explored a statistical approach to set the regularization parameter of Lasso-type problems. We showed that partial smoothing preserves the statistical properties of pivotal estimators, while making the optimization problem amenable to efficient coordinate descent algorithms. We have extensively illustrated the interest of these estimators on real M/EEG data: on visual and auditory tasks. However this approach relies on unrealistic hypotheses and quantities unknown in practice.

Finally we investigated hyperparameter selection through the lens of bilevel optimization. We extended usual first-order methods for bilevel optimization problems with smooth inner problems, to nonsmooth inner problems. Leveraging sparsity, we were able to speed-up hypergradient computations. This enabled efficient sparse linear models calibration with a large number of hyperparameters.

Over the last decades, convex optimization has lead to a wealth of algorithms to solve single-level optimization problems, leaving the question of model selection of optimization-based estimators to statisticians. We hope we convinced the reader of the interest of solving directly bilevel optimization problems to moderate precision, instead of single-level optimization problems to machine precision.

Even in the case of smooth inner problems, practical packages, such as `scikit-learn` or `glmnet`, rely on zero-order methods for hyperparameter optimization, through grid-search, random-search, or Bayesian techniques. First-order method packages exist (`hoag`, Pedregosa 2016), but they still rely on other hyperparameters requiring manual calibration: in practice, they do not lead to automated algorithms. In particular, they are currently based on nested for-loops (Algorithm 6.6), which can lack of clear stopping criterion to avoid solving the inner optimisation problems to unnecessary precision. Some "online" algorithms have been proposed, but currently mostly rely on heuristics (Baydin et al., 2017; MacKay et al., 2019; Vicol et al., 2021).

Note also that bilevel optimization is studied in game theory for almost a century under the name of Stackelberg equilibrium (von Stackelberg, 1934), for which a wealth of specific algorithms have been developed (Korpelevich, 1976; Rakhlin and Sridharan, 2013; Mescheder et al., 2017). Unifying bilevel optimization and Stackelberg game theory (Sinha et al., 2017), with the design of principled, efficient, and practical "online" algorithms appears a major area of research for the coming years. This would allow to select the regularization parameter, without even solving exactly one inner problem.

Bibliography

P. Ablin, G. Peyré, and T. Moreau. Super-efficiency of automatic differentiation for functions defined as a minimum. In *ICML*, pages 32–41. PMLR, 2020. page 135

A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and J. Z. Kolter. Differentiable convex optimization layers. In *NeurIPS*, pages 9558–9570, 2019. pages 138, 146

H. Akaike. A new look at the statistical model identification. *IEEE Trans. Automat. Control*, AC-19:716–723, 1974. pages 25, 131

D. M. Allen. The relationship between variable selection and data agumentation and a method for prediction. *technometrics*, 16(1):125–127, 1974. page 26

B. Amos and J. Z. Kolter. Optnet: Differentiable optimization as a layer in neural networks. In *ICML*, volume 70, pages 136–145, 2017. page 138

D. G. Anderson. Iterative procedures for nonlinear integral equations. *Journal of the ACM*, 12(4):547–560, 1965. page 64

A. Argyriou, T. Evgeniou, and M. Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008. pages 21, 109, 112

S. Arlot and A. Celisse. A survey of cross-validation procedures for model selection. *Statistics surveys*, 4:40–79, 2010. pages 26, 131

F. Bach. Consistency of the group Lasso and multiple kernel learning. *J. Mach. Learn. Res.*, 9:1179–1225, 2008. page 46

F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Convex optimization with sparsity-inducing norms. *Foundations and Trends in Machine Learning*, 4(1):1–106, 2012. pages 82, 109, 112

S. Bai, J. Z. Kolter, and V. Koltun. Deep equilibrium models. *NeurIPS*, 2019. page 135

S. Bai, V. Koltun, and J. Z. Kolter. Multiscale deep equilibrium models. *NeurIPS*, 2020. page 135

S. Baillet, J. C. Mosher, and R. M. Leahy. Electromagnetic brain mapping. *IEEE Signal processing magazine*, 18(6):14–30, 2001. pages 17, 19

M. Barré, A. Taylor, and A. d'Aspremont. Convergence of constrained Anderson acceleration. *arXiv preprint arXiv:2010.15482*, 2020. page 70

H. H. Bauschke and P. L. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*. Springer, New York, 2011. page 89

A. G. Baydin, R. Cornish, D. M. Rubio, M. Schmidt, and F. Wood. Online learning rate adaptation with hypergradient descent. *arXiv preprint arXiv:1703.04782*, 2017. page 161

A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind. Automatic differentiation in machine learning: a survey. *J. Mach. Learn. Res.*, 18(153):1–43, 2018. page 135

A. Beck. *First-Order Methods in Optimization*, volume 25. SIAM, 2017. pages 63, 108

A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sci.*, 2(1):183–202, 2009. pages 63, 74

A. Beck and M. Teboulle. Smoothing and first order methods: A unified framework. *SIAM J. Optim.*, 22(2):557–580, 2012. pages 83, 104, 110, 111, 112

A. Beck and L. Tetruashvili. On the convergence of block coordinate type methods. *SIAM J. Imaging Sci.*, 23(4):651–694, 2013. pages 31, 35, 36, 45, 46

S. Behnel, R. Bradshaw, C. Citro, L. Dalcin, D. S. Seljebotn, and K. Smith. Cython: The best of both worlds. *Computing in Science Engineering*, 13(2):31 –39, 2011. page 73

Y. Bekhti, F. Lucka, J. Salmon, and A. Gramfort. A hierarchical bayesian perspective on majorization-minimization for non-convex sparse regression: application to m/eeg source imaging. *Inverse Problems*, 34(8):085010, 2018. page 24

A. Belloni, V. Chernozhukov, and L. Wang. Square-root Lasso: pivotal recovery of sparse signals via conic programming. *Biometrika*, 98(4):791–806, 2011. pages 22, 23, 82, 83, 103, 131

Y. Bengio. Gradient-based optimization of hyperparameters. *Neural computation*, 12 (8):1889–1900, 2000. pages 132, 134

H. Berger. Über das elektroenkephalogramm des menschen. *Archiv für psychiatrie und nervenkrankheiten*, 87(1):527–570, 1929. page 18

J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *J. Mach. Learn. Res.*, 13(2), 2012. pages 131, 151

J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for hyper-parameter optimization. In *NeurIPS*, 2011. page 131

J. Bergstra, D. Yamins, and D. D. Cox. Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms. In *Proceedings of the 12th Python in science conference*, pages 13–20, 2013. pages 131, 149

Q. Bertrand and M. Massias. Anderson acceleration of coordinate descent. In *AISTATS*, 2021.

Q. Bertrand, M. Massias, A. Gramfort, and J. Salmon. Handling correlated and repeated measurements with the smoothed multivariate square-root Lasso. In *NeurIPS*, 2019.

Q. Bertrand, Q. Klopfenstein, M. Blondel, S. Vaiter, A. Gramfort, and J. Salmon. Implicit differentiation of Lasso-type models for hyperparameter optimization. *ICML*, 2020. page 138

Q. Bertrand, Q. Klopfenstein, M. Massias, M. Blondel, S. Vaiter, A. Gramfort, and J. Salmon. Implicit differentiation for fast hyperparameter selection in non-smooth convex learning. *arXiv preprint arXiv:2105.01637*, 2021.

D. P. Bertsekas. On the Goldstein-Levitin-Polyak gradient projection method. *IEEE Trans. Autom. Control*, 21(2):174–184, 1976. page 45

D. P. Bertsekas. *Convex optimization algorithms*. Athena Scientific Belmont, 2015. page 33

P. J. Bickel, Y. Ritov, and A. B. Tsybakov. Simultaneous analysis of Lasso and Dantzig selector. *Ann. Statist.*, 37(4):1705–1732, 2009. pages 22, 82, 83, 85, 103, 131

M. Blausen. Medical gallery of blausen medical 2014. *WikiJournal of Medicine*, 1(2): 1–79, 2014. page 18

M. Blondel and F. Pedregosa. Lightning: large-scale linear classification, regression and ranking in python, 2016. pages 27, 64, 142, 147, 148

M. Blondel, Q. Berthet, M.Cuturi, R. Frostig, S. Hoyer, F. Llinares-López, F. Pedregosa, and J-P. Vert. Efficient and modular implicit differentiation. *arXiv preprint arXiv:2105.15183*, 2021. page 38

T. Blu and F. Luisier. The sure-let approach to image denoising. *IEEE Trans. Image Process.*, 16(11):2778–2786, 2007. page 26

R. Bollapragada, D. Scieur, and A. d'Aspremont. Nonlinear acceleration of momentum and primal-dual algorithms. *arXiv preprint arXiv:1810.04539*, 2018. pages 66, 67, 68

W. M. Bolstad and J. M. Curran. *Introduction to Bayesian statistics*. John Wiley & Sons, 2016. page 24

J. Bolte and E. Pauwels. Conservative set valued fields, automatic differentiation, stochastic gradient methods and deep learning. *Mathematical Programming*, pages 1–33, 2020a. page 136

J. Bolte and E. Pauwels. A mathematical model for automatic differentiation in machine learning. *arXiv preprint arXiv:2006.02080*, 2020b. page 136

V. De Bortoli, A. Durmus, M. Pereyra, and A. F. Vidal. Maximum likelihood estimation of regularization parameters in high-dimensional inverse problems: An empirical bayesian approach. part ii: Theoretical analysis. *SIAM Journal on Imaging Sciences*, 13(4):1990–2028, 2020. page 25

B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992. pages 43, 59, 130

S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, 2004. page 114

J. Bracken and J. T. McGill. Mathematical programs with optimization problems in the constraints. *Operations Research*, 21(1):37–44, 1973. page 25

J. Bradbury, R. Frostig, P. Hawkins, M. James J., C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax. page 38

P. Breheny and J. Huang. Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. *Ann. Appl. Stat.*, 5(1): 232, 2011. page 27

C. Brezinski, M. Redivo-Zaglia, and Y. Saad. Shanks sequence transformations and anderson acceleration. *SIAM Review*, 60(3):646–669, 2018. page 64

E. Brochu, V. M. Cora, and N. De Freitas. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599*, 2010. page 131

P. Bühlmann and S. van de Geer. *Statistics for high-dimensional data*. Springer Series in Statistics. Springer, Heidelberg, 2011. Methods, theory and applications. page 82

F. Bunea, J. Lederer, and Y. She. The group square-root Lasso: Theoretical properties and fast algorithms. *IEEE Trans. Inf. Theory*, 60(2):1313–1325, 2014. pages 84, 86

J. V. Burke and J. J. Moré. On the identification of active constraints. *SIAM J. Numer. Anal.*, 25(5):1197–1211, 1988. page 45

C. Cai, A. Hashemi, M. Diwakar, S. Haufe, K. Sekihara, and S. S. Nagarajan. Robust estimation of noise for electromagnetic brain imaging with the champagne algorithm. *NeuroImage*, 225:117411, 2021. page 23

D. Calvetti and E. Somersalo. Hypermodels in the bayesian imaging framework. *Inverse Problems*, 24(3):034013, 2008. page 24

D. Calvetti, H. Hakula, S. Pursiainen, and E. Somersalo. Conditionally gaussian hypermodels for cerebral source localization. *SIAM Journal on Imaging Sciences*, 2(3): 879–909, 2009. page 24

E. J. Candès, M. B. Wakin, and S. P. Boyd. Enhancing sparsity by reweighted l_1 minimization. *J. Fourier Anal. Applicat.*, 14(5-6):877–905, 2008. page 21

W. Candler and R. Norton. *Multi-level programming and development policy*. The World Bank, 1977. page 25

A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *J. Math. Imaging Vis.*, 40(1):120–145, 2011. pages 82, 113

C.-C Chang and C.-J. Lin. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011. pages 23, 57

O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *Machine learning*, 46(1-3):131–159, 2002. page 134

S. Chen and A. Banerjee. Alternating estimation for structured high-dimensional multi-response models. In *NeurIPS*, pages 2838–2848, 2017. pages 104, 116

S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM J. Sci. Comput.*, 20(1):33–61, 1998. pages 43, 130

H. Cherkaoui, J. Sulam, and T. Moreau. Learning to solve TV regularised problems with unrolled algorithms. *NeurIPS*, 33, 2020. page 138

D. Cohen. Magnetoencephalography: evidence of magnetic fields produced by alpha-rhythm currents. *Science*, 161(3843):784–786, 1968. page 18

B. Colson, P. Marcotte, and G. Savard. An overview of bilevel optimization. *Annals of operations research*, 153(1):235–256, 2007. pages 25, 131

P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling & Simulation*, 4(4):1168–1200, 2005. pages 45, 74, 135

S. F. Cotter, B. D. Rao, K. Engan, and K. Kreutz-Delgado. Sparse solutions to linear inverse problems with multiple measurement vectors. *IEEE Trans. Signal Process.*, 53(7):2477–2488, 2005. page 24

M. Crouzeix. Bounds for analytical functions of matrices. *Integral Equations and Operator Theory*, 48(4):461–477, 2004. page 68

M. Crouzeix. Numerical range and functional calculus in hilbert space. *Journal of Functional Analysis*, 244(2):668–690, 2007. page 68

M. Crouzeix and C. Palencia. The numerical range is a (1+2)-spectral set. *SIAM Journal on Matrix Analysis and Applications*, 38(2):649–655, 2017. page 68

A. S. Dalalyan, M. Hebiri, K. Meziani, and J. Salmon. Learning heteroscedastic models by convex programming under group sparsity. In *ICML*, pages 379–387, 2013. page 104

J. Daye, J. Chen, and H. Li. High-dimensional heteroscedastic regression with an application to eQTL data analysis. *Biometrics*, 68(1):316–326, 2012. page 104

A. Defazio, F. Bach, and S. Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *NeurIPS*, pages 1646–1654, 2014. page 130

C.-A. Deledalle, S. Vaiter, J. Fadili, and G. Peyré. Stein Unbiased GrAdient estimator of the Risk (SUGAR) for multiple parameter selection. *SIAM J. Imaging Sci.*, 7(4): 2448–2487, 2014. pages 26, 27, 132, 135, 136, 140, 145

S. Dempe, V. Kalashnikov, G. A. Pérez-Valdés, and N. Kalashnykova. Bilevel programming problems. *Energy Systems. Springer, Berlin*, 2015. page 145

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the emalgorithm. *Journal of the Royal statistical Society*, 39(1):1–38, 1977. page 24

L. Devroye and T. Wagner. Distribution-free performance bounds for potential function rules. *IEEE Transactions on Information Theory*, 25(5):601–604, 1979. pages 25, 26, 131

S. Diamond and S. Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *J. Mach. Learn. Res.*, 17(83):1–5, 2016. pages 95, 146

J. Domke. Generic methods for optimization-based modeling. In *AISTATS*, volume 22, pages 318–326, 2012. page 135

D. L. Donoho and I. M. Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *J. Amer. Statist. Assoc.*, 90(432):1200–1224, 1995. page 26

A. Durmus and E. Moulines. Sampling from a strongly log-concave distribution with the unadjusted langevin algorithm. *hal preprint 01304430v1*, 2016. page 25

A. Durmus and E. Moulines. High-dimensional bayesian inference via the unadjusted langevin algorithm. *Bernoulli*, 25(4A):2854–2882, 2019. page 25

A. Durmus, E. Moulines, and M. Pereyra. Efficient bayesian computation by proximal markov chain monte carlo: when langevin meets moreau. *SIAM Journal on Imaging Sciences*, 11(1):473–506, 2018. page 25

R. P. Eddy. Extrapolating to the limit of a vector sequence. In *Information linkage between applied mathematics and industry*, pages 387–396. Elsevier, 1979. page 64

B. Efron. How biased is the apparent error rate of a prediction rule? *J. Amer. Statist. Assoc.*, 81(394):461–470, 1986. pages 26, 131

L. El Ghaoui, V. Viallon, and T. Rabbani. Safe feature elimination in sparse supervised learning. *J. Pacific Optim.*, 8(4):667–698, 2012. page 116

D.-A. Engemann and A. Gramfort. Automated model selection in covariance estimation and spatial whitening of MEG and EEG signals. *NeuroImage*, 108:328–342, 2015. pages 23, 104

L. C. Evans and R. F. Gariepy. *Measure theory and fine properties of functions*. CRC Press, 1992. page 133

V. Eyert. A comparative study on methods for convergence acceleration of iterative vector sequences. *Journal of Computational Physics*, 124(2):271–285, 1996. page 66

J. Fadili, J. Malick, and G. Peyré. Sensitivity analysis for mirror-stratifiable convex functions. *SIAM J. Optim.*, 28(4):2975–3000, 2018. page 46

J. Fadili, G. Garrigos, J. Malick, and G. Peyré. Model consistency for learning with mirror-stratifiable regularizers. In *AISTATS*, pages 1236–1244. PMLR, 2019. page 46

R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *J. Mach. Learn. Res.*, 9:1871–1874, 2008. pages 27, 64, 75

J. Fan and J. Lv. Sure independence screening for ultrahigh dimensional feature space. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 70(5):849–911, 2008. pages 115, 130

O. Fercoq and P. Richtárik. Accelerated, parallel and proximal coordinate descent. *SIAM J. Optim.*, 25(3):1997 – 2013, 2015. pages 31, 32, 44, 64, 74, 76

O. Fercoq, A. Gramfort, and J. Salmon. Mind the duality gap: safer rules for the lasso. In *ICML*, pages 333–342, 2015. page 116

M. Feurer and F. Hutter. Hyperparameter optimization. In *Automated Machine Learning*, pages 3–33. Springer, Cham, 2019. pages 25, 131

M. Feurer, J. N. van Rijn, A. Kadra, P. Gijsbers, N. Mallik, S. Ravi, A. Müller, J. Vanschoren, and F. Hutter. Openml-python: an extensible python api for openml. *arXiv:1911.02490*, 2019. page 75

M. Figueiredo. Adaptive sparseness using jeffreys prior. In *NeurIPS*, pages 697–704, 2001. page 24

C. S. Foo, C. B. Do, and A. Y. Ng. Efficient multiple hyperparameter learning for log-linear models. In *NeurIPS*, pages 377–384, 2008. page 134

A. Forrester, A. Sobester, and A. Keane. *Engineering design via surrogate modelling: a practical guide*. John Wiley & Sons, 2008. page 131

L. Franceschi, M. Donini, P. Frasconi, and M. Pontil. Forward and reverse gradient-based hyperparameter optimization. In *ICML*, pages 1165–1173, 2017. pages 132, 135

L. Franceschi, P. Frasconi, S. Salzo, and M. Pontil. Bilevel programming for hyperparameter optimization and meta-learning. In *ICML*, pages 1563–1572, 2018. pages 134, 135, 152

P.I. Frazier. A tutorial on Bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018. page 151

J. Frecon, S. Salzo, and M. Pontil. Bilevel learning of the group lasso structure. In *NeurIPS*, pages 8301–8311, 2018. pages 138, 145

J. Friedman, T. J. Hastie, H. Höfling, and R. Tibshirani. Pathwise coordinate optimization. *Ann. Appl. Stat.*, 1(2):302–332, 2007. pages 43, 44

J. Friedman, T. J. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008. page 117

J. Friedman, T. Hastie, and R. Tibshirani. glmnet: Lasso and elastic-net regularized generalized linear models. *R package version*, 1(4), 2009. pages 27, 64

J. Friedman, T. J. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *J. Stat. Softw.*, 33(1):1, 2010. pages 30, 43, 63, 136

K. Friston, L. Harrison, J. Daunizeau, S. Kiebel, C. Phillips, N. Trujillo-Barreto, R. Henson, G. Flandin, and J. Mattout. Multiple sparse priors for the m/eeg inverse problem. *NeuroImage*, 39(3):1104–1120, 2008. page 24

A. Fu, J. Zhang, and S. Boyd. Anderson accelerated Douglas-Rachford splitting. *arXiv preprint arXiv:1908.11482*, 2019. page 64

N. P. Katsaggelos A. Galatsanos. Methods for choosing the regularization parameter and estimating the noise variance in image restoration and their... *IEEE Trans. Image Process.*, 1(3), 1992. page 26

J. Ge, X. Li, H. Jiang, H. Liu, T. Zhang, M. Wang, and T. Zhao. Picasso: A sparse learning library for high dimensional data analysis in r and python. *J. Mach. Learn. Res.*, 20(1):1692–1696, 2019. page 27

S. Geisser. A predictive approach to the random effect model. *Biometrika*, 61(1): 101–107, 1974. page 26

A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin. *Bayesian data analysis*. CRC press, 2013. page 21

S. Ghadimi and M. Wang. Approximation methods for bilevel programming. *arXiv preprint arXiv:1802.02246*, 2018. page 135

C. Giraud. *Introduction to high-dimensional statistics*, volume 138. CRC Press, 2014. pages 99, 101

A. Gittens and J. A. Tropp. Tail bounds for all eigenvalues of a sum of random matrices. *arXiv preprint arXiv:1104.4513*, 2011. page 101

G. H. Golub and R. S. Varga. Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second order richardson iterative methods. *Numerische Mathematik*, 3(1):147–156, 1961. page 64

I. Goodfellow, A. Courville, and Y. Bengio. *Deep learning*, volume 1. MIT press Cambridge, 2016. page 135

S. Gould, B. Fernando, A. Cherian, P. Anderson, R. S. Cruz, and E. Guo. On differentiating parameterized argmin and argmax problems with application to bi-level optimization. *arXiv preprint arXiv:1607.05447.*, 2016. page 138

A. Gramfort, M. Kowalski, and M. S. Hämäläinen. Mixed-norm estimates for the m/eeg inverse problem using accelerated gradient methods. *Physics in Medicine & Biology*, 57(7):1937, 2012. page 21

A. Gramfort, D. Strohmeier, J. Haueisen, M. S. Hämäläinen, and M. Kowalski. Time-frequency mixed-norm estimates: Sparse M/EEG imaging with non-stationary source activations. *NeuroImage*, 70:410–422, 2013. page 104

A. Gramfort, M. Luessi, E. Larson, D. A. Engemann, D. Strohmeier, C. Brodbeck, L. Parkkonen, and M. S. Hämäläinen. MNE software for processing MEG and EEG data. *NeuroImage*, 86:446 – 460, 2014. pages 19, 120

E. Grave, G. Obozinski, and F. Bach. Trace lasso: a trace norm regularization for correlated designs. In *NeurIPS*, pages 2187–2195, 2011. page 92

R. Grazzi, L. Franceschi, M. Pontil, and S. Salzo. On the iteration complexity of hypergradient computation. *ICML*, 2020. page 134

M. Gurbuzbalaban, A. E. Ozdaglar, P. A. Parrilo, and N. D. Vanli. When cyclic coordinate descent outperforms randomized coordinate descent. *NeurIPS*, 2017. page 36

M. Hämäläinen, R. Hari, R. J. Ilmoniemi, J. Knuutila, and O. V. Lounasmaa. Magnetoencephalography—theory, instrumentation, and applications to noninvasive studies of the working human brain. *Reviews of modern Physics*, 65(2):413, 1993. page 19

M. S. Hämäläinen and R. J. Ilmoniemi. Interpreting magnetic fields of the brain: minimum norm estimates. *Medical & biological engineering & computing*, 32(1):35–42, 1994. pages 20, 21

W. L. Hare. Identifying active manifolds in regularization problems. In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, pages 261–271. Springer, 2011. page 45

W. L. Hare and A. S. Lewis. Identifying active constraints via partial smoothness and prox-regularity. *Journal of Convex Analysis*, 11(2):251–266, 2004. pages 45, 47, 48, 51, 61

W. L. Hare and A. S. Lewis. Identifying active manifolds. *Algorithmic Operations Research*, 2(2):75–75, 2007. pages 46, 138

C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. Fern'andez del R'ıo, M. Wiebe, P. Peterson, P. G'erard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020. page 145

T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu. The entire regularization path for the support vector machine. *J. Mach. Learn. Res.*, 5(Oct):1391–1415, 2004. page 136

L. Heller and D. B. van Hulsteyn. Brain stimulation using electromagnetic sources: theoretical aspects. *Biophysical journal*, 63(1):129–138, 1992. page 19

M. R. Hestenes and E. Stiefel. *Methods of conjugate gradients for solving linear systems*, volume 49. NBS Washington, DC, 1952. pages 66, 134

N. J. Higham. *Accuracy and stability of numerical algorithms*. SIAM, 2002. page 159

A. Hillebrand, K. D. Singh, I. E. Holliday, P. L. Furlong, and G. R. Barnes. A new approach to neuroimaging with magnetoencephalography. *Human brain mapping*, 25 (2):199–211, 2005. page 21

J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex analysis and minimization algorithms. II*, volume 306. Springer-Verlag, Berlin, 1993. page 89

A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970. page 130

M. Hong, X. Wang, M. Razaviyayn, and Z-Q. Luo. Iteration complexity analysis of block coordinate descent methods. *Mathematical Programming*, 163(1-2):85–114, 2017. page 45

D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of physiology*, 160(1):106–154, 1962. page 17

P. J. Huber. *Robust Statistics*. John Wiley & Sons Inc., 1981. page 103

P. J. Huber and R. Dutter. Numerical solution of robust regression problems. In *Compstat 1974 (Proc. Sympos. Computational Statist., Univ. Vienna, Vienna, 1974)*, pages 165–172. Physica Verlag, Vienna, 1974. pages 82, 103

J. D. Hunter. Matplotlib: A 2d graphics environment. *IEEE Annals of the History of Computing*, 9(03):90–95, 2007. page 145

F. Hutter, J. Lücke, and L. Schmidt-Thieme. Beyond manual tuning of hyperparameters. *KI-Künstliche Intelligenz*, 29(4):329–337, 2015. pages 25, 131

F. Iutzeler and J. Malick. Nonsmoothness in machine learning: specific structure, proximal identification, and applications. *Set-Valued and Variational Analysis*, 28(4): 661–678, 2020. page 45

H. Jasper and W. Penfield. Electrocorticograms in man: effect of voluntary movement upon the electrical activity of the precentral gyrus. *Archiv für Psychiatrie und Nervenkrankheiten*, 183(1):163–174, 1949. page 18

K. Ji, J. Yang, and Y. Liang. Provably faster algorithms for bilevel optimization and applications to meta-learning. *arXiv preprint arXiv:2010.07962*, 2020. pages 135, 145

R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *NeurIPS*, pages 315–323, 2013. page 130

T. B. Johnson and C. Guestrin. Blitz: A principled meta-algorithm for scaling sparse optimization. In *ICML*, pages 1171–1179, 2015. pages 116, 142

D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998. page 131

M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999. page 24

S. Kaczmarz. Angenaherte auflosung von systemen linearer gleichungen: Bulletin international de l'académie polonaise des sciences et des lettres. 1937. page 30

J. Kaipio and E. Somersalo. *Statistical and computational inverse problems*, volume 160. Springer Science & Business Media, 2006. page 24

D. P. Kingma and J. L. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. page 145

Q. Klopfenstein, **Q. Bertrand**, A. Gramfort, J. Salmon, and S. Vaiter. Model identification and local linear convergence of coordinate descent. *arXiv preprint arXiv:2010.11825*, 2020. pages 137, 144, 156

K. Koh, S.-J. Kim, and S. Boyd. An interior-point method for large-scale l1-regularized logistic regression. *J. Mach. Learn. Res.*, 8(8):1519–1555, 2007. pages 22, 130

R. Kohavi and G. H. John. Automatic parameter selection by minimizing estimated error. In *Machine Learning Proceedings 1995*, pages 304–312. Elsevier, 1995. pages 25, 131

M. Kolar and J. Sharpnack. Variance function estimation in high-dimensions. In *ICML*, pages 1447–1454, 2012. page 104

V. Koltchinskii, K. Lounici, and A. B. Tsybakov. Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion. *Ann. Statist.*, 39(5):2302–2329, 2011. page 94

G. M. Korpelevich. The extragradient method for finding saddle points and other problems. *Matecon*, 12:747–756, 1976. page 161

K. Kunisch and T. Pock. A bilevel optimization approach for parameter learning in variational models. *SIAM J. Imaging Sci.*, 6(2):938–983, 2013. page 135

S. K. Lam, A. Pitrou, and S. Seibert. Numba: A LLVM-based Python JIT Compiler. In *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, pages 1–6. ACM, 2015. pages 73, 116, 145, 147

J. Larsen, L. K. Hansen, C. Svarer, and M. Ohlsson. Design and regularization of neural networks: the optimal use of a validation set. In *Neural Networks for Signal Processing VI. Proceedings of the 1996 IEEE Signal Processing Society Workshop*, 1996. page 132

Y. A. LeCun, L. Bottou, G. B. Orr, and K-R. Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 1998. pages 132, 135

O. Ledoit and M. Wolf. A well-conditioned estimator for large-dimensional covariance matrices. *J. Multivariate Anal.*, 88(2):365–411, 2004. page 23

W. Lee and Y. Liu. Simultaneous multiple response regression and inverse covariance matrix estimation via penalized Gaussian maximum likelihood. *Journal of multivariate analysis*, 111:241–255, 2012. pages 104, 116

D. Leventhal and A. S. Lewis. Randomized methods for linear constraints: convergence rates and conditioning. *Mathematics of Operations Research*, 35(3):641–654, 2010. page 44

A. S. Lewis. Active sets, nonsmoothness, and sensitivity. *SIAM Journal on Optimization*, 13(3):702–725, 2002. pages 47, 50

X. Li, J. Haupt, R. Arora, H. Liu, M. Hong, and T. Zhao. On fast convergence of proximal algorithms for sqrt-lasso optimization: Don't worry about its nonsmooth loss function. *arXiv preprint arXiv:1605.07950*, 2016. page 86

X. Li, T. Zhao, R. Arora, H. Liu, and M. Hong. On faster convergence of cyclic block coordinate descent-type methods for strongly convex minimization. *Journal of Machine Learning Research*, 18(1):6741–6764, 2017. page 45

J. Liang, J. Fadili, and G. Peyré. Local linear convergence of forward–backward under partial smoothness. In *NeurIPS*, pages 1970–1978, 2014. pages 45, 51, 137, 144, 154, 155

J. Liang, J. Fadili, and G. Peyré. Activity Identification and Local Linear Convergence of Forward–Backward-type Methods. *SIAM J. Optim.*, 27(1):408–437, 2017. pages 45, 51, 138

F.-H. Lin, J. W. Belliveau, A. M. Dale, and M. S. Hämäläinen. Distributed current estimates using cortical orientation constraints. *Human brain mapping*, 27(1):1–13, 2006a. page 19

F.-H. Lin, T. Witzel, S. P. Ahlfors, S. M. Stufflebeam, J. W. Belliveau, and M. S. Hämäläinen. Assessing and improving the spatial accuracy in meg source localization by depth-weighted minimum-norm estimates. *Neuroimage*, 31(1):160–171, 2006b. page 21

Q. Lin, Z. Lu, and L. Xiao. An accelerated proximal coordinate gradient method. In *NeurIPS*, pages 3059–3067. Citeseer, 2014. pages 64, 74, 76

S. Linnainmaa. The representation of the cumulative rounding error of an algorithm as a Taylor expansion of the local rounding errors. *Master's Thesis (in Finnish), Univ. Helsinki*, pages 6–7, 1970. pages 132, 135

P-L. Lions and B. Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, 16(6):964–979, 1979. pages 45, 74, 134, 135

D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989. page 130

H. Liu, L. Wang, and T. Zhao. Calibrated multivariate regression with application to neural semantic basis discovery. *J. Mach. Learn. Res.*, 16:1579–1606, 2015. page 83

R. Liu, P. Mu, X. Yuan, S. Zeng, and J. Zhang. A generic first-order algorithmic framework for bi-level programming beyond lower-level singleton. *ICML*, 2020. page 145

N. K. Logothetis, J. Pauls, M. Augath, T. Trinath, and A. Oeltermann. Neurophysiological investigation of the basis of the fmri signal. *nature*, 412(6843):150–157, 2001. page 18

J. Lorraine, P. Vicol, and D. Duvenaud. Optimizing millions of hyperparameters by implicit differentiation. *arXiv preprint arXiv:1911.02590*, 2019. page 134

K. Lounici. Sup-norm convergence rate and sign concentration property of Lasso and Dantzig estimators. *Electron. J. Stat.*, 2:90–102, 2008. pages 22, 23, 83, 84, 85, 131

K. Lounici, M. Pontil, A. Tsybakov, and S. van de Geer. Taking Advantage of Sparsity in Multi-Task Learning. *arXiv preprint arXiv:0903.1468*, 2009. pages 22, 91, 95, 99, 100

K. Lounici, M. Pontil, S. van de Geer, and A. B. Tsybakov. Oracle inequalities and optimal inference under group sparsity. *Ann. Statist.*, 39(4):2164–2204, 2011. pages 83, 88

K. Lounici, K. Meziani, and B. Riu. Muddling labels for regularization, a novel approach to generalization. *arXiv preprint arXiv:2102.08769*, 2021. page 152

F. Lucka. Fast markov chain monte carlo sampling for sparse bayesian inference in high-dimensional inverse problems using l1-type priors. *Inverse Problems*, 28(12):125012, 2012. page 24

Z-Q. Luo and P. Tseng. On the convergence of the coordinate descent method for convex differentiable minimization. *Journal of Optimization Theory and Applications*, 72(1): 7–35, 1992. pages 31, 45

D. J. C. MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003. page 24

Matthew MacKay, Paul Vicol, Jon Lorraine, David Duvenaud, and Roger Grosse. Self-tuning networks: Bilevel optimization of hyperparameters using structured best-response functions. *arXiv preprint arXiv:1903.03088*, 2019. page 161

V. V. Mai and M. Johansson. Anderson acceleration of proximal gradient methods. *ICML*, 2019. pages 64, 74

J. Mairal and B. Yu. Complexity analysis of the lasso regularization path. In *ICML*, pages 353–360, 2012. page 136

J. Mairal, F. Bach, and J. Ponce. Task-driven dictionary learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(4):791–804, 2012. page 138

B. Malézieux, T. Moreau, and Matthieu M. Kowalski. Dictionary and prior learning with unrolled algorithms for unsupervised inverse problems. *arXiv preprint arXiv:2106.06338*, 2021. page 138

M. Massias, O. Fercoq, A. Gramfort, and J. Salmon. Generalized concomitant multi-task Lasso for sparse multimodal regression. In *AISTATS*, pages 998–1007, 2018a. pages 83, 92, 104, 106, 109, 114, 116

M. Massias, A. Gramfort, and J. Salmon. Celer: a fast solver for the Lasso with dual extrapolation. In *ICML*, pages 3321–3330, 2018b. pages 45, 116, 142

M. Massias, **Q. Bertrand**, A. Gramfort, and J. Salmon. Support recovery and sup-norm convergence rates for sparse pivotal estimation. In *AISTATS*, 2020a.

M. Massias, S. Vaiter, A. Gramfort, and J. Salmon. Dual extrapolation for sparse generalized linear models. *J. Mach. Learn. Res.*, 21(234):1–33, 2020b. pages 21, 46, 66, 130, 142, 147, 148

R. Mazumder, J. H. Friedman, and T. Hastie. Sparsenet: Coordinate descent with nonconvex penalties. *Journal of the American Statistical Association*, 106(495):1125–1138, 2011. page 27

W. McKinney. *Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython*. O'Reilly Media, 2012. page 73

S. Mehmood and P. Ochs. Differentiating the value function by using convex duality. In *AISTATS*, pages 3871–3879. PMLR, 2021. page 135

L. Mescheder, S. Nowozin, and A. Geiger. The numerics of GANs. *NeurIPS*, 2017. page 161

C. A. Micchelli, J. M. Morales, and M. Pontil. A family of penalty functions for structured sparsity. In *NeurIPS*, pages 1612–1623, 2010. page 82

J. Mockus. The bayesian approach to local optimization. In *Bayesian Approach to Global Optimization*, pages 125–156. Springer, 1989. page 131

R. Molina, A. K. Katsaggelos, and J. Mateos. Bayesian and regularization methods for hyperparameter estimation in image restoration. *IEEE transactions on image processing*, 8(2):231–246, 1999. page 24

A. J. Molstad. Insights and algorithms for the multivariate square-root lasso. *arXiv preprint arXiv:1909.05041*, 2019. pages 83, 86, 95, 113

J.-J. Moreau. Proximité et dualité dans un espace hilbertien. *Bull. Soc. Math. France*, 93:273–299, 1965. page 104

J. C. Mosher and R. M. Leahy. Recursive music: a framework for eeg and meg source localization. *IEEE Transactions on Biomedical Engineering*, 45(11):1342–1354, 1998. page 21

J. C. Mosher and R. M. Leahy. Source localization using recursively applied and projected (rap) music. *IEEE Transactions on signal processing*, 47(2):332–340, 1999. page 21

J. C. Mosher, S. Baillet, and R. M. Leahy. EEG source localization and imaging using multiple signal classification approaches. *Journal of Clinical Neurophysiology*, 16(3): 225–238, 1999. page 21

E. Ndiaye, O. Fercoq, A. Gramfort, and J. Salmon. Gap safe screening rules for sparse multi-task and multi-class models. In *NeurIPS*, pages 811–819, 2015. page 104

E. Ndiaye, O. Fercoq, A. Gramfort, V. Leclère, and J. Salmon. Efficient smoothed concomitant lasso estimation for high dimensional regression. *Journal of Physics: Conference Series*, 904(1):012006, 2017. pages 83, 109, 113

I. Necoara and A. Patrascu. A random coordinate descent algorithm for optimization problems with composite objective function and linear coupled constraints. *Computational Optimization and Applications*, 57(2):307–337, 2014. page 44

I. Necoara, Y. Nesterov, and F. Glineur. Linear convergence of first order methods for non-strongly convex optimization. *Mathematical Programming*, 175(1-2):69–107, 2019. page 61

Y. Nesterov. A method for solving a convex programming problem with rate of convergence $O(1/k^2)$. *Soviet Math. Doklady*, 269(3):543–547, 1983. page 63

Y. Nesterov. *Introductory lectures on convex optimization*, volume 87 of *Applied Optimization*. Kluwer Academic Publishers, Boston, MA, 2004. pages 63, 64, 131

Y. Nesterov. Smooth minimization of non-smooth functions. *Math. Program.*, 103(1): 127–152, 2005. pages 83, 104

Y. Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM J. Optim.*, 22(2):341–362, 2012. pages 29, 32, 36, 44, 64

A. Y. Ng. Feature selection, l1 vs. l2 regularization, and rotational invariance. In *ICML*, page 78, 2004. pages 27, 64

J. Nocedal and S. J. Wright. *Numerical optimization*. Springer Series in Operations Research and Financial Engineering. Springer, New York, second edition, 2006. page 145

P. L. Nunez and R. B. Silberstein. On the relationship of synaptic activity to macroscopic measurements: does co-registration of eeg with fmri make sense? *Brain topography*, 13(2):79–96, 2000. page 19

J. Nutini. *Greed is good: greedy optimization methods for large-scale structured problems.* PhD thesis, University of British Columbia, 2018. pages 45, 51, 137

J. Nutini, M. W. Schmidt, I. H. Laradji, M. P. Friedlander, and H. A. Koepke. Coordinate descent converges faster with the Gauss-Southwell rule than random selection. In *ICML*, pages 1632–1641, 2015. pages 29, 44

J. Nutini, I. Laradji, and M. Schmidt. Let's make block coordinate descent go fast: Faster greedy rules, message-passing, active-set complexity, and superlinear convergence. *arXiv preprint arXiv:1712.08859*, 2017. pages 45, 51

J. Nutini, M. Schmidt, and W. Hare. "active-set complexity" of proximal gradient: How long does it take to find the sparsity pattern? *Optimization Letters*, 13(4):645–655, 2019. pages 45, 47, 51, 60, 137

G. Obozinski, B. Taskar, and M. I. Jordan. Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing*, 20(2):231–252, 2010. pages 104, 117

G. Obozinski, M. J. Wainwright, and M. Jordan. Support union recovery in high-dimensional multivariate regression. *Ann. Statist.*, 39(1):1–47, 2011. pages 21, 22, 27

P. Ochs, R. Ranftl, T. Brox, and T. Pock. Bilevel optimization with nonsmooth lower level problems. In *SSVM*, pages 654–665, 2015. page 138

B. O'Donoghue, E. Chu, N. Parikh, and S. Boyd. Conic optimization via operator splitting and homogeneous self-dual embedding. *Journal of Optimization Theory and Applications*, 169(3):1042–1068, June 2016. pages 95, 147

B. O'Donoghue, E. Chu, N. Parikh, and S. Boyd. SCS: Splitting conic solver, version 2.1.2, 2019. page 147

S. Ogawa, T.-M. Lee, A. R. Kay, and D. W. Tank. Brain magnetic resonance imaging with contrast dependent on blood oxygenation. *proceedings of the National Academy of Sciences*, 87(24):9868–9872, 1990. page 18

W. Ou, M. Hämaläinen, and P. Golland. A distributed spatio-temporal EEG/MEG inverse solver. *NeuroImage*, 44(3):932–946, Feb 2009. pages 21, 104, 120

A. B. Owen. A robust hybrid of lasso and ridge regression. *Cont. Math.*, 443:59–72, 2007. pages 82, 103, 109

N. Parikh, S. Boyd, E. Chu, B. Peleato, and J. Eckstein. Proximal algorithms. *Foundations and Trends in Machine Learning*, 1(3):1–108, 2013. page 107

F. Pedregosa. Hyperparameter optimization with approximate gradient. In *ICML*, pages 737–746, 2016. pages 134, 135, 143, 145, 157, 161

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.*, 12:2825–2830, 2011. pages 27, 44, 64

F. Pedregosa, R. Leblond, and S. Lacoste-Julien. Breaking the nonsmooth barrier: A scalable parallel method for composite optimization. *NeurIPS*, pages 56–65, 2017. page 130

W. Penfield and H. Jasper. Epilepsy and the functional anatomy of the human brain. 1954. page 17

W. Penfield and T. Rasmussen. The cerebral cortex of man; a clinical study of localization of function. 1950. page 17

M. Pereyra, J. M. Bioucas-Dias, and M. Figueiredo. Maximum-a-posteriori estimation with unknown regularisation parameters. In *2015 23rd European Signal Processing Conference (EUSIPCO)*, pages 230–234. IEEE, 2015. page 24

J-C Pesquet, A. Benazza-Benyahia, and C. Chaux. A sure approach for digital signal/image deconvolution problems. *IEEE Transactions on Signal Processing*, 57(12): 4616–4632, 2009. page 26

G. Peyré and J. M. Fadili. Learning analysis sparsity priors. In *Sampta*, 2011. page 138

J. C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *ADVANCES IN LARGE MARGIN CLASSIFIERS*, pages 61–74. MIT Press, 1999. page 130

R. Plonsey and D. B. Heppner. Considerations of quasi-stationarity in electrophysiological systems. *The Bulletin of mathematical biophysics*, 29(4):657–664, 1967. page 19

R. Poliquin and R. Rockafellar. Prox-regular functions in variational analysis. *Transactions of the American Mathematical Society*, 348(5):1805–1838, 1996a. page 51

R. A. Poliquin and R. T. Rockafellar. Generalized hessian properties of regularized nonsmooth functions. *SIAM Journal on Optimization*, 6(4):1121–1137, 1996b. page 136

B. T. Polyak. Introduction to optimization. optimization software. *Inc., Publications Division, New York*, 1, 1987. pages 28, 35, 53, 56, 153

C. Poon and J. Liang. Trajectory of alternating direction method of multipliers and adaptive acceleration. In *NeurIPS*, pages 7357–7365, 2019. pages 45, 51, 64

C. Poon and J. Liang. Geometry of first-order methods and adaptive acceleration. *arXiv preprint arXiv:2003.03910*, 2020. pages 64, 66, 74

C. Poon, J. Liang, and C.-B. Schönlieb. Local convergence properties of SAGA/Prox-SVRG and acceleration. In *ICML*, volume 90, pages 4121–4129, 2018. page 45

P. Pulay. Convergence acceleration of iterative sequences. the case of scf iteration. *Chemical Physics Letters*, 73(2):393–398, 1980. page 66

Z. Qu and P. Richtárik. Coordinate descent with arbitrary sampling i: Algorithms and complexity. *Optimization Methods and Software*, 31(5):829–857, 2016a. page 44

Z. Qu and P. Richtárik. Coordinate descent with arbitrary sampling ii: Expected separable overapproximation. *Optimization Methods and Software*, 31(5):858–884, 2016b. page 44

P. Rai, A. Kumar, and H. Daume. Simultaneously leveraging output and task structures for multiple-output regression. In *NeurIPS*, pages 3185–3193, 2012. page 104

A. Rajeswaran, C. Finn, S. M. Kakade, and S. Levine. Meta-learning with implicit gradients. In *NeurIPS*, pages 113–124, 2019. page 135

A. Rakhlin and K. Sridharan. Optimization, learning, and games with predictable sequences. *arXiv preprint arXiv:1311.1869*, 2013. page 161

S. Ramani, T. Blu, and M. Unser. Monte-carlo sure: A black-box optimization of regularization parameters for general denoising algorithms. *IEEE Trans. Image Process.*, 17(9):1540–1554, 2008. page 26

L. A. Rastrigin. The convergence of the random search method in the extremal control of a many parameter system. *Automaton & Remote Control*, 24:1337–1342, 1963. page 131

M. Razaviyayn, M. Hong, and Z.-QLuo. A unified convergence analysis of block successive minimization methods for nonsmooth optimization. *SIAM J. Optim.*, 23(2): 1126–1153, 2013. pages 31, 45

P. Richtárik and M. Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1-2): 1–38, 2014. pages 32, 36, 44, 74

P. Richtárik and M. Takáč. Parallel coordinate descent methods for big data optimization. *Mathematical Programming*, 156(1-2):433–484, 2016. page 44

H. Robbins and S. Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951. page 30

S. Rosset and J. Zhu. Piecewise linear regularized solution paths. *Ann. Statist.*, 35(3): 1012–1030, 2007. page 136

A. J. Rothman, E. Levina, and J. Zhu. Sparse multivariate regression with covariance estimation. *Journal of Computational and Graphical Statistics*, 19(4):947–962, 2010. pages 104, 117

A. Saha and A. Tewari. On the nonasymptotic convergence of cyclic coordinate descent methods. *SIAM J. Optim.*, 23(1):576–601, 2013. page 45

S. Salzo and S. Villa. Parallel random block-coordinate forward–backward algorithm: a unified convergence analysis. *Mathematical Programming*, pages 1–45, 2021. page 44

M. Sato, T. Yoshioka, S. Kajihara, K. Toyama, N. Goda, K. Doya, and M. Kawato. Hierarchical bayesian estimation for meg inverse problem. *NeuroImage*, 23(3):806–826, 2004. page 24

M. Scherg. Fundamentals of dipole source potential analysis. *Auditory evoked magnetic fields and electric potentials. Advances in audiology*, 6:40–69, 1990. page 21

M. Scherg and D. Von Cramon. Two bilateral sources of the late aep as identified by a spatio-temporal dipole model. *Electroencephalography and Clinical Neurophysiology/Evoked Potentials Section*, 62(1):32–44, 1985. page 21

G. Schwarz. Estimating the dimension of a model. *Ann. Statist.*, 6(2):461–464, 1978. pages 25, 131

D. Scieur. Generalized framework for nonlinear acceleration. *arXiv preprint arXiv:1903.08764*, 2019. page 66

D. Scieur, A. d'Aspremont, and F. Bach. Regularized nonlinear acceleration. In *Advances In Neural Information Processing Systems*, pages 712–720, 2016. pages 64, 66, 73

M. W. Seeger. Cross-validation optimization for large scale structured classification kernel methods. *J. Mach. Learn. Res.*, 9:1147–1178, 2008. page 134

M. W. Seeger and D. P. Wipf. Variational bayesian inference techniques. *IEEE Signal Processing Magazine*, 27(6):81–91, 2010. page 24

K. Sekihara, S. S. Nagarajan, D. Poeppel, and A. Marantz. Performance of an meg adaptive-beamformer technique in the presence of correlated neural activities: effects on signal intensity and time-course estimates. *IEEE Transactions on Biomedical Engineering*, 49(12):1534–1546, 2002. page 21

S. Shalev-Shwartz and A. Tewari. Stochastic methods for l1-regularized loss minimization. *Journal of Machine Learning Research*, 12:1865–1892, 2011. pages 32, 44

S. Shalev-Shwartz and T. Zhang. Accelerated proximal stochastic dual coordinate ascent for regularized loss minimization. *arXiv preprint arXiv:1309.2375*, 2013a. pages 27, 63

S. Shalev-Shwartz and T. Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 14(Feb):567–599, 2013b. pages 32, 44

J. Shao. Linear model selection by cross-validation. *J. Amer. Statist. Assoc.*, 88(422): 486–494, 1993. page 26

J. Shao. An asymptotic theory for linear model selection. *Statistica sinica*, pages 221–242, 1997. page 26

J. She and M. Schmidt. Linear convergence and support vector identification of sequential minimal optimization. In *10th NIPS Workshop on Optimization for Machine Learning*, volume 5, 2017. page 46

X. Shen and J. Ye. Adaptive model selection. *J. Amer. Statist. Assoc.*, 97(457):210–221, 2002. page 26

H.-J. M. Shi, S. Tu, Y. Xu, and W. Yin. A primer on coordinate descent algorithms. *ArXiv e-prints*, 2016. pages 28, 44, 63

A. Sidi. *Vector extrapolation methods with applications*. SIAM, 2017. page 64

N. Simon, J. Friedman, T. J. Hastie, and R. Tibshirani. A sparse-group lasso. *J. Comput. Graph. Statist.*, 22(2):231–245, 2013. ISSN 1061-8600. pages 22, 27

A. Sinha, P. Malo, and K. Deb. A review on bilevel optimization: from classical to evolutionary approaches and applications. *IEEE Transactions on Evolutionary Computation*, 22(2):276–295, 2017. page 161

D. A. Smith, W. F. Ford, and A. Sidi. Extrapolation methods for vector sequences. *SIAM review*, 29(2):199–233, 1987. page 64

J. Snoek, H. Larochelle, and R. P. Adams. Practical bayesian optimization of machine learning algorithms. In *NeurIPS*, pages 2960–2968, 2012. page 131

R. V. Southwell. *Relaxation methods in engineering science-a treatise on approximate computation*. Oxford University Press, 1940. page 29

C. M. Stein. Estimation of the mean of a multivariate normal distribution. *Ann. Statist.*, 9(6):1135–1151, 1981. pages 25, 26, 131

L. R. A. Stone and J.C. Ramer. Estimating WAIS IQ from Shipley Scale scores: Another cross-validation. *Journal of clinical psychology*, 21(3):297–297, 1965. pages 25, 131

M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the royal statistical society: Series B (Methodological)*, 36(2):111–133, 1974. page 26

D. Strohmeier, J. Haueisen, and A. Gramfort. Improved meg/eeg source localization with reweighted mixed-norms. In *2014 International Workshop on Pattern Recognition in Neuroimaging*, pages 1–4. IEEE, 2014. page 21

T. Strohmer and R. Vershynin. A randomized kaczmarz algorithm with exponential convergence. *Journal of Fourier Analysis and Applications*, 15(2):262–278, 2009. page 30

R. Sun and M. Hong. Improved iteration complexity bounds of cyclic block coordinate descent for convex problems. In *Advances in Neural Information Processing Systems*, pages 1306–1314, 2015. page 45

R. Sun and Y. Ye. Worst-case complexity of cyclic coordinate descent: $O(n^2)$ gap with randomized version. *Mathematical Programming*, pages 1–34, 2019. pages 35, 36, 64

T. Sun and C.-H. Zhang. Scaled sparse linear regression. *Biometrika*, 99(4):879–898, 2012. pages 83, 103

Y. Sun, H. Jeong, J. Nutini, and M. Schmidt. Are we there yet? manifold identification of gradient-related proximal methods. In *AISTATS*, volume 89, pages 1110–1119, 2019. page 60

S. Tao, D. Boley, and S. Zhang. Local linear convergence of ISTA and FISTA on the LASSO problem. *SIAM J. Optim.*, 26(1):313–336, 2016. page 51

R. Tibshirani. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 58(1):267–288, 1996. pages 21, 22, 27, 43, 45, 58, 63, 82, 103, 130

R. Tibshirani, J. Bien, J. Friedman, T. J. Hastie, N. Simon, J. Taylor, and R. J. Tibshirani. Strong rules for discarding predictors in lasso-type problems. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 74(2):245–266, 2012. pages 115, 130

R. J. Tibshirani. The lasso problem and uniqueness. *Electron. J. Stat.*, 7:1456–1490, 2013. page 145

M. E. Tipping. Sparse bayesian learning and the relevance vector machine. *Journal of machine learning research*, 1(Jun):211–244, 2001. page 24

J. H. Tripp. Physical concepts and mathematical models. In *Biomagnetism*, pages 101–139. Springer, 1983. page 19

P. Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *J. Optim. Theory Appl.*, 109(3):475–494, 2001. pages 29, 31, 45, 114

P. Tseng and S. Yun. Block-coordinate gradient descent method for linearly constrained nonsmooth separable optimization. *J. Optim. Theory Appl.*, 140(3):513, 2009a. pages 21, 27, 29, 31, 44, 45, 114, 115, 130, 134

P. Tseng and S. Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, 117(1-2):387–423, 2009b. pages 27, 63, 74

K. Uutela, M. Hamalainen, and R. Salmelin. Global optimization in the localization of neuromagnetic sources. *IEEE Transactions on Biomedical Engineering*, 45(6):716–723, 1998. page 21

S. Vaiter, C-A Deledalle, G. Peyré, C. Dossal, and J. Fadili. Local behavior of sparse analysis regularization: Applications to risk estimation. *Applied and Computational Harmonic Analysis*, 35(3):433–451, 2013. page 26

S. Vaiter, M. Golbabaee, J. Fadili, and G. Peyré. Model selection with low complexity priors. *Information and Inference: A Journal of the IMA*, 4(3):230–287, 2015. page 47

S. Vaiter, G. Peyré, and J. M. Fadili. Model consistency of partly smooth regularizers. *IEEE Trans. Inf. Theory*, 64(3):1725–1737, 2018. pages 45, 47, 48, 137

S. van de Geer. *Estimation and testing under sparsity*, volume 2159 of *Lecture Notes in Mathematics*. Springer, 2016. Lecture notes from the 45th Probability Summer School held in Saint-Four, 2015, École d'Été de Probabilités de Saint-Flour. pages 23, 83, 89, 92, 109, 110

S. van de Geer and B. Stucky. χ 2-confidence sets in high-dimensional regression. In *Statistical analysis for high-dimensional data*, pages 279–306. Springer, 2016. pages 83, 84, 95, 104, 109, 113

P. Vicol, L. Metz, and J. Sohl-Dickstein. Unbiased gradient estimation in unrolled computation graphs with persistent evolution strategies. In *ICML*, pages 10553–10563. PMLR, 2021. page 161

A. F. Vidal, V. De Bortoli, M. Pereyra, and A. Durmus. Maximum likelihood estimation of regularization parameters in high-dimensional inverse problems: An empirical bayesian approach part i: Methodology and experiments. *SIAM Journal on Imaging Sciences*, 13(4):1945–1989, 2020. page 25

P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020. page 145

H. von Stackelberg. *Marktform und Gleichgewicht*. J. Springer, 1934. page 161

C. Vonesch, S. Ramani, and M. Unser. Recursive risk estimation for non-linear image deconvolution with a wavelet-domain sparsity constraint. In *2008 15th IEEE International Conference on Image Processing*, pages 665–668. IEEE, 2008. page 26

J. Wagener and H. Dette. Bridge estimators and the adaptive Lasso under heteroscedasticity. *Math. Methods Statist.*, 21:109–126, 2012. page 104

R. E. Wengert. A simple automatic derivative evaluation program. *Communications of the ACM*, 7(8):463–464, 1964. pages 132, 135

S. J. Williamson, G.-L Romani, L. Kaufman, and I. Modena. *Biomagnetism: an interdisciplinary approach*, volume 66. Springer Science & Business Media, 2013. page 19

E. Winston and Z. Kolter. Neural monotone operator equilibrium networks. *NeurIPS*, 2020. page 138

D. Wipf and S. Nagarajan. A unified Bayesian framework for MEG/EEG source imaging. *NeuroImage*, 44(3):947–966, 2009. page 24

D. P. Wipf and B. D. Rao. Sparse bayesian learning for basis selection. *IEEE Transactions on Signal processing*, 52(8):2153–2164, 2004. page 24

S. J. Wright. Identifiable surfaces in constrained optimization. *SIAM Journal on Control and Optimization*, 31(4):1063–1079, 1993. page 45

S. J. Wright. Accelerated block-coordinate relaxation for regularized optimization. *SIAM J. Optim.*, 22(1):159–186, 2012. page 46

S. J. Wright. Coordinate descent algorithms. *Mathematical Programming*, 151(1):3–34, 2015. pages 21, 28, 63

P. Wynn. Acceleration techniques for iterated vector and matrix problems. *Mathematics of Computation*, 16(79):301–322, 1962. page 64

Y. Xu and W. Yin. A globally convergent algorithm for nonconvex optimization based on block coordinate update. *Journal of Scientific Computing*, 72(2):700–734, 2017. page 45

J. Ye. On measuring and correcting the effects of data mining and model selection. *J. Amer. Statist. Assoc.*, 93(441):120–131, 1998. page 26

J. Zhang, B. O'Donoghue, and S. Boyd. Globally convergent type-I Anderson acceleration for non-smooth fixed-point iterations. *arXiv preprint arXiv:1808.03971*, 2018. page 64

L. Zhang, M. Mahdavi, and R. Jin. Linear convergence with condition number independent access of full gradients. *NeurIPS*, 26:980–988, 2013. page 130

P. Zhang. Model selection via multifold cross validation. *The annals of statistics*, pages 299–313, 1993. page 26

T. Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *ICML*, page 116, 2004. page 44

X.-P. Zhang and M. D. Desai. Adaptive denoising based on sure risk. *IEEE signal processing letters*, 5(10):265–267, 1998. page 26

Z. Zhang and B. D. Rao. Sparse signal recovery with temporally correlated source vectors using sparse bayesian learning. *IEEE Journal of Selected Topics in Signal Processing*, 5(5):912–926, 2011. page 24

P. Zhao and B. Yu. On model selection consistency of lasso. *J. Mach. Learn. Res.*, 7: 2541–2563, 2006. page 46

H. Zou. The adaptive lasso and its oracle properties. *J. Amer. Statist. Assoc.*, 101(476): 1418–1429, 2006. page 27

H. Zou and T. J. Hastie. Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 67(2):301–320, 2005. pages 22, 27, 43, 64, 76, 130

université PARIS-SACLAY
ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre : Sélection d'hyperparamètres pour l'apprentissage parcimonieux en grande dimension : application à la neuroimagerie

Mots clés : Selection d'hyperparamètres, optimisation d'hyperparamètres, optimisation bi-niveaux, parcimonie, descente par coordonnée, problèmes inverses, neuroimagerie

Résumé : Grâce à leur caractère non invasif et leur excellente résolution temporelle, la magnéto- et l'électroencéphalographie (M/EEG) sont devenues des outils incontournables pour observer l'activité cérébrale. La reconstruction des signaux cérébraux à partir des enregistrements M/EEG peut être vue comme un problème inverse de grande dimension mal posé. Les estimateurs typiques des signaux cérébraux se basent sur des problèmes d'optimisation difficiles à résoudre, composés de la somme d'un terme d'attache aux données et d'un terme favorisant la parcimonie. À cause du paramètre de régularisation notoirement difficile à calibrer, les estimateurs basés sur la parcimonie ne sont actuellement pas massivement utilisés par les praticiens. L'objectif de cette thèse est de fournir un moyen simple, rapide et automatisé de calibrer des modèles linéaires parcimonieux.
Nous étudions d'abord quelques propriétés de la descente par coordonnées : identification du modèle, convergence linéaire locale, et accélération. En nous appuyant sur les schémas d'extrapolation d'Anderson, nous proposons un moyen efficace d'accélérer la descente par coordonnées en théorie et en pratique.
Nous explorons ensuite une approche statistique pour calibrer le paramètre de régularisation des problèmes de type Lasso. Il est possible de construire des estimateurs pour lesquels le paramètre de régularisation optimal ne dépend pas du niveau de bruit. Cependant, ces estimateurs nécessitent de résoudre des problèmes d'optimisation "non lisses + non lisses". Nous montrons que le lissage partiel préserve leurs propriétés statistiques et nous proposons une application aux problèmes de localisation de sources M/EEG.
Enfin, nous étudions l'optimisation d'hyperparamètres, qui comprend notamment la validation croisée. Cela nécessite de résoudre des problèmes d'optimisation à deux niveaux avec des problèmes internes non lisses. De tels problèmes sont résolus de manière usuelle via des techniques d'ordre zéro, telles que la recherche sur grille ou la recherche aléatoire. Nous présentons une technique efficace pour résoudre ces problèmes d'optimisation à deux niveaux en utilisant des méthodes du premier ordre.

Title : Hyperparameter selection for high dimensional sparse learning: application to neuro-imaging

Keywords : Hyperparameter selection, hyperparameter optimization, convex optimization, bilevel optimization, sparsity, coordinate descent, inverse problem, neuro-imaging

Abstract : Due to non-invasiveness and excellent time resolution, magneto- and electroencephalography (M/EEG) have emerged as tools of choice to monitor brain activity. Reconstructing brain signals from M/EEG measurements can be cast as a high dimensional ill-posed inverse problem. Typical estimators of brain signals involve challenging optimization problems, composed of the sum of a data-fidelity term, and a sparsity promoting term. Because of their notoriously hard to tune regularization hyperparameters, sparsity-based estimators are currently not massively used by practitioners. The goal of this thesis is to provide a simple, fast, and automatic way to calibrate sparse linear models.
We first study some properties of coordinate descent: model identification, local linear convergence, and acceleration. Relying on Anderson extrapolation schemes, we propose an effective way to speed up coordinate descent in theory and practice.
We then explore a statistical approach to set the regularization parameter of Lasso-type problems. A closed-form formula can be derived for the optimal regularization parameter of L1 penalized linear regressions. Unfortunately, it relies on the true noise level, unknown in practice. To remove this dependency, one can resort to estimators for which the regularization parameter does not depend on the noise level. However, they require to solve challenging "nonsmooth + nonsmooth" optimization problems. We show that partial smoothing preserves their statistical properties and we propose an application to M/EEG source localization problems.
Finally we investigate hyperparameter optimization, encompassing held-out or cross-validation hyperparameter selection. It requires tackling bilevel optimization with nonsmooth inner problems. Such problems are canonically solved using zeros order techniques, such as grid-search or random-search. We present an efficient technique to solve these challenging bilevel optimization problems using first-order methods.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France