Foundations and Trends® in
Machine Learning
Vol. 4, No. 1 (2011) 1–106

DOI: 10.1561/2200000015

now
the essence of knowledge

# Optimization with Sparsity-Inducing Penalties

By Francis Bach, Rodolphe Jenatton,
Julien Mairal and Guillaume Obozinski

## Contents

Foundations and Trends® in
Machine Learning
Vol. 4, No. 1 (2011) 1–106

DOI: 10.1561/2200000015

now
the essence of knowledge

# Optimization with Sparsity-Inducing Penalties

**Francis Bach[1], Rodolphe Jenatton[2], Julien Mairal[3] and Guillaume Obozinski[4]**

[1] *INRIA — SIERRA Project-Team, Laboratoire d'Informatique de l'Ecole Normale Supérieure, 23, avenue d'Italie, Paris, 75013, France, francis.bach@inria.fr*
[2] *INRIA — SIERRA Project-Team, rodolphe.jenatton@inria.fr*
[3] *Department of Statistics, University of California, Berkeley, CA 94720-1776, USA, julien@stat.berkeley.edu*
[4] *INRIA — SIERRA Project-Team, guillaume.obozinski@inria.fr*

## Abstract

Sparse estimation methods are aimed at using or obtaining parsimonious representations of data or models. They were first dedicated to linear variable selection but numerous extensions have now emerged such as structured sparsity or kernel selection. It turns out that many of the related estimation problems can be cast as convex optimization problems by regularizing the empirical risk with appropriate nonsmooth norms. The goal of this monograph is to present from a general perspective optimization tools and techniques dedicated to such sparsity-inducing penalties. We cover proximal methods, block-coordinate descent, reweighted $\ell_2$-penalized techniques, working-set and homotopy methods, as well as non-convex formulations and extensions, and provide an extensive set of experiments to compare various algorithms from a computational point of view.

# 1

# Introduction

The principle of parsimony is central to many areas of science: the simplest explanation of a given phenomenon should be preferred over more complicated ones. In the context of machine learning, it takes the form of variable or feature selection, and it is commonly used in two situations. First, to make the model or the prediction more interpretable or computationally cheaper to use, i.e., even if the underlying problem is not sparse, one looks for the best sparse approximation. Second, sparsity can also be used given prior knowledge that the model should be sparse.

For variable selection in linear models, parsimony may be directly achieved by penalization of the empirical risk or the log-likelihood by the cardinality of the support[1] of the weight vector. However, this leads to hard combinatorial problems (see, e.g., [96, 136]). A traditional convex approximation of the problem is to replace the cardinality of the support by the $\ell_1$-norm. Estimators may then be obtained as solutions of convex programs.

Casting sparse estimation as convex optimization problems has two main benefits: First, it leads to efficient estimation algorithms — and

[1] We call the set of non-zeros entries of a vector the support.

this monograph focuses primarily on these. Second, it allows a fruitful theoretical analysis answering fundamental questions related to estimation consistency, prediction efficiency [19, 99] or model consistency [145, 158]. In particular, when the sparse model is assumed to be well-specified, regularization by the $\ell_1$-norm is adapted to high-dimensional problems, where the number of variables to learn from may be exponential in the number of observations.

Reducing parsimony to finding the model of lowest cardinality turns out to be limiting, and *structured parsimony* [15, 62, 64, 66] has emerged as a natural extension, with applications to computer vision [32, 62, 70], text processing [68], bioinformatics [64, 73] or audio processing [80]. Structured sparsity may be achieved by penalizing other functions than the cardinality of the support or regularizing by other norms than the $\ell_1$-norm. In this monograph, we focus not only on norms which can be written as linear combinations of norms on subsets of variables, but we also consider traditional extensions such as multiple kernel learning and spectral norms on matrices (see Sections 1.3 and 1.5). One main objective of this monograph is to present methods which are adapted to most sparsity-inducing norms with loss functions potentially beyond least-squares.

Finally, similar tools are used in other communities such as signal processing. While the objectives and the problem set-ups are different, the resulting convex optimization problems are often similar, and most of the techniques reviewed in this monograph also apply to sparse estimation problems in signal processing. Moreover, we consider in Section 7 non-convex formulations and extensions.

This monograph aims at providing a general overview of the main optimization techniques that have emerged as most relevant and efficient for methods of variable selection based on sparsity-inducing norms. We survey and compare several algorithmic approaches as they apply not only to the $\ell_1$-norm, group norms, but also to norms inducing structured sparsity and to general multiple kernel learning problems. We complement these by a presentation of some greedy and nonconvex methods. Our presentation is essentially based on existing literature, but the process of constructing a general framework leads naturally to new results, connections and points of view.

This monograph is organized as follows:

Sections 1.1 and 1.2 introduce respectively the notations used throughout the monograph and the optimization problem (1.1) which is central to the learning framework that we will consider.

Section 1.3 gives an overview of common sparsity and structured sparsity-inducing norms, with some of their properties and examples of structures which they can encode.

Section 1.4 provides an essentially self-contained presentation of concepts and tools from convex analysis that will be needed in the rest of the monograph, and which are relevant to understand algorithms for solving the main optimization problem (1.1). Specifically, since sparsity-inducing norms are nondifferentiable convex functions,[2] we introduce relevant elements of subgradient theory and Fenchel duality — which are particularly well suited to formulate the optimality conditions associated to learning problems regularized with these norms. We also introduce a general quadratic variational formulation for a certain class of norms in Section 1.4.2; the part on subquadratic norms is essentially relevant in view of sections on structured multiple kernel learning and can safely be skipped in a first reading.

Section 1.5 introduces *multiple kernel learning* (MKL) and shows that it can be interpreted as an extension of plain sparsity to reproducing kernel Hilbert spaces (RKHS), but formulated in the dual. This connection is further exploited in Section 1.5.2, where it is shown how structured counterparts of MKL can be associated with structured sparsity-inducing norms. These sections rely on Section 1.4.2. All sections on MKL can be skipped in a first reading.

In Section 2, we discuss classical approaches to solving the optimization problem arising from simple sparsity-inducing norms, such as interior point methods and subgradient descent, and point at their shortcomings in the context of machine learning.

Section 3 is devoted to a simple presentation of proximal methods. After two short sections introducing the main concepts and algorithms, the longer Section 3.3 focusses on the *proximal operator* and presents

[2] Throughout this monograph, we refer to sparsity-inducing norms such as the $\ell_1$-norm as nonsmooth norms; note that all norms are nondifferentiable at zero, but some norms have more nondifferentiability points (see more details in Section 1.3).

algorithms to compute it for a variety of norms. Section 3.4 shows how proximal methods for structured norms extend naturally to the RKHS/MKL setting.

Section 4 presents block coordinate descent algorithms, which provide an efficient alternative to proximal method for *separable* norms like the $\ell_1$- and $\ell_1/\ell_2$-norms, and can be applied to MKL. This section uses the concept of proximal operator introduced in Section 3.

Section 5 presents reweighted-$\ell_2$ algorithms that are based on the quadratic variational formulations introduced in Section 1.4.2. These algorithms are particularly relevant for the least-squares loss, for which they take the form of iterative reweighted least-squares algorithms (IRLS). Section 5.2 presents a generally applicable quadratic variational formulation for general norms that extends the variational formulation of Section 1.4.2.

Section 6 covers algorithmic schemes that take advantage computationally of the sparsity of the solution by extending the support of the solution gradually. These schemes are particularly relevant to construct approximate or exact regularization paths of solutions for a range of values of the regularization parameter. Specifically, Section 6.1 presents working-set techniques, which are meta-algorithms that can be used with the optimization schemes presented in all the previous sections. Section 6.2 focuses on the homotopy algorithm, which can efficiently construct the entire regularization path of the Lasso.

Section 7 presents nonconvex as well as Bayesian approaches that provide alternatives to, or extensions of the convex methods that were presented in the previous sections. More precisely, Section 7.1 presents so-called greedy algorithms, that aim at solving the cardinality-constrained problem and include matching pursuit, orthogonal matching pursuit and forward selection; Section 7.2 presents continuous optimization problems, in which the penalty is chosen to be closer to the so-called $\ell_0$-penalty (i.e., a penalization of the cardinality of the model regardless of the amplitude of the coefficients) at the expense of losing convexity, and corresponding optimization schemes. Section 7.3 discusses the application of sparse norms regularization to the problem of matrix factorization, which is intrinsically nonconvex, but for which the algorithms presented in the rest of this monograph are relevant.

Finally, we discuss briefly in Section 7.4 Bayesian approaches to sparsity and the relations to sparsity-inducing norms.

Section 8 presents experiments comparing the performance of the algorithms presented in Sections 2, 3, 4, 5, in terms of speed of convergence of the algorithms. Precisely, Section 8.1 is devoted to the $\ell_1$-regularization case, and Sections 8.2 and 8.3 are respectively covering the $\ell_1/\ell_p$-norms with disjoint groups and to more general structured cases.

We discuss briefly methods and cases which were not covered in the rest of the monograph in Section 9 and we conclude in Section 10.

Some of the material from this monograph is taken from an earlier book chapter [12] and the dissertations of Rodolphe Jenatton [65] and Julien Mairal [85].

## 1.1 Notation

Vectors are denoted by bold lower case letters and matrices by upper case ones. We define for $q \geq 1$ the $\ell_q$-norm of a vector $\boldsymbol{x}$ in $\mathbb{R}^n$ as $\|\boldsymbol{x}\|_q := (\sum_{i=1}^n |\boldsymbol{x}_i|^q)^{1/q}$, where $\boldsymbol{x}_i$ denotes the $i$th coordinate of $\boldsymbol{x}$, and $\|\boldsymbol{x}\|_\infty := \max_{i=1,\ldots,n} |\boldsymbol{x}_i| = \lim_{q\to\infty} \|\boldsymbol{x}\|_q$. We also define the $\ell_0$-penalty as the number of nonzero elements in a vector[3]: $\|\boldsymbol{x}\|_0 := \#\{i \text{ s.t. } \boldsymbol{x}_i \neq 0\} = \lim_{q\to 0^+} (\sum_{i=1}^n |\boldsymbol{x}_i|^q)$. We consider the Frobenius norm of a matrix $\boldsymbol{X}$ in $\mathbb{R}^{m\times n}$: $\|\boldsymbol{X}\|_{\mathrm{F}} := (\sum_{i=1}^m \sum_{j=1}^n \boldsymbol{X}_{ij}^2)^{1/2}$, where $\boldsymbol{X}_{ij}$ denotes the entry of $\boldsymbol{X}$ at row $i$ and column $j$. For an integer $n > 0$, and for any subset $J \subseteq \{1,\ldots,n\}$, we denote by $\boldsymbol{x}_J$ the vector of size $|J|$ containing the entries of a vector $\boldsymbol{x}$ in $\mathbb{R}^n$ indexed by $J$, and by $\boldsymbol{X}_J$ the matrix in $\mathbb{R}^{m\times|J|}$ containing the $|J|$ columns of a matrix $\boldsymbol{X}$ in $\mathbb{R}^{m\times n}$ indexed by $J$.

## 1.2 Loss Functions

We consider in this monograph convex optimization problems of the form

$$\min_{\boldsymbol{w}\in\mathbb{R}^p} f(\boldsymbol{w}) + \lambda\Omega(\boldsymbol{w}), \tag{1.1}$$

[3] Note that it would be more proper to write $\|\boldsymbol{x}\|_0^0$ instead of $\|\boldsymbol{x}\|_0$ to be consistent with the traditional notation $\|\boldsymbol{x}\|_q$. However, for the sake of simplicity, we will keep this notation unchanged in the rest of the monograph.

where $f: \mathbb{R}^p \to \mathbb{R}$ is a convex differentiable function and $\Omega: \mathbb{R}^p \to \mathbb{R}$ is a sparsity-inducing — typically nonsmooth and non-Euclidean — norm.

In supervised learning, we predict outputs $y$ in $\mathcal{Y}$ from observations $\boldsymbol{x}$ in $\mathcal{X}$; these observations are usually represented by $p$-dimensional vectors with $\mathcal{X} = \mathbb{R}^p$. In this supervised setting, $f$ generally corresponds to the empirical risk of a loss function $\ell: \mathcal{Y} \times \mathbb{R} \to \mathbb{R}_+$. More precisely, given $n$ pairs of data points $\{(\boldsymbol{x}^{(i)}, y^{(i)}) \in \mathbb{R}^p \times \mathcal{Y};\ i = 1, \dots, n\}$, we have for linear models[4] $f(\boldsymbol{w}) := \frac{1}{n} \sum_{i=1}^n \ell(y^{(i)}, \boldsymbol{w}^\top \boldsymbol{x}^{(i)})$. Typical examples of differentiable loss functions are the square loss for least squares regression, i.e., $\ell(y, \hat{y}) = \frac{1}{2}(y - \hat{y})^2$ with $y$ in $\mathbb{R}$, and the logistic loss $\ell(y, \hat{y}) = \log(1 + e^{-y\hat{y}})$ for logistic regression, with $y$ in $\{-1, 1\}$. Clearly, several loss functions of interest are nondifferentiable, such as the hinge loss $\ell(y, \hat{y}) = (1 - y\hat{y})_+$ or the absolute deviation loss $\ell(y, \hat{y}) = |y - \hat{y}|$, for which most of the approaches we present in this monograph would not be applicable or require appropriate modifications. Given the tutorial character of this monograph, we restrict ourselves to smooth functions $f$, which we consider is a reasonably broad setting, and we refer the interested reader to appropriate references in Section 9. We refer the readers to [126] for a more complete description of loss functions.

**Penalty or constraint?** Given our convex data-fitting term $f(\boldsymbol{w})$, we consider in this monograph adding a convex penalty $\lambda\Omega(\boldsymbol{w})$. Within such a convex optimization framework, this is essentially equivalent to adding a constraint of the form $\Omega(\boldsymbol{w}) \leq \mu$. More precisely, under weak assumptions on $f$ and $\Omega$ (on top of convexity), from Lagrange multiplier theory (see [20], Section 4.3) $\boldsymbol{w}$ is a solution of the constrained problem for a certain $\mu > 0$ if and only if it is a solution of the penalized problem for a certain $\lambda \geq 0$. Thus, the two regularization paths, i.e., the set of solutions when $\lambda$ and $\mu$ vary, are equivalent. However, there is no direct mapping between corresponding values of $\lambda$ and $\mu$. Moreover, in a machine learning context, where the parameters $\lambda$ and $\mu$ have to be selected, for example, through cross-validation, the penalized formulation tends to be empirically easier to tune, as the performance is

[4] In Section 1.5, we consider extensions to nonlinear predictors through multiple kernel learning.

usually quite robust to small changes in $\lambda$, while it is not robust to small changes in $\mu$. Finally, we could also replace the penalization with a norm by a penalization with the squared norm. Indeed, following the same reasoning as for the nonsquared norm, a penalty of the form $\lambda\Omega(\boldsymbol{w})^2$ is "equivalent" to a constraint of the form $\Omega(\boldsymbol{w})^2 \leqslant \mu$, which itself is equivalent to $\Omega(\boldsymbol{w}) \leqslant \mu^{1/2}$, and thus to a penalty of the form $\lambda'\Omega(\boldsymbol{w})^2$, for $\lambda' \neq \lambda$. Thus, using a squared norm, as is often done in the context of multiple kernel learning (see Section 1.5), does not change the regularization properties of the formulation.

## 1.3 Sparsity-Inducing Norms

In this section, we present various norms as well as their main sparsity-inducing effects. These effects may be illustrated geometrically through the singularities of the corresponding unit balls (see Figure 1.4).

**Sparsity through the $\ell_1$-norm.** When one knows *a priori* that the solutions $\boldsymbol{w}^\star$ of problem (1.1) should have a few nonzero coefficients, $\Omega$ is often chosen to be the $\ell_1$-norm, i.e., $\Omega(\boldsymbol{w}) = \sum_{j=1}^{p} |\boldsymbol{w}_j|$. This leads for instance to the Lasso [133] or basis pursuit [37] with the square loss and to $\ell_1$-regularized logistic regression (see, for instance, [75, 127]) with the logistic loss. Regularizing by the $\ell_1$-norm is known to induce sparsity in the sense that, a number of coefficients of $\boldsymbol{w}^\star$, depending on the strength of the regularization, will be *exactly* equal to zero.

**$\ell_1/\ell_q$-norms.** In some situations, the coefficients of $\boldsymbol{w}^\star$ are naturally partitioned in subsets, or *groups*, of variables. This is typically the case, when working with ordinal variables.[5] It is then natural to select or remove *simultaneously* all the variables forming a group. A regularization norm exploiting explicitly this group structure, or $\ell_1$-*group norm*, can be shown to improve the prediction performance and/or interpretability of the learned models [61, 83, 106, 116, 141, 156]. The

[5] Ordinal variables are integer-valued variables encoding levels of a certain feature, such as levels of severity of a certain symptom in a biomedical application, where the values do not correspond to an intrinsic linear scale: in that case it is common to introduce a vector of binary variables, each encoding a specific level of the symptom, that encodes collectively this single feature.

arguably simplest group norm is the so-called-$\ell_1/\ell_2$ norm:

$$\Omega(\boldsymbol{w}) := \sum_{g \in \mathcal{G}} d_g \|\boldsymbol{w}_g\|_2, \tag{1.2}$$

where $\mathcal{G}$ is a partition of $\{1, \ldots, p\}$, $(d_g)_{g \in \mathcal{G}}$ are some strictly positive weights, and $\boldsymbol{w}_g$ denotes the vector in $\mathbb{R}^{|g|}$ recording the coefficients of $\boldsymbol{w}$ indexed by $g$ in $\mathcal{G}$. Without loss of generality we may assume all weights $(d_g)_{g \in \mathcal{G}}$ to be equal to one (when $\mathcal{G}$ is a partition, we can rescale the components of $\boldsymbol{w}$ appropriately). As defined in Equation (1.2), $\Omega$ is known as a mixed $\ell_1/\ell_2$-norm. It behaves like an $\ell_1$-norm on the vector $(\|\boldsymbol{w}_g\|_2)_{g \in \mathcal{G}}$ in $\mathbb{R}^{|\mathcal{G}|}$, and therefore, $\Omega$ induces group sparsity. In other words, each $\|\boldsymbol{w}_g\|_2$, and equivalently each $\boldsymbol{w}_g$, is encouraged to be set to zero. On the other hand, within the groups $g$ in $\mathcal{G}$, the $\ell_2$-norm does not promote sparsity. Combined with the square loss, it leads to the *group Lasso* formulation [141, 156]. Note that when $\mathcal{G}$ is the set of singletons, we retrieve the $\ell_1$-norm. More general mixed $\ell_1/\ell_q$-norms for $q > 1$ are also used in the literature [157] (using $q = 1$ leads to a weighted $\ell_1$-norm with no group-sparsity effects):

$$\Omega(\boldsymbol{w}) = \sum_{g \in \mathcal{G}} \|\boldsymbol{w}_g\|_q := \sum_{g \in \mathcal{G}} d_g \left\{ \sum_{j \in g} |\boldsymbol{w}_j|^q \right\}^{1/q}.$$

In practice though, the $\ell_1/\ell_2$- and $\ell_1/\ell_\infty$-settings remain the most popular ones. Note that using $\ell_\infty$-norms may have the undesired effect to favor solutions $\boldsymbol{w}$ with many components of equal magnitude (due to the extra nondifferentiabilities away from zero). Grouped $\ell_1$-norms are typically used when extra-knowledge is available regarding an appropriate partition, in particular in the presence of categorical variables with orthogonal encoding [116], for multi-task learning where joint variable selection is desired [106], and for multiple kernel learning (see Section 1.5).

**Norms for overlapping groups: a direct formulation.** In an attempt to better encode structural links between variables at play (e.g., spatial or hierarchical links related to the physics of the problem at hand), recent research has explored the setting where $\mathcal{G}$ in Equation (1.2) can contain groups of variables that *overlap* [9, 64, 66,

73, 121, 157]. In this case, if the groups span the entire set of variables, $\Omega$ is still a norm, and it yields sparsity in the form of specific patterns of variables. More precisely, the solutions $\boldsymbol{w}^\star$ of problem (1.1) can be shown to have a set of zero coefficients, or simply *zero pattern*, that corresponds to a union of some groups $g$ in $\mathcal{G}$ [66]. This property makes it possible to control the sparsity patterns of $\boldsymbol{w}^\star$ by appropriately defining the groups in $\mathcal{G}$. Note that here the weights $d_g$ should not be taken equal to one (see, e.g., [66] for more details). This form of *structured sparsity* has notably proven to be useful in various contexts, which we now illustrate through concrete examples:

— **One-dimensional sequence:** Given $p$ variables organized in a sequence, if we want to select only contiguous nonzero patterns, we represent in Figure 1.1 the set of groups $\mathcal{G}$ to consider. In this case, we have $|\mathcal{G}| = O(p)$. Imposing the contiguity of the nonzero patterns is for instance relevant in the context of time series, or for the diagnosis of tumors, based on the profiles of arrayCGH [112]. Indeed, because of the specific spatial organization of bacterial artificial chromosomes along the genome, the set of discriminative features is expected to have specific contiguous patterns.

— **Two-dimensional grid:** In the same way, assume now that the $p$ variables are organized on a two-dimensional grid. If we want the possible nonzero patterns $\mathcal{P}$ to be the set of all rectangles on this grid, the appropriate groups $\mathcal{G}$ to consider can be shown (see [66]) to be those represented in Figure 1.2. In this setting, we have $|\mathcal{G}| = O(\sqrt{p})$.

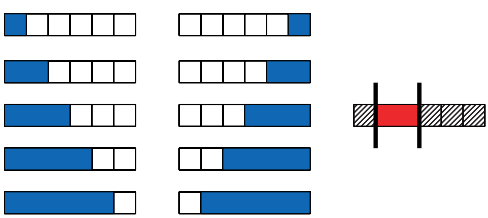


Fig. 1.1. (Left) The set of blue groups to penalize in order to select contiguous patterns in a sequence. (Right) In red, an example of such a nonzero pattern with its corresponding zero pattern (hatched area).

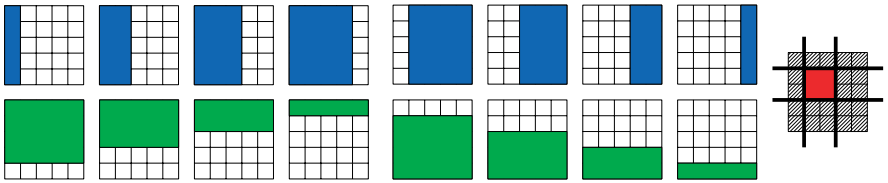


Fig. 1.2. Vertical and horizontal groups: (Left) the set of blue and green groups to penalize in order to select rectangles. (Right) In red, an example of nonzero pattern recovered in this setting, with its corresponding zero pattern (hatched area).

Sparsity-inducing regularizations built upon such group structures have resulted in good performances for background subtraction [62, 86, 88], topographic dictionary learning [72, 88], wavelet-based denoising [111], and for face recognition with corruption by occlusions [70].

— **Hierarchical structure:** A third interesting example assumes that the variables have a hierarchical structure. Specifically, we consider that the $p$ variables correspond to the nodes of a tree $\mathcal{T}$ (or a forest of trees). Moreover, we assume that we want to select the variables according to a certain order: a feature can be selected only if all its ancestors in $\mathcal{T}$ are already selected. This hierarchical rule can be shown to lead to the family of groups displayed on Figure 1.3.
This resulting penalty was first used in [157]; since then, this group structure has led to numerous applications, for instance, wavelet-based denoising [15, 62, 69, 157], hierarchical dictionary learning for both topic modeling and image restoration [68, 69], log-linear models for the selection of potential orders of interaction in a probabilistic graphical model [121], bioinformatics, to exploit the tree structure of gene networks for multi-task regression [73], and multi-scale mining of fMRI data for the prediction of some cognitive task [67]. More recently, this hierarchical penalty was proved to be efficient for template selection in natural language processing [92].

— **Extensions:** The possible choices for the sets of groups $\mathcal{G}$ are not limited to the aforementioned examples. More

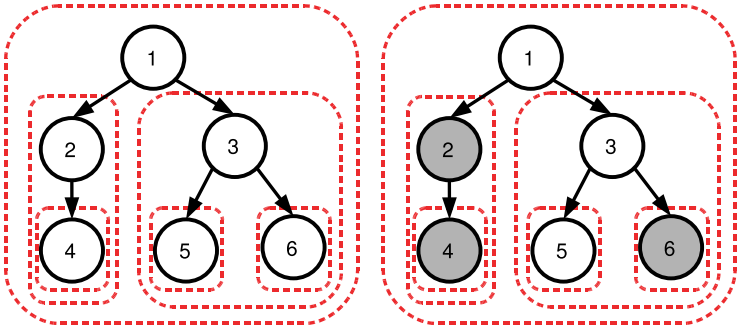



Fig. 1.3. Left: example of a tree-structured set of groups $\mathcal{G}$ (dashed contours in red), corresponding to a tree $\mathcal{T}$ with $p = 6$ nodes represented by black circles. Right: example of a sparsity pattern induced by the tree-structured norm corresponding to $\mathcal{G}$; the groups $\{2,4\}, \{4\}$ and $\{6\}$ are set to zero, so that the corresponding nodes (in gray) that form subtrees of $\mathcal{T}$ are removed. The remaining nonzero variables $\{1,3,5\}$ form a rooted and connected subtree of $\mathcal{T}$. This sparsity pattern obeys the following equivalent rules: (i) if a node is selected, the same goes for all its ancestors; (ii) if a node is not selected, then its descendant are not selected.

complicated topologies can be considered, for instance, three-dimensional spaces discretized in cubes or spherical volumes discretized in slices; for instance, see [143] for an application to neuroimaging that pursues this idea. Moreover, directed acyclic graphs that extends the trees presented in Figure 1.3 have notably proven to be useful in the context of hierarchical variable selection [9, 121, 157],

**Norms for overlapping groups: a latent variable formulation.** The family of norms defined in Equation (1.2) is adapted to *intersection-closed* sets of nonzero patterns. However, some applications exhibit structures that can be more naturally modelled by *union-closed* families of supports. This idea was developed in [64, 105] where, given a set of groups $\mathcal{G}$, the following *latent group Lasso* norm was proposed:

$$\Omega_{\text{union}}(\boldsymbol{w}) := \min_{\boldsymbol{v} \in \mathbb{R}^{p \times |\mathcal{G}|}} \sum_{g \in \mathcal{G}} d_g \|\boldsymbol{v}^g\|_q, \quad \text{s.t.} \begin{cases} \sum_{g \in \mathcal{G}} \boldsymbol{v}^g = \boldsymbol{w}, \\ \forall g \in \mathcal{G},\ \boldsymbol{v}^g_j = 0 \quad \text{if } j \notin g. \end{cases}$$

The idea is to introduce latent parameter vectors $\boldsymbol{v}^g$ constrained each to be supported on the corresponding group $g$, which should explain $\boldsymbol{w}$

linearly and which are themselves regularized by a usual $\ell_1/\ell_q$-norm. $\Omega_{\text{union}}$ reduces to the usual $\ell_1/\ell_q$ norm when groups are disjoint and provides therefore a different generalization of the latter to the case of overlapping groups than the norm considered in the previous paragraphs. In fact, it is easy to see that solving Equation (1.1) with the norm $\Omega_{\text{union}}$ is equivalent to solving

$$\min_{(\boldsymbol{v}^g \in \mathbb{R}^{|g|})_{g \in \mathcal{G}}} \frac{1}{n} \sum_{i=1}^{n} \ell\left(y^{(i)}, \sum_{g \in \mathcal{G}} \boldsymbol{v}_g^{g\top} \boldsymbol{x}_g^{(i)}\right) + \lambda \sum_{g \in \mathcal{G}} d_g \|\boldsymbol{v}^g\|_q \tag{1.3}$$

and setting $\boldsymbol{w} = \sum_{g \in \mathcal{G}} \boldsymbol{v}^g$. This last equation shows that using the norm $\Omega_{\text{union}}$ can be interpreted as implicitly duplicating the variables belonging to several groups and regularizing with a weighted $\ell_1/\ell_q$ norm for disjoint groups in the expanded space. It should be noted that a careful choice of the weights is much more important in the situation of overlapping groups than in the case of disjoint groups, as it influences possible sparsity patterns [105].

This latent variable formulation pushes some of the vectors $\boldsymbol{v}^g$ to zero while keeping others with no zero components, hence leading to a vector $\boldsymbol{w}$ with a support which is in general the union of the selected groups. Interestingly, it can be seen as a convex relaxation of a non-convex penalty encouraging similar sparsity patterns which was introduced by [62]. Moreover, this norm can also be interpreted as a particular case of the family of *atomic norms*, which were recently introduced by [35].

*Graph Lasso.* One type of a priori knowledge commonly encountered takes the form of graph defined on the set of input variables, which is such that connected variables are more likely to be simultaneously relevant or irrelevant; this type of prior is common in genomics where regulation, co-expression or interaction networks between genes (or their expression level) used as predictors are often available. To favor the selection of neighbors of a selected variable, it is possible to consider the edges of the graph as groups in the previous formulation (see [64, 111]).

*Patterns consisting of a small number of intervals.* A quite similar situation occurs, when one knows a priori—typically for variables forming sequences (times series, strings, polymers)—that the support should consist of a small number of connected subsequences. In that case,

one can consider the sets of variables forming connected subsequences (or connected subsequences of length at most $k$) as the overlapping groups.

**Multiple kernel learning.** For most of the sparsity-inducing terms described in this monograph, we may replace real variables and their absolute values by pre-defined groups of variables with their Euclidean norms (we have already seen such examples with $\ell_1/\ell_2$-norms), or more generally, by members of reproducing kernel Hilbert spaces. As shown in Section 1.5, most of the tools that we present in this monograph are applicable to this case as well, through appropriate modifications and borrowing of tools from kernel methods. These tools have applications in particular in multiple kernel learning. Note that this extension requires tools from convex analysis presented in Section 1.4.

**Trace norm.** In learning problems on matrices, such as matrix completion, the rank plays a similar role to the cardinality of the support for vectors. Indeed, the rank of a matrix $\mathbf{M}$ may be seen as the number of non-zero singular values of $\mathbf{M}$. The rank of $\mathbf{M}$ however is not a continuous function of $\mathbf{M}$, and, following the convex relaxation of the $\ell_0$-pseudo-norm into the $\ell_1$-norm, we may relax the rank of $\mathbf{M}$ into the sum of its singular values, which happens to be a norm, and is often referred to as the trace norm or nuclear norm of $\mathbf{M}$, and which we denote by $\|\mathbf{M}\|_*$. As shown in this monograph, many of the tools designed for the $\ell_1$-norm may be extended to the trace norm. Using the trace norm as a convex surrogate for rank has many applications in control theory [48], matrix completion [1, 130], multi-task learning [109], or multi-label classification [4], where low-rank priors are adapted.

**Sparsity-inducing properties: A geometrical intuition.** Although we consider in Equation (1.1) a regularized formulation, as already described in Section 1.2, we could equivalently focus on a *constrained* problem, that is,

$$\min_{\boldsymbol{w}\in\mathbb{R}^p} f(\boldsymbol{w}) \quad \text{such that} \quad \Omega(\boldsymbol{w}) \leq \mu, \tag{1.4}$$

for some $\mu \in \mathbb{R}_+$. The set of solutions of Equation (1.4) parameterized by $\mu$ is the same as that of Equation (1.1), as described by some value

of $\lambda_\mu$ depending on $\mu$ (e.g., see Section 3.2 in [20]). At optimality, the gradient of $f$ evaluated at any solution $\hat{\boldsymbol{w}}$ of (1.4) is known to belong to the normal cone of $\mathcal{B} = \{\boldsymbol{w} \in \mathbb{R}^p;\ \Omega(\boldsymbol{w}) \leq \mu\}$ at $\hat{\boldsymbol{w}}$ [20]. In other words, for sufficiently small values of $\mu$, i.e., so that the constraint is active, the level set of $f$ for the value $f(\hat{\boldsymbol{w}})$ is tangent to $\mathcal{B}$.

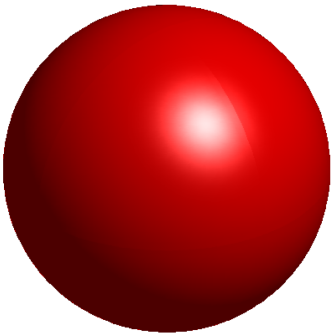
As a consequence, the geometry of the ball $\mathcal{B}$ is directly related to the properties of the solutions $\hat{\boldsymbol{w}}$. If $\Omega$ is taken to be the $\ell_2$-norm, then the resulting ball $\mathcal{B}$ is the standard, isotropic, "round" ball that does not favor any specific direction of the space. On the other hand, when $\Omega$ is the $\ell_1$-norm, $\mathcal{B}$ corresponds to a diamond-shaped pattern in two dimensions, and to a pyramid in three dimensions. In particular, $\mathcal{B}$ is anisotropic and exhibits some singular points due to the extra non-smoothness of $\Omega$. Moreover, these singular points are located along the axis of $\mathbb{R}^p$, so that if the level set of $f$ happens to be tangent at one of those points, sparse solutions are obtained. We display in Figure 1.4 the balls $\mathcal{B}$ for the $\ell_1$-, $\ell_2$-norms, and two different grouped $\ell_1/\ell_2$-norms.

**Extensions.** The design of sparsity-inducing norms is an active field of research and similar tools to the ones we present here can be derived for other norms. As shown in Section 3, computing the proximal operator readily leads to efficient algorithms, and for the extensions we present below, these operators can be efficiently computed.

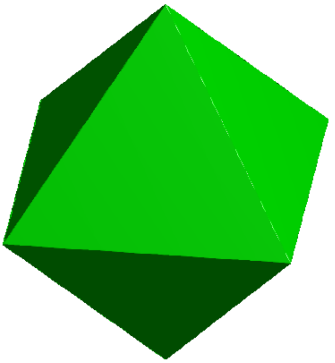
In order to impose prior knowledge on the support of predictor, the norms based on overlapping $\ell_1/\ell_\infty$-norms can be shown to be convex relaxations of submodular functions of the support, and further ties can be made between convex optimization and combinatorial optimization (see [10] for more details). Moreover, similar developments may be carried through for norms which try to enforce that the predictors have many equal components and that the resulting clusters have specific shapes, e.g., contiguous in a pre-defined order, see some examples in Section 3, and, e.g., [11, 33, 86, 134, 144] and references therein.

## 1.4 Optimization Tools

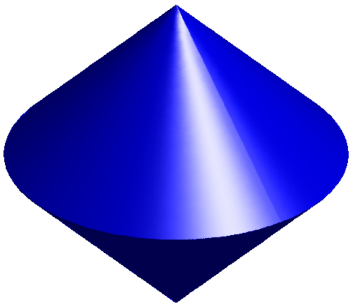
The tools used in this monograph are relatively basic and should be accessible to a broad audience. Most of them can be found in



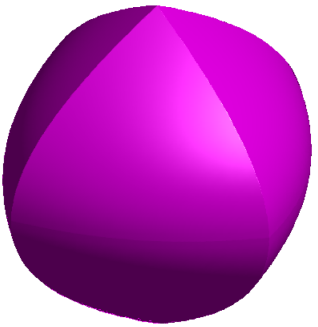
(a) $\ell_2$-norm ball.



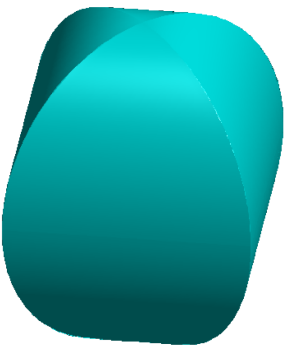
(b) $\ell_1$-norm ball.



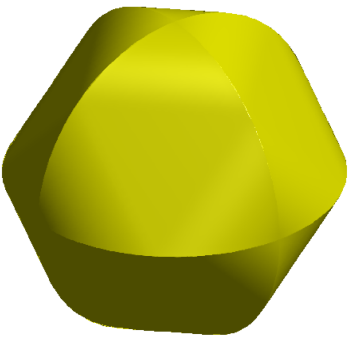
(c) $\ell_1/\ell_2$-norm ball:
$\Omega(\boldsymbol{w}) = \|\boldsymbol{w}_{\{1,2\}}\|_2 + |\boldsymbol{w}_3|$.



(d) $\ell_1/\ell_2$-norm ball:
$\Omega(\boldsymbol{w}) = \|\boldsymbol{w}\|_2 + |\boldsymbol{w}_1| + |\boldsymbol{w}_2|$.



(e) $\Omega_{\text{union}}$ ball for
$\mathcal{G} = \big\{\{1,3\},\{2,3\}\big\}$.



(f) $\Omega_{\text{union}}$ ball for
$\mathcal{G} = \big\{\{1,3\},\{2,3\},\{1,2\}\big\}$.

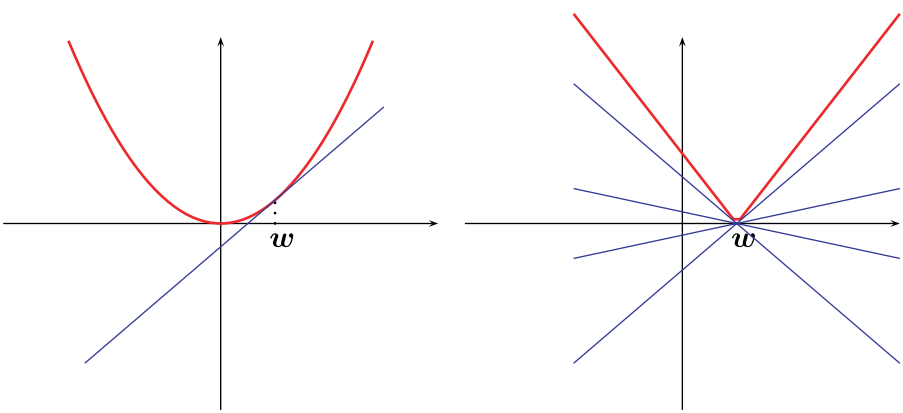
Fig. 1.4. Comparison between different balls of sparsity-inducing norms in three dimensions. The singular points appearing on these balls describe the sparsity-inducing behavior of the underlying norms $\Omega$.

classical books on convex optimization [18, 20, 25, 104], but for self-containedness, we present here a few of them related to nonsmooth unconstrained optimization. In particular, these tools allow the derivation of rigorous approximate optimality conditions based on duality gaps (instead of relying on weak stopping criteria based on small changes or low-norm gradients).

**Subgradients.** Given a convex function $g:\mathbb{R}^p \to \mathbb{R}$ and a vector $\boldsymbol{w}$ in $\mathbb{R}^p$, let us define the *subdifferential* of $g$ at $\boldsymbol{w}$ as

$$\partial g(\boldsymbol{w}) := \{\boldsymbol{z} \in \mathbb{R}^p \mid g(\boldsymbol{w}) + \boldsymbol{z}^\top(\boldsymbol{w}' - \boldsymbol{w}) \leq g(\boldsymbol{w}') \text{ for all vectors } \boldsymbol{w}' \in \mathbb{R}^p\}.$$

The elements of $\partial g(\boldsymbol{w})$ are called the *subgradients* of $g$ at $\boldsymbol{w}$. Note that all convex functions defined on $\mathbb{R}^p$ have non-empty subdifferentials at every point. This definition admits a clear geometric interpretation: any subgradient $\boldsymbol{z}$ in $\partial g(\boldsymbol{w})$ defines an affine function $\boldsymbol{w}' \mapsto g(\boldsymbol{w}) + \boldsymbol{z}^\top(\boldsymbol{w}' - \boldsymbol{w})$ which is tangent to the graph of the function $g$ (because of the convexity of $g$, it is a lower-bounding tangent). Moreover, there is a bijection (one-to-one correspondence) between such "tangent affine functions" and the subgradients, as illustrated in Figure 1.5.




(a) Smooth case. (b) Non-smooth case.

Fig. 1.5. Red curves represent the graph of a smooth (left) and a nonsmooth (right) function $f$. Blue affine functions represent subgradients of the function $f$ at a point $\boldsymbol{w}$.

Subdifferentials are useful for studying nonsmooth optimization problems because of the following proposition (whose proof is straightforward from the definition):

---

**Proposition 1.1 (Subgradients at Optimality).**
For any convex function $g: \mathbb{R}^p \to \mathbb{R}$, a point $\boldsymbol{w}$ in $\mathbb{R}^p$ is a global minimum of $g$ if and only if the condition $0 \in \partial g(\boldsymbol{w})$ holds.

---

Note that the concept of subdifferential is mainly useful for nonsmooth functions. If $g$ is differentiable at $\boldsymbol{w}$, the set $\partial g(\boldsymbol{w})$ is indeed the singleton $\{\nabla g(\boldsymbol{w})\}$, where $\nabla g(\boldsymbol{w})$ is the gradient of $g$ at $\boldsymbol{w}$, and the condition $0 \in \partial g(\boldsymbol{w})$ reduces to the classical first-order optimality condition $\nabla g(\boldsymbol{w}) = 0$. As a simple example, let us consider the following optimization problem

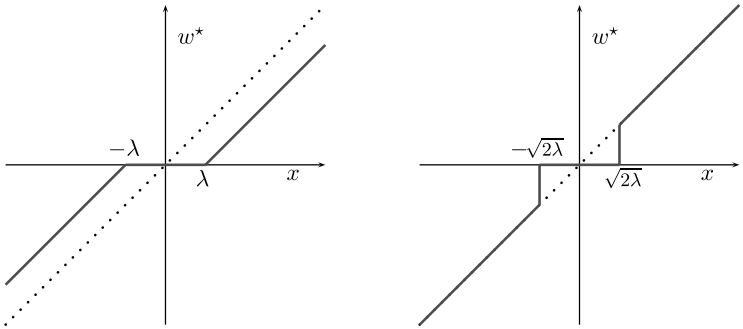
$$\min_{w \in \mathbb{R}} \frac{1}{2}(x - w)^2 + \lambda |w|.$$

Applying the previous proposition and noting that the subdifferential $\partial |\cdot|$ is $\{+1\}$ for $w > 0$, $\{-1\}$ for $w < 0$ and $[-1, 1]$ for $w = 0$, it is easy to show that the unique solution admits a closed form called the *soft-thresholding* operator, following a terminology introduced in [42]; it can be written

$$w^\star = \begin{cases} 0, & \text{if } |x| \leq \lambda \\ (1 - \frac{\lambda}{|x|})x, & \text{otherwise,} \end{cases} \tag{1.5}$$

or equivalently $w^\star = \text{sign}(x)(|x| - \lambda)_+$, where $\text{sign}(x)$ is equal to 1 if $x > 0$, $-1$ if $x < 0$ and 0 if $x = 0$. This operator is a core component of many optimization techniques for sparse estimation, as we shall see later. Its counterpart for nonconvex optimization problems is the hard-thresholding operator. Both of them are presented in Figure 1.6. Note that similar developments could be carried through using directional derivatives instead of subgradients (see, e.g., [20]).

**Dual norm and optimality conditions.** The next concept we introduce is the dual norm, which is important to study sparsity-inducing regularizations [9, 66, 99]. It notably arises in the analysis of estimation bounds [99], and in the design of working-set strategies




(a) soft-thresholding operator, $w^\star = \mathrm{sign}(x)(|x| - \lambda)_+$, $\min_w \frac{1}{2}(x-w)^2 + \lambda|w|$.

(b) hard-thresholding operator $w^\star = \mathbf{1}_{|x| \geq \sqrt{2\lambda}} x$ $\min_w \frac{1}{2}(x-w)^2 + \lambda \mathbf{1}_{|w|>0}$.

Fig. 1.6. Soft- and hard-thresholding operators.

as will be shown in Section 6.1. The dual norm $\Omega^*$ of the norm $\Omega$ is defined for any vector $\boldsymbol{z}$ in $\mathbb{R}^p$ by

$$\Omega^*(\boldsymbol{z}) := \max_{\boldsymbol{w} \in \mathbb{R}^p} \boldsymbol{z}^\top \boldsymbol{w} \text{ such that } \Omega(\boldsymbol{w}) \leq 1. \tag{1.6}$$

Moreover, the dual norm of $\Omega^*$ is $\Omega$ itself, and as a consequence, the formula above holds also if the roles of $\Omega$ and $\Omega^*$ are exchanged. It is easy to show that in the case of an $\ell_q$-norm, $q \in [1; +\infty]$, the dual norm is the $\ell_{q'}$-norm, with $q'$ in $[1; +\infty]$ such that $\frac{1}{q} + \frac{1}{q'} = 1$. In particular, the $\ell_1$- and $\ell_\infty$-norms are dual to each other, and the $\ell_2$-norm is self-dual (dual to itself).

The dual norm plays a direct role in computing optimality conditions of sparse regularized problems. By applying Proposition 1.1 to Equation (1.1), we obtain the following proposition:

---

**Proposition 1.2 (Optimality conditions for Equation (1.1)).** Let us consider problem (1.1) where $\Omega$ is a norm on $\mathbb{R}^p$. A vector $\boldsymbol{w}$ in $\mathbb{R}^p$ is optimal if and only if $-\frac{1}{\lambda}\nabla f(\boldsymbol{w}) \in \partial\Omega(\boldsymbol{w})$ with

$$\partial\Omega(\boldsymbol{w}) = \begin{cases} \{\boldsymbol{z} \in \mathbb{R}^p;\ \Omega^*(\boldsymbol{z}) \leq 1\}, & \text{if } \boldsymbol{w} = 0, \\ \{\boldsymbol{z} \in \mathbb{R}^p;\ \Omega^*(\boldsymbol{z}) = 1 \text{ and } \boldsymbol{z}^\top \boldsymbol{w} = \Omega(\boldsymbol{w})\}, & \text{otherwise.} \end{cases} \tag{1.7}$$

---

Computing the subdifferential of a norm is a classical course exercise [20] and its proof will be presented in the next section, in Remark 1.1. As a consequence, the vector $\mathbf{0}$ is solution if and only if $\Omega^*\big(\nabla f(\mathbf{0})\big) \leq \lambda$. Note that this shows that for all $\lambda$ larger than $\Omega^*\big(\nabla f(\mathbf{0})\big)$, $\boldsymbol{w} = \mathbf{0}$ is a solution of the regularized optimization problem (hence this value is the start of the non-trivial regularization path).

These general optimality conditions can be specialized to the Lasso problem [133], also known as basis pursuit [37]:

$$\min_{\boldsymbol{w} \in \mathbb{R}^p} \frac{1}{2n} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 + \lambda \|\boldsymbol{w}\|_1, \tag{1.8}$$

where $\boldsymbol{y}$ is in $\mathbb{R}^n$, and $\boldsymbol{X}$ is a design matrix in $\mathbb{R}^{n \times p}$. With Equation (1.7) in hand, we can now derive necessary and sufficient optimality conditions:

---

**Proposition 1.3 (Optimality conditions for the Lasso).**
A vector $\boldsymbol{w}$ is a solution of the Lasso problem (1.8) if and only if

$$\forall j = 1, \dots, p, \quad \begin{cases} |\boldsymbol{X}_j^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})| \leq n\lambda, & \text{if } \boldsymbol{w}_j = 0 \\ \boldsymbol{X}_j^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}) = n\lambda \operatorname{sign}(\boldsymbol{w}_j), & \text{if } \boldsymbol{w}_j \neq 0, \end{cases} \tag{1.9}$$

where $\boldsymbol{X}_j$ denotes the $j$th column of $\boldsymbol{X}$, and $\boldsymbol{w}_j$ the $j$th entry of $\boldsymbol{w}$.

---

*Proof.* We apply Proposition 1.2. The condition $-\frac{1}{\lambda} \nabla f(\boldsymbol{w}) \in \partial \|\boldsymbol{w}\|_1$ can be rewritten: $\boldsymbol{X}^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}) \in n\lambda \partial \|\boldsymbol{w}\|_1$, which is equivalent to: (i) if $\boldsymbol{w} = 0$, $\|\boldsymbol{X}^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})\|_\infty \leq n\lambda$ (using the fact that the $\ell_\infty$-norm is dual to the $\ell_1$-norm); (ii) if $\boldsymbol{w} \neq 0$, $\|\boldsymbol{X}^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w})\|_\infty = n\lambda$ and $\boldsymbol{w}^\top \boldsymbol{X}^\top (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}) = n\lambda \|\boldsymbol{w}\|_1$. It is then easy to check that these conditions are equivalent to Equation (1.9). □

As we will see in Section 6.2, it is possible to derive from these conditions interesting properties of the Lasso, as well as efficient algorithms for solving it. We have presented a useful duality tool for norms. More generally, there exists a related concept for convex functions, which we now introduce.

### 1.4.1 Fenchel Conjugate and Duality Gaps

Let us denote by $f^*$ the Fenchel conjugate of $f$ [115], defined by

$$f^*(\boldsymbol{z}) := \sup_{\boldsymbol{w} \in \mathbb{R}^p} [\boldsymbol{z}^\top \boldsymbol{w} - f(\boldsymbol{w})].$$

Fenchel conjugates are particularly useful to derive dual problems and duality gaps.[6] Under mild conditions, the conjugate of the conjugate of a convex function is itself, leading to the following representation of $f$ as a maximum of affine functions:

$$f(\boldsymbol{w}) = \sup_{\boldsymbol{z} \in \mathbb{R}^p} [\boldsymbol{z}^\top \boldsymbol{w} - f^*(\boldsymbol{z})].$$

In the context of this tutorial, it is notably useful to specify the expression of the conjugate of a norm. Perhaps surprisingly and misleadingly, the conjugate of a norm is not equal to its dual norm, but corresponds instead to the indicator function of the unit ball of its dual norm. More formally, let us introduce the indicator function $\iota_{\Omega^*}$ such that $\iota_{\Omega^*}(\boldsymbol{z})$ is equal to 0 if $\Omega^*(\boldsymbol{z}) \leq 1$ and $+\infty$ otherwise. Then, we have the following well-known results, which appears in several text books (e.g., see Example 3.26 in [25]):

---

**Proposition 1.4 (Fenchel conjugate of a norm).** Let $\Omega$ be a norm on $\mathbb{R}^p$. The following equality holds for any $\boldsymbol{z} \in \mathbb{R}^p$

$$\sup_{\boldsymbol{w} \in \mathbb{R}^p} [\boldsymbol{z}^\top \boldsymbol{w} - \Omega(\boldsymbol{w})] = \iota_{\Omega^*}(\boldsymbol{w}) = \begin{cases} 0, & \text{if } \Omega^*(\boldsymbol{z}) \leq 1 \\ +\infty, & \text{otherwise.} \end{cases}$$

---

*Proof.* On the one hand, assume that the dual norm of $\boldsymbol{z}$ is greater than 1, that is, $\Omega^*(\boldsymbol{z}) > 1$. According to the definition of the dual norm (see Equation (1.6)), and since the supremum is taken over the compact set $\{\boldsymbol{w} \in \mathbb{R}^p;\ \Omega(\boldsymbol{w}) \leq 1\}$, there exists a vector $\boldsymbol{w}$ in this ball such that $\Omega^*(\boldsymbol{z}) = \boldsymbol{z}^\top \boldsymbol{w} > 1$. For any scalar $t \geq 0$, consider $\boldsymbol{v} = t\boldsymbol{w}$ and notice that

$$\boldsymbol{z}^\top \boldsymbol{v} - \Omega(\boldsymbol{v}) = t[\boldsymbol{z}^\top \boldsymbol{w} - \Omega(\boldsymbol{w})] \geq t,$$

[6] For many of our norms, *conic* duality tools would suffice (see, e.g., [25]).

which shows that when $\Omega^*(\boldsymbol{z}) > 1$, the Fenchel conjugate is unbounded. Now, assume that $\Omega^*(\boldsymbol{z}) \leq 1$. By applying the generalized Cauchy–Schwarz's inequality, we obtain for any $\boldsymbol{w}$

$$\boldsymbol{z}^\top \boldsymbol{w} - \Omega(\boldsymbol{w}) \leq \Omega^*(\boldsymbol{z})\,\Omega(\boldsymbol{w}) - \Omega(\boldsymbol{w}) \leq 0.$$

Equality holds for $\boldsymbol{w} = \boldsymbol{0}$, and the conclusion follows. □

An important and useful duality result is the so-called Fenchel–Young inequality (see [20]), which we will shortly illustrate geometrically:

---

**Proposition 1.5 (Fenchel–Young inequality).** Let $\boldsymbol{w}$ be a vector in $\mathbb{R}^p$, $f$ be a function on $\mathbb{R}^p$, and $\boldsymbol{z}$ be a vector in the domain of $f^*$ (which we assume non-empty). We have then the following inequality

$$f(\boldsymbol{w}) + f^*(\boldsymbol{z}) \geq \boldsymbol{w}^\top \boldsymbol{z},$$

with equality if and only if $\boldsymbol{z}$ is in $\partial f(\boldsymbol{w})$.

---

We can now illustrate geometrically the duality principle between a function and its Fenchel conjugate in Figure 1.7.

---

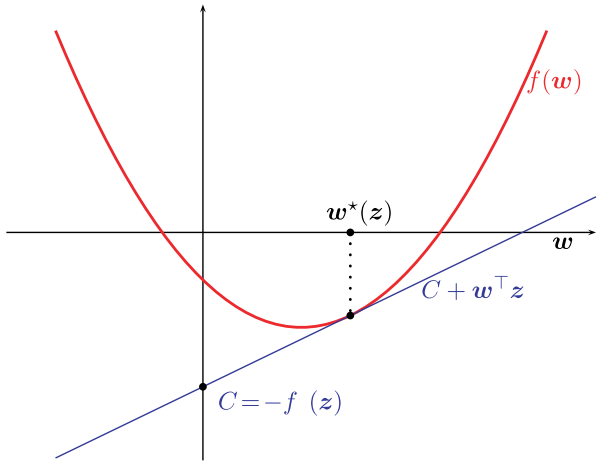
**Remark 1.1.** With Proposition 1.4 in place, we can formally (and easily) prove the relationship in Equation (1.7) that make explicit the subdifferential of a norm. Based on Proposition 1.4, we indeed know that the conjugate of $\Omega$ is $\iota_{\Omega^*}$. Applying the Fenchel–Young inequality (Proposition 1.5), we have

$$\boldsymbol{z} \in \partial\Omega(\boldsymbol{w}) \Leftrightarrow \Big[\boldsymbol{z}^\top \boldsymbol{w} = \Omega(\boldsymbol{w}) + \iota_{\Omega^*}(\boldsymbol{z})\Big],$$

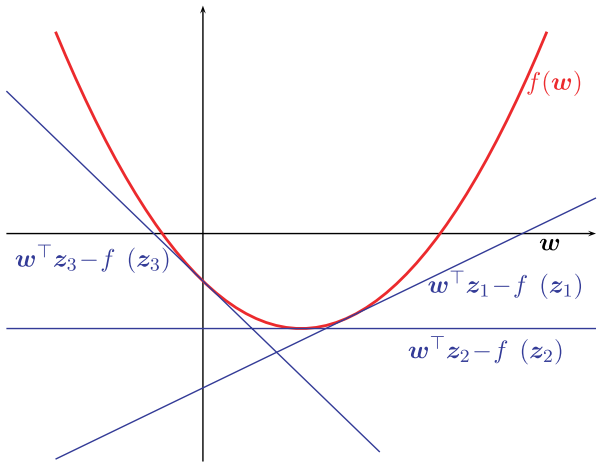
which leads to the desired conclusion.

---

For many objective functions, the Fenchel conjugate admits closed forms, and can therefore be computed efficiently [20]. Then, it is




(a) Fenchel conjugate, tangent hyperplanes and subgradients.




(b) The graph of $f$ is the envelope of the tangent hyperplanes $\mathcal{P}(\boldsymbol{z})$.

Fig. 1.7. For all $\boldsymbol{z}$ in $\mathbb{R}^p$, we denote by $\mathcal{P}(\boldsymbol{z})$ the hyperplane with normal $\boldsymbol{z}$ and tangent to the graph of the convex function $f$. (a) For any contact point between the graph of $f$ and an hyperplane $\mathcal{P}(\boldsymbol{z})$, we have that $f(\boldsymbol{w}) + f^*(\boldsymbol{z}) = \boldsymbol{w}^\top \boldsymbol{z}$ and $\boldsymbol{z}$ is in $\partial f(\boldsymbol{w})$ (the Fenchel–Young inequality is an equality). (b) The graph of $f$ is the convex envelope of the collection of hyperplanes $(\mathcal{P}(\boldsymbol{z}))_{\boldsymbol{z} \in \mathbb{R}^p}$.

possible to derive a duality gap for problem (1.1) from standard Fenchel duality arguments (see [20]), as shown in the following proposition:

---

**Proposition 1.6 (Duality for Problem (1.1)).** If $f^*$ and $\Omega^*$ are respectively, the Fenchel conjugate of a convex and differentiable function $f$ and the dual norm of $\Omega$, then we have

$$\max_{\boldsymbol{z}\in\mathbb{R}^p:\Omega^*(\boldsymbol{z})\leq\lambda} -f^*(\boldsymbol{z}) \leq \min_{\boldsymbol{w}\in\mathbb{R}^p} f(\boldsymbol{w}) + \lambda\Omega(\boldsymbol{w}). \tag{1.10}$$

Moreover, equality holds as soon as the domain of $f$ has non-empty interior.

---

*Proof.* This result is a specific instance of Theorem 3.3.5 in [20]. In particular, we use the fact that the conjugate of a norm $\Omega$ is the indicator function $\iota_{\Omega^*}$ of the unit ball of the dual norm $\Omega^*$ (see Proposition 1.4). □

If $\boldsymbol{w}^\star$ is a solution of Equation (1.1), and $\boldsymbol{w}, \boldsymbol{z}$ in $\mathbb{R}^p$ are such that $\Omega^*(\boldsymbol{z}) \leq \lambda$, this proposition implies that we have

$$f(\boldsymbol{w}) + \lambda\Omega(\boldsymbol{w}) \geq f(\boldsymbol{w}^\star) + \lambda\Omega(\boldsymbol{w}^\star) \geq -f^*(\boldsymbol{z}). \tag{1.11}$$

The difference between the left and right term of Equation (1.11) is called a duality gap. It represents the difference between the value of the primal objective function $f(\boldsymbol{w}) + \lambda\Omega(\boldsymbol{w})$ and a dual objective function $-f^*(\boldsymbol{z})$, where $\boldsymbol{z}$ is a dual variable. The proposition says that the duality gap for a pair of optima $\boldsymbol{w}^\star$ and $\boldsymbol{z}^\star$ of the primal and dual problem is equal to 0. When the optimal duality gap is zero one says that *strong duality* holds. In our situation, the duality gap for the pair of primal/dual problems in Equation (1.10), may be decomposed as the sum of two non-negative terms (as the consequence of Fenchel–Young inequality):

$$(f(\boldsymbol{w}) + f^*(\boldsymbol{z}) - \boldsymbol{w}^\top\boldsymbol{z}) + \lambda(\Omega(\boldsymbol{w}) + \boldsymbol{w}^\top(\boldsymbol{z}/\lambda) + \iota_{\Omega^*}(\boldsymbol{z}/\lambda)).$$

It is equal to zero if and only if the two terms are simultaneously equal to zero.

Duality gaps are important in convex optimization because they provide an upper bound on the difference between the current value of

an objective function and the optimal value, which makes it possible to set proper stopping criteria for iterative optimization algorithms. Given a current iterate $\boldsymbol{w}$, computing a duality gap requires choosing a "good" value for $\boldsymbol{z}$ (and in particular a feasible one). Given that at optimality, $\boldsymbol{z}(\boldsymbol{w}^\star) = \nabla f(\boldsymbol{w}^\star)$ is the unique solution to the dual problem, a natural choice of dual variable is $\boldsymbol{z} = \min\big(1, \frac{\lambda}{\Omega^*(\nabla f(\boldsymbol{w}))}\big)\nabla f(\boldsymbol{w})$, which reduces to $\boldsymbol{z}(\boldsymbol{w}^\star)$ at the optimum and therefore yields a zero duality gap at optimality.

Note that in most formulations that we will consider, the function $f$ is of the form $f(\boldsymbol{w}) = \psi(\boldsymbol{X}\boldsymbol{w})$ with $\psi\colon \mathbb{R}^n \to \mathbb{R}$ and $\boldsymbol{X} \in \mathbb{R}^{n\times p}$ a design matrix. Indeed, this corresponds to linear prediction on $\mathbb{R}^p$, given $n$ observations $\boldsymbol{x}_i$, $i = 1, \ldots, n$, and the predictions $\boldsymbol{X}\boldsymbol{w} = (\boldsymbol{w}^\top \boldsymbol{x}_i)_{i=1,\ldots,n}$. Typically, the Fenchel conjugate of $\psi$ is easy to compute[7] while the design matrix $\boldsymbol{X}$ makes it hard[8] to compute $f^*$. In that case, Equation (1.1) can be rewritten as

$$\min_{\boldsymbol{w}\in\mathbb{R}^p, \boldsymbol{u}\in\mathbb{R}^n} \psi(\boldsymbol{u}) + \lambda\, \Omega(\boldsymbol{w}), \quad \text{s.t. } \boldsymbol{u} = \boldsymbol{X}\boldsymbol{w}, \tag{1.12}$$

and equivalently as the optimization of the Lagrangian

$$\begin{aligned} &\min_{\boldsymbol{w}\in\mathbb{R}^p, \boldsymbol{u}\in\mathbb{R}^n} \max_{\boldsymbol{\alpha}\in\mathbb{R}^n} \psi(\boldsymbol{u}) + \lambda\Omega(\boldsymbol{w}) + \lambda\boldsymbol{\alpha}^\top(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{u}), \\ &\min_{\boldsymbol{w}\in\mathbb{R}^p, \boldsymbol{u}\in\mathbb{R}^n} \max_{\boldsymbol{\alpha}\in\mathbb{R}^n} (\psi(\boldsymbol{u}) - \lambda\boldsymbol{\alpha}^\top\boldsymbol{u}) + \lambda(\Omega(\boldsymbol{w}) + \boldsymbol{\alpha}^\top\boldsymbol{X}\boldsymbol{w}), \end{aligned} \tag{1.13}$$

which is obtained by introducing the Lagrange multiplier $\boldsymbol{\alpha}$ for the constraint $\boldsymbol{u} = \boldsymbol{X}\boldsymbol{w}$. The corresponding Fenchel dual[9] is then

$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^n} -\psi^*(\lambda\boldsymbol{\alpha}) \quad \text{such that } \Omega^*(\boldsymbol{X}^\top\boldsymbol{\alpha}) \leq 1, \tag{1.14}$$

which does not require any inversion of $\boldsymbol{X}^\top\boldsymbol{X}$ (which would be required for computing the Fenchel conjugate of $f$). Thus, given a candidate $\boldsymbol{w}$, we consider $\boldsymbol{\alpha} = \min\big(1, \frac{\lambda}{\Omega^*(\boldsymbol{X}^\top\nabla\psi(\boldsymbol{X}\boldsymbol{w}))}\big)\nabla\psi(\boldsymbol{X}\boldsymbol{w})$, and can get

[7] For the least-squares loss with output vector $\boldsymbol{y} \in \mathbb{R}^n$, we have $\psi(\boldsymbol{u}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{u}\|_2^2$ and $\psi^*(\boldsymbol{\beta}) = \frac{1}{2}\|\boldsymbol{\beta}\|_2^2 + \boldsymbol{\beta}^\top\boldsymbol{y}$. For the logistic loss, we have $\psi(\boldsymbol{u}) = \sum_{i=1}^n \log(1 + \exp(-\boldsymbol{y}_i\boldsymbol{u}_i))$ and $\psi^*(\boldsymbol{\beta}) = \sum_{i=1}^n (1 + \boldsymbol{\beta}_i\boldsymbol{y}_i)\log(1 + \boldsymbol{\beta}_i\boldsymbol{y}_i) - \boldsymbol{\beta}_i\boldsymbol{y}_i\log(-\boldsymbol{\beta}_i\boldsymbol{y}_i)$ if $\forall i,\ -\boldsymbol{\beta}_i\boldsymbol{y}_i \in [0,1]$ and $+\infty$ otherwise.

[8] It would require to compute the pseudo-inverse of $\boldsymbol{X}$.

[9] Fenchel conjugacy naturally extends to this case; see Theorem 3.3.5 in [20] for more details.

an upper bound on optimality using primal (1.12) and dual (1.14) problems. Concrete examples of such duality gaps for various sparse regularized problems are presented in Appendix D of [85], and are implemented in the open-source software SPAMS,[10] which we have used in the experimental section of this monograph.

### 1.4.2 Quadratic Variational Formulation of Norms

Several variational formulations are associated with norms, the most natural one being the one that results directly from (1.6) applied to the dual norm:

$$\Omega(\boldsymbol{w}) = \max_{\boldsymbol{z}\in\mathbb{R}^p} \boldsymbol{w}^\top \boldsymbol{z} \quad \text{s.t.} \quad \Omega^*(\boldsymbol{z}) \leq 1.$$

However, another type of variational form is quite useful, especially for sparsity-inducing norms; among other purposes, as it is obtained by a variational upper-bound (as opposed to a lower-bound in the equation above), it leads to a general algorithmic scheme for learning problems regularized with this norm, in which the difficulties associated with optimizing the loss and that of optimizing the norm are partially decoupled. We present it in Section 5. We introduce this variational form first for the $\ell_1$- and $\ell_1/\ell_2$-norms and subsequently generalize it to norms that we call *subquadratic norms*.

**The case of the $\ell_1$- and $\ell_1/\ell_2$-norms.** The two basic variational identities we use are, for $a, b > 0$,

$$2ab = \inf_{\eta\in\mathbb{R}_+^*} \eta^{-1} a^2 + \eta\, b^2, \tag{1.15}$$

where the infimum is attained at $\eta = a/b$, and, for $\boldsymbol{a} \in \mathbb{R}_+^p$,

$$\left(\sum_{i=1}^p \boldsymbol{a}_i\right)^2 = \inf_{\boldsymbol{\eta}\in(\mathbb{R}_+^*)^p} \sum_{i=1}^p \frac{\boldsymbol{a}_i^2}{\boldsymbol{\eta}_i} \text{ s.t. } \sum_{i=1}^p \boldsymbol{\eta}_i = 1. \tag{1.16}$$

The last identity is a direct consequence of the Cauchy–Schwarz inequality:

$$\sum_{i=1}^p \boldsymbol{a}_i = \sum_{i=1}^p \frac{\boldsymbol{a}_i}{\sqrt{\boldsymbol{\eta}_i}} \cdot \sqrt{\boldsymbol{\eta}_i} \leq \left(\sum_{i=1}^p \frac{\boldsymbol{a}_i^2}{\boldsymbol{\eta}_i}\right)^{1/2} \left(\sum_{i=1}^p \boldsymbol{\eta}_i\right)^{1/2}. \tag{1.17}$$

[10] `http://www.di.ens.fr/willow/SPAMS/`.

The infima in the previous expressions can be replaced by a minimization if the function $q{:}\,\mathbb{R}\times\mathbb{R}_+\to\mathbb{R}_+$ with $q(x,y)=\frac{x^2}{y}$ is extended in $(0,0)$ using the convention "0/0=0", since the resulting function[11] is a proper closed convex function. We will use this convention implicitly from now on. The minimum is then attained when equality holds in the Cauchy–Schwarz inequality, that is for $\sqrt{\boldsymbol{\eta}_i}\propto \boldsymbol{a}_i/\sqrt{\boldsymbol{\eta}_i}$, which leads to $\boldsymbol{\eta}_i=\frac{\boldsymbol{a}_i}{\|\boldsymbol{a}\|_1}$ if $\boldsymbol{a}\neq 0$ and 0 else.

Introducing the simplex $\triangle_p=\{\boldsymbol{\eta}\in\mathbb{R}^p_+ \mid \sum_{i=1}^p \boldsymbol{\eta}_i=1\}$, we apply these variational forms to the $\ell_1$- and $\ell_1/\ell_2$-norms (with nonoverlapping groups) with $\|\boldsymbol{w}\|_{\ell_1/\ell_2}=\sum_{g\in\mathcal{G}}\|\boldsymbol{w}_g\|_2$ and $|\mathcal{G}|=m$, so that we obtain directly:

$$\|\boldsymbol{w}\|_1=\min_{\boldsymbol{\eta}\in\mathbb{R}^p_+}\frac{1}{2}\sum_{i=1}^p\left[\frac{\boldsymbol{w}_i^2}{\boldsymbol{\eta}_i}+\boldsymbol{\eta}_i\right],\qquad \|\boldsymbol{w}\|_1^2=\min_{\boldsymbol{\eta}\in\triangle_p}\sum_{i=1}^p\frac{\boldsymbol{w}_i^2}{\boldsymbol{\eta}_i},$$

$$\|\boldsymbol{w}\|_{\ell_1/\ell_2}=\min_{\boldsymbol{\eta}\in\mathbb{R}^m_+}\frac{1}{2}\sum_{g\in\mathcal{G}}\left[\frac{\|\boldsymbol{w}_g\|_2^2}{\boldsymbol{\eta}^g}+\boldsymbol{\eta}^g\right],\quad \|\boldsymbol{w}\|_{\ell_1/\ell_2}^2=\min_{\boldsymbol{\eta}\in\triangle_m}\sum_{g\in\mathcal{G}}\frac{\|\boldsymbol{w}_g\|_2^2}{\boldsymbol{\eta}^g}.$$

**Quadratic variational forms for subquadratic norms.** The variational form of the $\ell_1$-norm admits a natural generalization for certain norms that we call *subquadratic* norms. Before we introduce them, we review a few useful properties of norms. In this section, we will denote $|\boldsymbol{w}|$ the vector $(|\boldsymbol{w}_1|,\dots,|\boldsymbol{w}_p|)$.

---

**Definition 1.1 (Absolute and monotonic norm).** We say that:

- A norm $\Omega$ is **absolute** if for all $v\in\mathbb{R}^p$, $\Omega(\boldsymbol{v})=\Omega(|\boldsymbol{v}|)$.
- A norm $\Omega$ is **monotonic** if for all $\boldsymbol{v},\boldsymbol{w}\in\mathbb{R}^p$ s.t. $|\boldsymbol{v}_i|\leq|\boldsymbol{w}_i|$, $i=1,\dots,p$, it holds that $\Omega(\boldsymbol{v})\leq\Omega(\boldsymbol{w})$.

---

These definitions are in fact equivalent (see, e.g., [16]):

---

**Proposition 1.7.** A norm is *monotonic* if and only if it is *absolute*.

---

[11] This extension is in fact the function $\tilde{q}{:}\,(x,y)\mapsto\min\left\{t\in\mathbb{R}_+ \mid \begin{bmatrix}t & x\\ x & y\end{bmatrix}\succeq 0\right\}$.

*Proof.* If $\Omega$ is monotonic, the fact that $\big|\boldsymbol{v}\big| = \big||\boldsymbol{v}|\big|$ implies $\Omega(\boldsymbol{v}) = \Omega(|\boldsymbol{v}|)$ so that $\Omega$ is absolute.

If $\Omega$ is absolute, we first show that $\Omega^*$ is absolute. Indeed,

$$\Omega^*(\boldsymbol{\kappa}) = \max_{\boldsymbol{w}\in\mathbb{R}^p,\ \Omega(|\boldsymbol{w}|)\leq 1} \boldsymbol{w}^\top\boldsymbol{\kappa} = \max_{\boldsymbol{w}\in\mathbb{R}^p,\ \Omega(|\boldsymbol{w}|)\leq 1} |\boldsymbol{w}|^\top|\boldsymbol{\kappa}| = \Omega^*(|\boldsymbol{\kappa}|).$$

Then if $|\boldsymbol{v}| \leq |\boldsymbol{w}|$, since $\Omega^*(\boldsymbol{\kappa}) = \Omega^*(|\boldsymbol{\kappa}|)$,

$$\Omega(\boldsymbol{v}) = \max_{\boldsymbol{\kappa}\in\mathbb{R}^p,\ \Omega^*(|\boldsymbol{\kappa}|)\leq 1} |\boldsymbol{v}|^\top|\boldsymbol{\kappa}| \leq \max_{\boldsymbol{\kappa}\in\mathbb{R}^p,\ \Omega^*(|\boldsymbol{\kappa}|)\leq 1} |\boldsymbol{w}|^\top|\boldsymbol{\kappa}| = \Omega(\boldsymbol{w}),$$

which shows that $\Omega$ is monotonic. □

We now introduce a family of norms, which have recently been studied in [93].

---

**Definition 1.2 ($H$-norm).** Let $H$ be a compact convex subset of $\mathbb{R}^p_+$, such that $H \cap (\mathbb{R}^*_+)^p \neq \varnothing$, we say that $\Omega_H$ is an $H$-norm if $\Omega_H(\boldsymbol{w}) = \min_{\boldsymbol{\eta}\in H} \sum_{i=1}^p \frac{\boldsymbol{w}_i^2}{\boldsymbol{\eta}_i}$.

---

The next proposition shows that $\Omega_H$ is indeed a norm and characterizes its dual norm.

---

**Proposition 1.8.** $\Omega_H$ is a norm and $\Omega_H^*(\boldsymbol{\kappa})^2 = \max_{\boldsymbol{\eta}\in H} \sum_{i=1}^p \boldsymbol{\eta}_i \boldsymbol{\kappa}_i^2$.

---

*Proof.* First, since $H$ contains at least one element whose components are all strictly positive, $\Omega$ is finite on $\mathbb{R}^p$. Symmetry, nonnegativity and homogeneity of $\Omega_H$ are straightforward from the definitions. Definiteness results from the fact that $H$ is bounded. $\Omega_H$ is convex, since it is obtained by minimization of $\boldsymbol{\eta}$ in a jointly convex formulation. Thus $\Omega_H$ is a norm. Finally,

$$\begin{aligned}\frac{1}{2}\Omega_H^*(\boldsymbol{\kappa})^2 &= \max_{\boldsymbol{w}\in\mathbb{R}^p} \boldsymbol{w}^\top\boldsymbol{\kappa} - \frac{1}{2}\Omega_H(\boldsymbol{w})^2 \\ &= \max_{\boldsymbol{w}\in\mathbb{R}^p} \max_{\boldsymbol{\eta}\in H} \boldsymbol{w}^\top\boldsymbol{\kappa} - \frac{1}{2}\boldsymbol{w}^\top \mathrm{Diag}(\boldsymbol{\eta})^{-1}\boldsymbol{w}.\end{aligned}$$

The form of the dual norm follows by maximizing w.r.t. $\boldsymbol{w}$. □

We finally introduce the family of norms that we call *subquadratic*.

**Definition 1.3 (Subquadratic norm).** Let $\Omega$ and $\Omega^*$ a pair of *absolute* dual norms. Let $\bar{\Omega}^*$ be the function defined as $\bar{\Omega}^*\colon \boldsymbol{\kappa} \mapsto [\Omega^*(|\boldsymbol{\kappa}|^{1/2})]^2$ where we use the notation $|\boldsymbol{\kappa}|^{1/2} = (|\boldsymbol{\kappa}_1|^{1/2}, \ldots, |\boldsymbol{\kappa}_p|^{1/2})^\top$. We say that $\Omega$ is *subquadratic* if $\bar{\Omega}^*$ is convex.

With this definition, we have:

**Lemma 1.9.** If $\Omega$ is *subquadratic*, then $\bar{\Omega}^*$ is a norm, and denoting $\bar{\Omega}$ the dual norm of the latter, we have:

$$\Omega(\boldsymbol{w}) = \frac{1}{2} \min_{\boldsymbol{\eta} \in \mathbb{R}^p_+} \sum_i \frac{\boldsymbol{w}_i^2}{\boldsymbol{\eta}_i} + \bar{\Omega}(\boldsymbol{\eta})$$

$$\Omega(\boldsymbol{w})^2 = \min_{\boldsymbol{\eta} \in H} \sum_i \frac{\boldsymbol{w}_i^2}{\boldsymbol{\eta}_i} \quad \text{where } H = \{\boldsymbol{\eta} \in \mathbb{R}^p_+ \mid \bar{\Omega}(\boldsymbol{\eta}) \leq 1\}.$$

*Proof.* Note that by construction, $\bar{\Omega}^*$ is homogeneous, symmetric and definite ($\bar{\Omega}^*(\boldsymbol{\kappa}) = 0 \Rightarrow \boldsymbol{\kappa} = 0$). If $\bar{\Omega}^*$ is convex then $\bar{\Omega}^*(\frac{1}{2}(\boldsymbol{v} + \boldsymbol{u})) \leq \frac{1}{2}\big(\bar{\Omega}^*(\boldsymbol{v}) + \bar{\Omega}^*(\boldsymbol{u})\big)$, which by homogeneity shows that $\bar{\Omega}^*$ also satisfies the triangle inequality. Together, these properties show that $\bar{\Omega}^*$ is a norm. To prove the first identity we have, applying (1.15), and since $\Omega$ is absolute,

$$\begin{aligned}
\Omega(\boldsymbol{w}) &= \max_{\boldsymbol{\kappa} \in \mathbb{R}^p_+} \boldsymbol{\kappa}^\top |\boldsymbol{w}|, \quad \text{s.t. } \Omega^*(\boldsymbol{\kappa}) \leq 1 \\
&= \max_{\boldsymbol{\kappa} \in \mathbb{R}^p_+} \sum_{i=1}^p \boldsymbol{\kappa}_i^{1/2} |\boldsymbol{w}_i|, \quad \text{s.t. } \Omega^*(\boldsymbol{\kappa}^{1/2})^2 \leq 1 \\
&= \max_{\boldsymbol{\kappa} \in \mathbb{R}^p_+} \min_{\boldsymbol{\eta} \in \mathbb{R}^p_+} \frac{1}{2} \sum_{i=1}^p \frac{\boldsymbol{w}_i^2}{\boldsymbol{\eta}_i} + \boldsymbol{\kappa}^\top \boldsymbol{\eta}, \quad \text{s.t. } \bar{\Omega}^*(\boldsymbol{\kappa}) \leq 1 \\
&= \min_{\boldsymbol{\eta} \in \mathbb{R}^p_+} \max_{\boldsymbol{\kappa} \in \mathbb{R}^p_+} \frac{1}{2} \sum_{i=1}^p \frac{\boldsymbol{w}_i^2}{\boldsymbol{\eta}_i} + \boldsymbol{\kappa}^\top \boldsymbol{\eta}, \quad \text{s.t. } \bar{\Omega}^*(\boldsymbol{\kappa}) \leq 1,
\end{aligned}$$

which proves the first variational formulation (note that we can switch the order of the max and min operations because strong duality holds,

which is due to the non-emptiness of the unit ball of the dual norm). The second one follows similarly by applying (1.16) instead of (1.15).

$$\begin{aligned}
\Omega(\boldsymbol{w})^2 &= \max_{\boldsymbol{\kappa}\in\mathbb{R}^p_+} \left(\sum_{i=1}^p \boldsymbol{\kappa}_i^{1/2}|\boldsymbol{w}_i|\right)^2, \quad \text{s.t. } \Omega^*(\boldsymbol{\kappa}^{1/2})^2 \leq 1 \\
&= \max_{\boldsymbol{\kappa}\in\mathbb{R}^p_+} \min_{\tilde{\boldsymbol{\eta}}\in\mathbb{R}^p_+} \sum_{i=1}^p \frac{\boldsymbol{\kappa}_i \boldsymbol{w}_i^2}{\tilde{\boldsymbol{\eta}}_i}, \quad \text{s.t. } \sum_{i=1}^p \tilde{\boldsymbol{\eta}}_i = 1,\ \bar{\Omega}^*(\boldsymbol{\kappa}) \leq 1 \\
&= \max_{\boldsymbol{\kappa}\in\mathbb{R}^p_+} \min_{\boldsymbol{\eta}\in\mathbb{R}^p_+} \sum_{i=1}^p \frac{\boldsymbol{w}_i^2}{\boldsymbol{\eta}_i}, \quad \text{s.t. } \boldsymbol{\eta}^\top \boldsymbol{\kappa} = 1,\ \bar{\Omega}^*(\boldsymbol{\kappa}) \leq 1. \qquad \square
\end{aligned}$$

Thus, given a subquadratic norm, we may define a convex set $H$, namely the intersection of the unit ball of $\bar{\Omega}$ with the positive orthant $\mathbb{R}^p_+$, such that $\Omega(\boldsymbol{w})^2 = \min_{\boldsymbol{\eta}\in H} \sum_{i=1}^p \frac{\boldsymbol{w}_i^2}{\boldsymbol{\eta}_i}$, i.e., a subquadratic norm is an $H$-norm. We now show that these two properties are in fact equivalent.

---

**Proposition 1.10.** $\Omega$ is *subquadratic* if and only if it is an $H$-norm.

---

*Proof.* The previous lemma shows that subquadratic norms are $H$-norms. Conversely, let $\Omega_H$ be an $H$-norm. By construction, $\Omega_H$ is absolute, and as a result of Proposition 1.8, $\bar{\Omega}^*_H(\boldsymbol{w}) = \big(\Omega^*_H(|\boldsymbol{w}|^{1/2})\big)^2 = \max_{\boldsymbol{\eta}\in H} \sum_i \boldsymbol{\eta}_i |\boldsymbol{w}_i|$, which shows that $\bar{\Omega}^*_H$ is a convex function, as a maximum of convex functions. $\square$

It should be noted that the set $H$ leading to a given $H$-norm $\Omega_H$ is not unique; in particular $H$ is not necessarily the intersection of the unit ball of a norm with the positive orthant. Indeed, for the $\ell_1$-norm, we can take $H$ to be the unit simplex.

---

**Proposition 1.11.** Given a convex compact set $H$, let $\Omega_H$ be the associated $H$-norm and $\bar{\Omega}_H$ as defined in Lemma 1.9. Define the mirror image of $H$ as the set $\mathrm{Mirr}(H) = \{\boldsymbol{v} \in \mathbb{R}^p \mid |\boldsymbol{v}| \in H\}$ and denote the convex hull of a set $S$ by $\mathrm{Conv}(S)$. Then the unit ball of $\bar{\Omega}_H$ is $\mathrm{Conv}(\mathrm{Mirr}(H))$.

---

*Proof.* By construction:

$$\begin{aligned}\bar{\Omega}^*_H(\boldsymbol{\kappa}) &= \Omega^*_H(|\boldsymbol{\kappa}|^{1/2})^2 = \max_{\boldsymbol{\eta}\in H} \boldsymbol{\eta}^\top |\boldsymbol{\kappa}| \\ &= \max_{|\boldsymbol{w}|\in H} \boldsymbol{w}^\top \boldsymbol{\kappa} = \max_{\boldsymbol{w}\in \mathrm{Conv}(\mathrm{Mirr}(H))} \boldsymbol{w}^\top \boldsymbol{\kappa},\end{aligned}$$

since the maximum of a convex function over a convex set is attained at its extreme points. But $C = \mathrm{Conv}(\mathrm{Mirr}(H))$ is by construction a centrally symmetric convex set, which is bounded and closed like $H$, and whose interior contains 0 since $H$ contains at least one point whose components are strictly positive. This implies by Theorem 15.2 in [115] that $C$ is the unit ball of a norm (namely $\boldsymbol{x} \mapsto \inf\{\lambda \in \mathbb{R}_+ \mid \boldsymbol{x} \in \lambda C\}$), which by duality has to be the unit ball of $\bar{\Omega}_H$. □

This proposition combined with the result of Lemma 1.9 therefore shows that if $\mathrm{Conv}(\mathrm{Mirr}(H)) = \mathrm{Conv}(\mathrm{Mirr}(H'))$ then $H$ and $H'$ define the same norm.

Several instances of the general variational form we considered in this section have appeared in the literature [70, 109, 110]. For norms that are not subquadratic, it is often the case that their dual norm is itself subquadratic, in which case symmetric variational forms can be obtained [2]. Finally, we show in Section 5 that all norms admit a quadratic variational form provided the bilinear form considered is allowed to be non-diagonal.

## 1.5 Multiple Kernel Learning

A seemingly unrelated problem in machine learning, the problem of *multiple kernel learning* is in fact intimately connected with sparsity-inducing norms by duality. It actually corresponds to the most natural extension of sparsity to reproducing kernel Hilbert spaces. We will show that for a large class of norms and, among them, many sparsity-inducing norms, there exists for each of them a corresponding multiple kernel learning scheme, and, vice-versa, each multiple kernel learning scheme defines a new norm.

The problem of kernel learning is a priori quite unrelated with parsimony. It emerges as a consequence of a convexity property of the

so-called "kernel trick", which we now describe. Consider a learning problem with $f(\boldsymbol{w}) = \psi(\boldsymbol{X}\boldsymbol{w})$. As seen before, this corresponds to linear predictions of the form $\boldsymbol{X}\boldsymbol{w} = (\boldsymbol{w}^\top \boldsymbol{x}_i)_{i=1,\dots,n}$. Assume that this learning problem is this time regularized by the square of the norm $\Omega$ (as shown in Section 1.2, this does not change the regularization properties), so the we have the following optimization problem:

$$\min_{\boldsymbol{w}\in\mathbb{R}^p} f(\boldsymbol{w}) + \frac{\lambda}{2}\Omega(\boldsymbol{w})^2. \tag{1.18}$$

As in Equation (1.12) we can introduce the linear constraint

$$\min_{\boldsymbol{u}\in\mathbb{R}^n, \boldsymbol{w}\in\mathbb{R}^p} \psi(\boldsymbol{u}) + \frac{\lambda}{2}\Omega(\boldsymbol{w})^2, \quad \text{s.t. } \boldsymbol{u} = \boldsymbol{X}\boldsymbol{w}, \tag{1.19}$$

and reformulate the problem as the saddle point problem

$$\min_{\boldsymbol{u}\in\mathbb{R}^n, \boldsymbol{w}\in\mathbb{R}^p} \max_{\boldsymbol{\alpha}\in\mathbb{R}^n} \psi(\boldsymbol{u}) + \frac{\lambda}{2}\Omega(\boldsymbol{w})^2 - \lambda\boldsymbol{\alpha}^\top(\boldsymbol{u} - \boldsymbol{X}\boldsymbol{w}). \tag{1.20}$$

Since the primal problem (1.19) is a convex problem with feasible linear constraints, it satisfies Slater's qualification conditions and the order of maximization and minimization can be exchanged:

$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^n} \min_{\boldsymbol{u}\in\mathbb{R}^n, \boldsymbol{w}\in\mathbb{R}^p} (\psi(\boldsymbol{u}) - \lambda\boldsymbol{\alpha}^\top\boldsymbol{u}) + \lambda\left(\frac{1}{2}\Omega(\boldsymbol{w})^2 + \boldsymbol{\alpha}^\top\boldsymbol{X}\boldsymbol{w}\right). \tag{1.21}$$

Now, the minimization in $\boldsymbol{u}$ and $\boldsymbol{w}$ can be performed independently. One property of norms is that the Fenchel conjugate of $\boldsymbol{w} \mapsto \frac{1}{2}\Omega(\boldsymbol{w})^2$ is $\boldsymbol{\kappa} \mapsto \frac{1}{2}\Omega^*(\boldsymbol{\kappa})^2$; this can be easily verified by finding the vector $\boldsymbol{w}$ achieving equality in the sequence of inequalities $\boldsymbol{\kappa}^\top\boldsymbol{w} \leq \Omega(\boldsymbol{w})\Omega^*(\boldsymbol{\kappa}) \leq \frac{1}{2}\big[\Omega(\boldsymbol{w})^2 + \Omega^*(\boldsymbol{\kappa})^2\big]$. As a consequence, the dual optimization problem is

$$\max_{\boldsymbol{\alpha}\in\mathbb{R}^n} -\psi^*(\lambda\boldsymbol{\alpha}) - \frac{\lambda}{2}\Omega^*(\boldsymbol{X}^\top\boldsymbol{\alpha})^2. \tag{1.22}$$

If $\Omega$ is the Euclidean norm (i.e., the $\ell_2$-norm) then the previous problem is simply

$$G(\boldsymbol{K}) := \max_{\boldsymbol{\alpha}\in\mathbb{R}^n} -\psi^*(\lambda\boldsymbol{\alpha}) - \frac{\lambda}{2}\boldsymbol{\alpha}^\top\boldsymbol{K}\boldsymbol{\alpha} \quad \text{with } \boldsymbol{K} = \boldsymbol{X}\boldsymbol{X}^\top. \tag{1.23}$$

Focussing on this last case, a few remarks are crucial:

(1) The dual problem depends on the design $\boldsymbol{X}$ only through the kernel matrix $\boldsymbol{K} = \boldsymbol{X}\boldsymbol{X}^\top \in \mathbb{R}^{n\times n}$.

(2) $G$ is a *convex* function of $\boldsymbol{K}$ (as a maximum of linear functions).
(3) The solutions $\boldsymbol{w}^\star$ and $\boldsymbol{\alpha}^\star$ to the primal and dual problems satisfy $\boldsymbol{w}^\star = \boldsymbol{X}^\top \boldsymbol{\alpha}^\star = \sum_{i=1}^n \boldsymbol{\alpha}_i^\star \boldsymbol{x}_i$.
(4) The exact same duality result applies for the generalization to $\boldsymbol{w}, \boldsymbol{x}_i \in \mathcal{H}$ for $\mathcal{H}$ a Hilbert space.

The first remark suggests a way to solve learning problems that are non-linear in the inputs $\boldsymbol{x}_i$: in particular consider a non-linear mapping $\phi$ which maps $\boldsymbol{x}_i$ to a high-dimensional $\phi(\boldsymbol{x}_i) \in \mathcal{H}$ with $\mathcal{H} = \mathbb{R}^d$ for $d \gg p$ or possibly an infinite dimensional Hilbert space. Then consider the problem (1.18) with now $f(\boldsymbol{w}) = \psi\big((\langle \boldsymbol{w}, \phi(\boldsymbol{x}_i)\rangle)_{i=1,\dots,n}\big)$, which is typically of the form of an empirical risk $f(\boldsymbol{w}) = \frac{1}{n}\sum_{i=1}^n \ell(y^{(i)}, \langle \boldsymbol{w}, \phi(\boldsymbol{x}_i)\rangle)$. It becomes high-dimensional to solve in the primal, while it is simply solved in the dual by choosing a kernel matrix with entries $\boldsymbol{K}_{i,j} = \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j)\rangle$, which is advantageous as soon as $n^2 \leq d$; this is the so-called "kernel trick" (see more details in [122, 126]).

In particular, if we consider functions $h \in \mathcal{H}$ where $\mathcal{H}$ is a reproducing kernel Hilbert space (RKHS) with reproducing kernel $K$ then

$$\min_{h \in \mathcal{H}} \psi\big((h(\boldsymbol{x}_i))_{i=1,\dots,n}\big) + \frac{\lambda}{2}\|h\|_{\mathcal{H}}^2 \tag{1.24}$$

is solved by solving Equation (1.23) with $\boldsymbol{K}_{i,j} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$. When applied to the mapping $\phi : \boldsymbol{x} \mapsto K(\boldsymbol{x}, \cdot)$, the third remark above yields a specific version of the representer theorem of Kimmeldorf and Wahba [74][12] stating that $h^\star(\cdot) = \sum_{i=1}^n \boldsymbol{\alpha}_i^\star K(\boldsymbol{x}_i, \cdot)$. In this case, the predictions may be written equivalently as $h(\boldsymbol{x}_i)$ or $\langle \boldsymbol{w}, \phi(\boldsymbol{x}_i)\rangle$, $i = 1, \dots, n$.

As shown in [77], the fact that $G$ is a convex function of $\boldsymbol{K}$ suggests the possibility of optimizing the objective with respect to the choice of the kernel itself by solving a problem of the form $\min_{\boldsymbol{K} \in \mathcal{K}} G(\boldsymbol{K})$ where $\mathcal{K}$ is a convex set of kernel matrices.

In particular, given a finite set of kernel functions $(K_i)_{1 \leq i \leq p}$ it is natural to consider to find the best *linear* combination of kernels, which

[12] Note that this provides a proof of the representer theorem for *convex* losses only and that the parameters $\boldsymbol{\alpha}$ are obtained through a dual *maximization* problem.

requires to add a positive definiteness constraint on the kernel, leading to a semi-definite program [77]:

$$\min_{\boldsymbol{\eta}\in\mathbb{R}^p} G\left(\sum_{i=1}^p \boldsymbol{\eta}_i \boldsymbol{K}_i\right), \quad \text{s.t.} \quad \sum_{i=1}^p \eta_i \boldsymbol{K}_i \succeq 0, \quad \text{tr}\left(\sum_{i=1}^p \boldsymbol{\eta}_i \boldsymbol{K}_i\right) \leq 1. \tag{1.25}$$

Assuming that the kernels have equal trace, the two constraints of the previous program are avoided by considering convex combinations of kernels, which leads to a quadratically constrained quadratic program (QCQP) [78]:

$$\min_{\boldsymbol{\eta}\in\mathbb{R}^p_+} G\left(\sum_{i=1}^p \boldsymbol{\eta}_i \boldsymbol{K}_i\right), \text{ s.t. } \sum_{i=1}^p \boldsymbol{\eta}_i = 1. \tag{1.26}$$

We now present a reformulation of Equation (1.26) using sparsity-inducing norms (see [7, 13, 110] for more details).

### 1.5.1 From $\ell_1/\ell_2$-Regularization to MKL

As we presented it above, MKL arises from optimizing the objective of a learning problem w.r.t. to a convex combination of kernels, in the context of plain $\ell_2$- or Hilbert norm regularization, which seems a priori unrelated to sparsity. We will show in this section that, in fact, the primal problem corresponding exactly to MKL (i.e., Equation 1.26) is an $\ell_1/\ell_2$-regularized problem (with the $\ell_1/\ell_2$-norm defined in Equation (1.2)), in the sense that its dual is the MKL problem for the set of kernels associated with each of the groups of variables. The proof to establish the relation between the two relies on the variational formulation presented in Section 1.4.2.

We indeed have, assuming that $\mathcal{G}$ is a partition of $\{1,\ldots,p\}$, with $|\mathcal{G}| = m$, and $\triangle_m$ denoting the simplex in $\mathbb{R}^m$,

$$\begin{aligned}
&\min_{\boldsymbol{w}\in\mathbb{R}^p} \psi(\boldsymbol{X}\boldsymbol{w}) + \frac{\lambda}{2}\left(\sum_{g\in\mathcal{G}} \|\boldsymbol{w}_g\|_2\right)^2 \\
&\quad = \min_{\boldsymbol{w}\in\mathbb{R}^p, \boldsymbol{\eta}\in\triangle_m} \psi(\boldsymbol{X}\boldsymbol{w}) + \frac{\lambda}{2}\sum_{g\in\mathcal{G}} \frac{\|\boldsymbol{w}_g\|_2^2}{\boldsymbol{\eta}_g}
\end{aligned}$$

$$
\begin{aligned}
&= \min_{\widetilde{\boldsymbol{w}} \in \mathbb{R}^p, \boldsymbol{\eta} \in \triangle_m} \psi\left(\sum_{g \in \mathcal{G}} \boldsymbol{\eta}_g^{1/2} \boldsymbol{X}_g \widetilde{\boldsymbol{w}}_g\right) + \frac{\lambda}{2} \sum_{g \in \mathcal{G}} \|\widetilde{\boldsymbol{w}}_g\|_2^2 \\
&= \min_{\widetilde{\boldsymbol{w}} \in \mathbb{R}^p, \boldsymbol{\eta} \in \triangle_m} \psi(\widetilde{\boldsymbol{X}} \widetilde{\boldsymbol{w}}) + \frac{\lambda}{2} \|\widetilde{\boldsymbol{w}}\|_2^2, \text{ s.t. } \widetilde{\boldsymbol{X}} = [\eta_{g_1}^{1/2} \boldsymbol{X}_{g_1}, \ldots, \eta_{g_m}^{1/2} \boldsymbol{X}_{g_m}] \\
&= \min_{\boldsymbol{\eta} \in \triangle_m} \max_{\boldsymbol{\alpha} \in \mathbb{R}^n} -\psi^*(\lambda \boldsymbol{\alpha}) - \frac{\lambda}{2} \boldsymbol{\alpha}^\top \left(\sum_{g \in \mathcal{G}} \boldsymbol{\eta}_g \boldsymbol{K}_g\right) \boldsymbol{\alpha} \\
&= \min_{\boldsymbol{\eta} \in \triangle_m} G\left(\sum_{g \in \mathcal{G}} \boldsymbol{\eta}_g \boldsymbol{K}_g\right),
\end{aligned}
$$

where the third line results from the change of variable $\widetilde{\boldsymbol{w}}_g \boldsymbol{\eta}_g^{1/2} = \boldsymbol{w}_g$, and the last step from the definition of $G$ in Equation (1.23).

Note that $\ell_1$-regularization corresponds to the special case where groups are singletons and where $\boldsymbol{K}_i = \boldsymbol{x}_i \boldsymbol{x}_i^\top$ is a rank-one kernel matrix. In other words, MKL with rank-one kernel matrices (i.e., feature spaces of dimension one) is equivalent to $\ell_1$-regularization (and thus simpler algorithms can be brought to bear in this situation).

We have shown that learning convex combinations of kernels through Equation (1.26) turns out to be equivalent to an $\ell_1/\ell_2$-norm penalized problems. In other words, learning a linear combination $\sum_{i=1}^m \boldsymbol{\eta}_i \boldsymbol{K}_i$ of kernel matrices, subject to $\boldsymbol{\eta}$ belonging to the simplex $\triangle_m$ is equivalent to penalizing the empirical risk with an $\ell_1$-norm applied to norms of predictors $\|\boldsymbol{w}_g\|_2$. This link between the $\ell_1$-norm and the simplex may be extended to other norms, among others to the subquadratic norms introduced in Section 1.4.2.

### 1.5.2 Structured Multiple Kernel Learning

In the relation established between $\ell_1/\ell_2$-regularization and MKL in the previous section, the vector of weights $\boldsymbol{\eta}$ for the different kernels corresponded to the vector of optimal variational parameters defining the norm. A natural way to extend MKL is, instead of considering a convex combination of kernels, to consider a linear combination of the same kernels, but with positive weights satisfying a different set of constraints than the simplex constraints. Given the relation between

kernel weights and the variational form of a norm, we will be able to show that, for norms that have a variational form as in Lemma 1.8, we can generalize the correspondence between the $\ell_1/\ell_2$-norm and MKL to a correspondence between other structured norms and structured MKL schemes.

Using the same line of proof as in the previous section, and given an $H$-norm (or equivalently a subquadratic norm) $\Omega_H$ as defined in Definition 1.2, we have:

$$\begin{aligned}
&\min_{\boldsymbol{w}\in\mathbb{R}^p} \psi(\boldsymbol{X}\boldsymbol{w}) + \frac{\lambda}{2}\Omega_H(\boldsymbol{w})^2 \\
&\quad= \min_{\boldsymbol{w}\in\mathbb{R}^p,\boldsymbol{\eta}\in H} \psi(\boldsymbol{X}\boldsymbol{w}) + \frac{\lambda}{2}\sum_{i=1}^p \frac{\boldsymbol{w}_i^2}{\boldsymbol{\eta}_i} \\
&\quad= \min_{\widetilde{\boldsymbol{w}}\in\mathbb{R}^p,\boldsymbol{\eta}\in H} \psi\left(\sum_{i=1}^p \boldsymbol{\eta}_i^{1/2}\boldsymbol{X}_i\widetilde{\boldsymbol{w}}_i\right) + \frac{\lambda}{2}\sum_{i=1}^p \widetilde{\boldsymbol{w}}_i^2 \\
&\quad= \min_{\widetilde{\boldsymbol{w}}\in\mathbb{R}^p,\boldsymbol{\eta}\in H} \psi(\widetilde{\boldsymbol{X}}\widetilde{\boldsymbol{w}}) + \frac{\lambda}{2}\|\widetilde{\boldsymbol{w}}\|_2^2, \text{ s.t. } \widetilde{\boldsymbol{X}} = [\boldsymbol{\eta}_1^{1/2}\boldsymbol{X}_1,\ldots,\boldsymbol{\eta}_p^{1/2}\boldsymbol{X}_p] \\
&\quad= \min_{\boldsymbol{\eta}\in H}\max_{\boldsymbol{\alpha}\in\mathbb{R}^n} -\psi^*(\lambda\boldsymbol{\alpha}) - \frac{\lambda}{2}\boldsymbol{\alpha}^\top\left(\sum_{i=1}^p \boldsymbol{\eta}_i\boldsymbol{K}_i\right)\boldsymbol{\alpha} \\
&\quad= \min_{\boldsymbol{\eta}\in H} G\left(\sum_{i=1}^p \boldsymbol{\eta}_i\boldsymbol{K}_i\right). \qquad (1.27)
\end{aligned}$$

This results shows that the regularization with the norm $\Omega_H$ in the primal is equivalent to a multiple kernel learning formulation in which the kernel weights are constrained to belong to the convex set $H$, which defines $\Omega_H$ variationally. Note that we have assumed that $H \subset \mathbb{R}^p_+$, so that formulations such as (1.25), where positive semidefiniteness of $\sum_{i=1}^p \boldsymbol{\eta}_i\boldsymbol{K}_i$ has to be added as a constraint, are not included.

Given the relationship of MKL to the problem of learning a function in a reproducing kernel Hilbert space, the previous result suggests a natural extension of structured sparsity to the RKHS settings. Indeed let, $h = (h_1,\ldots,h_p) \in \mathcal{B} := \mathcal{H}_1 \times \cdots \times \mathcal{H}_p$, where $\mathcal{H}_i$ are RKHSs. It is easy to verify that $\Lambda: h \mapsto \Omega_H\big((\|h_1\|_{\mathcal{H}_1},\ldots,\|h_p\|_{\mathcal{H}_p})\big)$ is a convex function, using the variational formulation of $\Omega_H$, and since it is also

non-negative definite and homogeneous, it is a norm.[13] Moreover, the learning problem obtained by summing the predictions from the different RKHSs, i.e.,

$$\min_{h\in\mathcal{B}} \psi((h_1(\boldsymbol{x}_i) + \cdots + h_p(\boldsymbol{x}_i))_{i=1,\ldots,n}) + \frac{\lambda}{2}\Omega_H((\|h_1\|_{\mathcal{H}_1}, \ldots, \|h_p\|_{\mathcal{H}_p}))^2 \tag{1.28}$$

is equivalent, by the above derivation, to the MKL problem $\min_{\boldsymbol{\eta}\in H} G(\sum_{i=1}^{p} \boldsymbol{\eta}_i \boldsymbol{K}_i)$ with $[\boldsymbol{K}_i]_{j,j'} = K_i(\boldsymbol{x}_j, \boldsymbol{x}_{j'})$ for $K_i$ the reproducing kernel of $\mathcal{H}_i$. See Section 3.4 for more details.

This means that, for most of the structured sparsity-inducing norms that we have considered in Section 1.3, we may replace individual variables by whole Hilbert spaces. For example, tree-structured sparsity (and its extension to directed acyclic graphs) was explored in [9] where each node of the graph was an RKHS, with an application to nonlinear variable selection.

[13] As we show in Section 3.4, it is actually sufficient to assume that $\Omega$ is monotonic for $\Lambda$ to be a norm.

# 2

# Generic Methods

The problem defined in Equation (1.1) is convex, as soon as both the loss $f$ and the regularizer $\Omega$ are convex functions. In this section, we consider optimization strategies which are essentially blind to problem structure. The first of these techniques is subgradient descent (see, e.g., [18]), which is widely applicable, has low running time complexity per iterations, but has a slow convergence rate. As opposed to proximal methods presented in Section 3.1, it does not use problem structure. At the other end of the spectrum, the second strategy is to consider reformulations such as linear programs (LP), quadratic programs (QP) or more generally, second-order cone programming (SOCP) or semidefinite programming (SDP) problems (see, e.g., [25]). The latter strategy is usually only possible with the square loss and makes use of general-purpose optimization toolboxes. Moreover, these toolboxes are only adapted to small-scale problems and usually lead to solution with very high precision (low duality gap), while simpler iterative methods can be applied to large-scale problems but only leads to solution with low or medium precision, which is sufficient in most applications to machine learning (see [22] for a detailed discussion).

**Subgradient descent.** For all convex unconstrained problems, subgradient descent can be used as soon as one subgradient can be computed efficiently. In our setting, this is possible when a subgradient of the loss $f$, and a subgradient of the regularizer $\Omega$ can be computed. This is true for all the norms that we have considered. The corresponding algorithm consists of the following iterations:

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \frac{\alpha}{t^\beta}(\boldsymbol{s} + \lambda \boldsymbol{s}'), \quad \text{where } \boldsymbol{s} \in \partial f(\boldsymbol{w}_t), \ \ \boldsymbol{s}' \in \partial \Omega(\boldsymbol{w}_t),$$

with $\alpha$ a well-chosen positive parameter and $\beta$ typically 1 or 1/2. Under certain conditions, these updates are globally convergent. More precisely, we have, from [100], $F(\boldsymbol{w}_t) - \min_{\boldsymbol{w} \in \mathbb{R}^p} F(\boldsymbol{w}) = O(\frac{\log t}{\sqrt{t}})$ for Lipschitz-continuous function and $\beta = 1/2$. However, the convergence is in practice slow (i.e., many iterations are needed), and the solutions obtained are usually not sparse. This is to be contrasted with the proximal methods presented in the next section which are less generic but more adapted to sparse problems, with in particular convergence rates in $O(1/t)$ and $O(1/t^2)$.

**Reformulation as LP, QP, SOCP, SDP.** For all the sparsity-inducing norms we consider in this monograph the corresponding regularized least-square problem can be represented by standard mathematical programming problems, all of them being SDPs, and often simpler (e.g., QP). For example, for the $\ell_1$-norm regularized least-square regression, we can reformulate $\min_{\boldsymbol{w} \in \mathbb{R}^p} \frac{1}{2n} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 + \lambda \Omega(\boldsymbol{w})$ as

$$\min_{\boldsymbol{w}_+, \boldsymbol{w}_- \in \mathbb{R}_+^p} \frac{1}{2n} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}_+ + \boldsymbol{X}\boldsymbol{w}_-\|_2^2 + \lambda (1^\top \boldsymbol{w}_+ + 1^\top \boldsymbol{w}_-),$$

which is a quadratic program. Grouped norms with combinations of $\ell_2$-norms leads to an SOCP, i.e., $\min_{w \in \mathbb{R}^p} \frac{1}{2n} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 + \lambda \sum_{g \in \mathcal{G}} d_g \|\boldsymbol{w}_g\|_2$ may be formulated as

$$\min_{\boldsymbol{w} \in \mathbb{R}^p, \ (\boldsymbol{t}_g)_{g \in \mathcal{G}} \in \mathbb{R}_+^{|\mathcal{G}|}} \frac{1}{2n} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 + \lambda \sum_{g \in \mathcal{G}} d_g \boldsymbol{t}_g, \quad \text{s.t } \forall g \in \mathcal{G}, \ \|\boldsymbol{w}_g\|_2 \leq \boldsymbol{t}_g.$$

Other problems can be similarly cast (for the trace norm, see [8, 48]). General-purpose toolboxes can then be used, to get solutions with high

precision (low duality gap). However, in the context of machine learning, this is inefficient for two reasons: (1) these toolboxes are generic and blind to problem structure and tend to be too slow, or cannot even run because of memory problems, (2) as outlined in [22], high precision is not necessary for machine learning problems, and a duality gap of the order of machine precision (which would be a typical result from such toolboxes) is not necessary.

We present in the following sections methods that are adapted to problems regularized by sparsity-inducing norms.

# 3

# Proximal Methods

This section reviews a class of techniques referred to as *proximal methods*, where the nonsmooth component of the objective (1.1) will only be involved in the computations through an associated *proximal operator,* which we formally define subsequently.

The presentation that we make of proximal methods in this section is deliberately simplified, and to be rigorous the methods that we will refer to as proximal methods in this section are known as *forward–backward splitting* methods. We refer the interested reader to Section 9 for a broader view and references.

## 3.1 Principle of Proximal Methods

Proximal methods (i.e., forward-backward splitting methods) are specifically tailored to optimize an objective of the form (1.1), i.e., which can be written as the sum of a generic smooth differentiable function $f$ with Lipschitz-continuous gradient, and a nondifferentiable function $\lambda\Omega$.

They have drawn increasing attention in the machine learning community, especially because of their convergence rates and their ability to

deal with large nonsmooth convex problems (e.g., [17, 38, 102, 151]). Proximal methods can be described as follows: at each iteration the function $f$ is linearized around the current point and a problem of the form

$$\min_{\boldsymbol{w}\in\mathbb{R}^p} f(\boldsymbol{w}^t) + \nabla f(\boldsymbol{w}^t)^\top(\boldsymbol{w}-\boldsymbol{w}^t) + \lambda\Omega(\boldsymbol{w}) + \frac{L}{2}\|\boldsymbol{w}-\boldsymbol{w}^t\|_2^2 \qquad (3.1)$$

is solved. The quadratic term, called proximal term, keeps the update in a neighborhood of the current iterate $\boldsymbol{w}^t$ where $f$ is close to its linear approximation; $L>0$ is a parameter, which should essentially be an upper bound on the Lipschitz constant of $\nabla f$ and is typically set with a line-search. This problem can be rewritten as

$$\min_{\boldsymbol{w}\in\mathbb{R}^p} \frac{1}{2}\left\|\boldsymbol{w} - \left(\boldsymbol{w}^t - \frac{1}{L}\nabla f(\boldsymbol{w}^t)\right)\right\|_2^2 + \frac{\lambda}{L}\Omega(\boldsymbol{w}). \qquad (3.2)$$

It should be noted that when the nonsmooth term $\Omega$ is not present, the solution of the previous proximal problem, also known as the backward or implicit step, just yields the standard gradient update rule $\boldsymbol{w}^{t+1} \leftarrow \boldsymbol{w}^t - \frac{1}{L}\nabla f(\boldsymbol{w}^t)$. Furthermore, if $\Omega$ is the indicator function of a set $\iota_C$, i.e., defined by $\iota_C(x)=0$ for $x\in C$ and $\iota_C(x)=+\infty$ otherwise, then solving (3.2) yields the projected gradient update with projection on the set $C$. This suggests that the solution of the proximal problem provides an interesting generalization of gradient updates, and motivates the introduction of the notion of a *proximal operator* associated with the regularization term $\lambda\Omega$.

The proximal operator, which we will denote $\mathrm{Prox}_{\mu\Omega}$, was defined in [94] as the function that maps a vector $\boldsymbol{u}\in\mathbb{R}^p$ to the unique[1] solution of

$$\min_{\boldsymbol{w}\in\mathbb{R}^p} \frac{1}{2}\|\boldsymbol{u}-\boldsymbol{w}\|_2^2 + \mu\,\Omega(\boldsymbol{w}). \qquad (3.3)$$

This operator is clearly central to proximal methods since their main step consists in computing $\mathrm{Prox}_{\frac{\lambda}{L}\Omega}\big(\boldsymbol{w}^t - \frac{1}{L}\nabla f(\boldsymbol{w}^t)\big)$.

In Section 3.3, we present analytical forms of proximal operators associated with simple norms and algorithms to compute them in some more elaborate cases. Note that the proximal term in Equation (3.1)

[1] Since the objective is strongly convex.

could be replaced by any Bregman divergences (see, e.g., [139]), which may be useful in settings where extra constraints (such as non-negativity) are added to the problem.

## 3.2 Algorithms

The basic proximal algorithm uses the solution of problem (3.2) as the next update $\boldsymbol{w}^{t+1}$; however fast variants such as the accelerated algorithm presented in [102] or FISTA [17] maintain two variables and use them to combine at marginal extra computational cost the solution of (3.2) with information about previous steps. Often, an upper bound on the Lipschitz constant of $\nabla f$ is not known, and even if it is,[2] it is often better to obtain a local estimate. A suitable value for $L$ can be obtained by iteratively increasing $L$ by a constant factor until the condition

$$f(\boldsymbol{w}_L^\star) \leq f(\boldsymbol{w}^t) + \nabla f(\boldsymbol{w}^t)^\top (\boldsymbol{w}_L^\star - \boldsymbol{w}^t) + \frac{L}{2}\|\boldsymbol{w}_L^\star - \boldsymbol{w}^t\|_2^2 \tag{3.4}$$

is met, where $\boldsymbol{w}_L^\star$ denotes the solution of (3.3).

For functions $f$ whose gradients are Lipschitz-continuous, the basic proximal algorithm has a global convergence rate in $O(\frac{1}{t})$ where $t$ is the number of iterations of the algorithm. Accelerated algorithms like FISTA can be shown to have global convergence rate — *on the objective function* — in $O(\frac{1}{t^2})$, which has been proved to be optimal for the class of first-order techniques [100].

Note that, unlike for the simple proximal scheme, we cannot guarantee that the sequence of iterates generated by the accelerated version is itself convergent [38].

Perhaps more importantly, both basic (ISTA) and accelerated [102] proximal methods are adaptive in the sense that if $f$ is strongly convex — and the problem is therefore better conditioned — the convergence is actually linear (i.e., with rates in $O(C^t)$ for some constant $C < 1$; see [102]). Finally, it should be noted that accelerated schemes are not necessarily descent algorithms, in the sense that the objective

[2] For problems common in machine learning where $f(\boldsymbol{w}) = \psi(\boldsymbol{X}\boldsymbol{w})$ and $\psi$ is twice differentiable, then $L$ may be chosen to be the largest eigenvalue of $\frac{1}{n}\boldsymbol{X}^\top\boldsymbol{X}$ times the supremum over $\boldsymbol{u} \in \mathbb{R}^n$ of the largest eigenvalue of the Hessian of $\psi$ at $\boldsymbol{u}$.

does not necessarily decrease at each iteration in spite of the global convergence properties.

## 3.3 Computing the Proximal Operator

Computing the *proximal operator* efficiently and exactly allows to attain the fast convergence rates of proximal methods.[3] We therefore focus here on properties of this operator and on its computation for several sparsity-inducing norms. For a complete study of the properties of the proximal operator, we refer the interested reader to [39].

**Dual proximal operator.** In the case where $\Omega$ is a norm, by Fenchel duality, the following problem is dual (see Proposition 1.6) to problem (3.2):

$$\max_{\boldsymbol{v}\in\mathbb{R}^p} -\frac{1}{2}\left[\|\boldsymbol{v}-\boldsymbol{u}\|_2^2 - \|\boldsymbol{u}\|^2\right] \quad \text{such that } \Omega^*(\boldsymbol{v}) \leq \mu. \tag{3.5}$$

---

**Lemma 3.1(Relation to dual proximal operator).** Let $\text{Prox}_{\mu\Omega}$ be the proximal operator associated with the regularization $\mu\Omega$, where $\Omega$ is a norm, and $\text{Proj}_{\{\Omega^*(\cdot)\leq\mu\}}$ be the projector on the ball of radius $\mu$ of the dual norm $\Omega^*$. Then $\text{Proj}_{\{\Omega^*(\cdot)\leq\mu\}}$ is the proximal operator for the dual problem (3.5) and, denoting the identity $I_d$, these two operators satisfy the relation

$$\text{Prox}_{\mu\Omega} = I_d - \text{Proj}_{\{\Omega^*(\cdot)\leq\mu\}}. \tag{3.6}$$

---

*Proof.* By Proposition 1.6, if $\boldsymbol{w}^\star$ is optimal for (3.3) and $\boldsymbol{v}^\star$ is optimal for (3.5), we have[4] $-\boldsymbol{v}^\star = \nabla f(\boldsymbol{w}^\star) = \boldsymbol{w}^\star - \boldsymbol{u}$. Since $\boldsymbol{v}^\star$ is the projection of $\boldsymbol{u}$ on the ball of radius $\mu$ of the norm $\Omega^*$, the result follows. □

This lemma shows that the proximal operator can always be computed as the residual of a Euclidean projection onto a convex set. More general results appear in [39].

[3] Note, however, that fast convergence rates can also be achieved while solving approximately the proximal problem, as long as the precision of the approximation iteratively increases with an appropriate rate (see [120] for more details).

[4] The dual variable from Fenchel duality is $-\boldsymbol{v}$ in this case.

**$\ell_1$-norm regularization.** Using optimality conditions for (3.5) and then (3.6) or subgradient condition (1.7) applied to (3.3), it is easy to check that $\mathrm{Proj}_{\{\|\cdot\|_\infty \leq \mu\}}$ and $\mathrm{Prox}_{\mu\|\cdot\|_1}$ respectively satisfy:

$$\left[\mathrm{Proj}_{\{\|\cdot\|_\infty \leq \mu\}}(\boldsymbol{u})\right]_j = \min\left(1, \tfrac{\mu}{|\boldsymbol{u}_j|}\right) \boldsymbol{u}_j,$$

and

$$\left[\mathrm{Prox}_{\mu\|\cdot\|_1}(\boldsymbol{u})\right]_j = \left(1 - \frac{\mu}{|\boldsymbol{u}_j|}\right)_+ \boldsymbol{u}_j = \mathrm{sign}(\boldsymbol{u}_j)(|\boldsymbol{u}_j| - \mu)_+,$$

for $j \in \{1, \ldots, p\}$, with $(x)_+ := \max(x, 0)$. Note that $\mathrm{Prox}_{\mu\|\cdot\|_1}$ is componentwise the *soft-thresholding operator* of [42] presented in Section 1.4.

**$\ell_1$-norm constraint.** Sometimes, the $\ell_1$-norm is used as a hard constraint and, in that case, the optimization problem is

$$\min_{\boldsymbol{w}} f(\boldsymbol{w}) \quad \text{such that } \|\boldsymbol{w}\|_1 \leq C.$$

This problem can still be viewed as an instance of (1.1), with $\Omega$ defined by $\Omega(\boldsymbol{u}) = 0$ if $\|\boldsymbol{u}\|_1 \leq C$ and $\Omega(\boldsymbol{u}) = +\infty$ otherwise. Proximal methods thus apply and the corresponding proximal operator is the projection on the $\ell_1$-ball, itself an instance of a *quadratic continuous knapsack problem* for which efficient pivot algorithms with linear complexity have been proposed [27, 84]. Note that when penalizing by the dual norm of the $\ell_1$-norm, i.e., the $\ell_\infty$-norm, the proximal operator is also equivalent to the projection onto an $\ell_1$-ball.

**$\ell_2^2$-regularization (ridge regression).** This regularization function does not induce sparsity and is therefore slightly off topic here. It is nonetheless widely used and it is worth mentioning its proximal operator, which is a scaling operator:

$$\mathrm{Prox}_{\frac{\mu}{2}\|\cdot\|_2^2}[\boldsymbol{u}] = \frac{1}{1+\mu}\boldsymbol{u}.$$

**$\ell_1 + \ell_2^2$-regularization (Elastic-net [159]).** This regularization function combines the $\ell_1$-norm and the classical squared $\ell_2$-penalty. For a vector $\boldsymbol{w}$ in $\mathbb{R}^p$, it can be written $\Omega(\boldsymbol{w}) = \|\boldsymbol{w}\|_1 + \frac{\gamma}{2}\|\boldsymbol{w}\|_2^2$, where $\gamma > 0$ is an additional parameter. It is not a norm, but the proximal operator

can be obtained in closed form:

$$\mathrm{Prox}_{\mu(\|.\|_1+\frac{\gamma}{2}\|.\|_2^2)} = \mathrm{Prox}_{\frac{\mu\gamma}{2}\|.\|_2^2} \circ \mathrm{Prox}_{\mu\|.\|_1} = \frac{1}{\mu\gamma+1}\mathrm{Prox}_{\mu\|.\|_1}.$$

Similarly to the $\ell_1$-norm, when $\Omega$ is used as a constraint instead of a penalty, the proximal operator can be obtained in linear time using pivot algorithms (see [86], Appendix B.2).

**1D-total variation.** Originally introduced in the image processing community [117], the total-variation penalty encourages piecewise constant signals. It can be found in the statistics literature under the name of "fused lasso" [134]. For one-dimensional signals, it can be seen as the $\ell_1$-norm of finite differences for a vector $\boldsymbol{w}$ in $\mathbb{R}^p$: $\Omega_{\text{TV-1D}}(\boldsymbol{w}) := \sum_{i=1}^{p-1} |\boldsymbol{w}_{i+1} - \boldsymbol{w}_i|$. Even though no closed form is available for $\mathrm{Prox}_{\mu\Omega_{\text{TV-1D}}}$, it can be easily obtained using a modification of the homotopy algorithm presented later in this monograph in Section 6.2 (see [57, 58]). Similarly, it is possible to combine this penalty with the $\ell_1$- and squared $\ell_2$-penalties and efficiently compute $\mathrm{Prox}_{\Omega_{\text{TV-1D}}+\gamma_1\|.\|_1+\frac{\gamma_2}{2}\|.\|_2^2}$ or use such a regularization function in a constrained formulation (see [86], Appendix B.2).

**Anisotropic 2D-total variation.** The regularization function above can be extended to more than one dimension. For a two dimensional-signal $\boldsymbol{W}$ in $\mathbb{R}^{p\times l}$ this penalty is defined as $\Omega_{\text{TV-2D}}(\boldsymbol{W}) := \sum_{i=1}^{p-1}\sum_{j=1}^{l-1} |\boldsymbol{W}_{i+1,j} - \boldsymbol{W}_{i,j}| + |\boldsymbol{W}_{i,j+1} - \boldsymbol{W}_{i,j}|$. Interestingly, it has been shown in [34] that the corresponding proximal operator can be obtained by solving a parametric maximum flow problem.

**$\ell_1/\ell_q$-norm ("group Lasso").** If $\mathcal{G}$ is a partition of $\{1,\ldots,p\}$, the dual norm of the $\ell_1/\ell_q$-norm is the $\ell_\infty/\ell_{q'}$ norm, with $\frac{1}{q}+\frac{1}{q'}=1$. It is easy to show that the orthogonal projection on a unit $\ell_\infty/\ell_{q'}$ ball is obtained by projecting separately each subvector $\boldsymbol{u}_g$ on a unit $\ell_{q'}$-ball in $\mathbb{R}^{|g|}$. For the $\ell_1/\ell_2$-norm $\Omega\colon \boldsymbol{w} \mapsto \sum_{g\in\mathcal{G}} \|\boldsymbol{w}_g\|_2$ we have

$$[\mathrm{Prox}_{\mu\Omega}(\boldsymbol{u})]_g = \Big(1 - \frac{\lambda}{\|\boldsymbol{u}_g\|_2}\Big)_+ \boldsymbol{u}_g, \quad g \in \mathcal{G}. \tag{3.7}$$

This is shown easily by considering that the subgradient of the $\ell_2$-norm is $\partial\|\boldsymbol{w}\|_2 = \big\{\frac{\boldsymbol{w}}{\|\boldsymbol{w}\|_2}\big\}$ if $\boldsymbol{w} \neq \boldsymbol{0}$ or $\partial\|\boldsymbol{w}\|_2 = \{\boldsymbol{z} \mid \|\boldsymbol{z}\|_2 \leq 1\}$ if $\boldsymbol{w} = 0$ and by applying the result of Equation (1.7).

For the $\ell_1/\ell_\infty$-norm, whose dual norm is the $\ell_\infty/\ell_1$-norm, an efficient algorithm to compute the proximal operator is based on Equation (3.6). Indeed this equation indicates that the proximal operator can be computed on each group $g$ as the residual of a projection on an $\ell_1$-norm ball in $\mathbb{R}^{|g|}$; the latter is done efficiently with the previously mentioned linear-time algorithms.

**$\ell_1/\ell_2$-norm constraint.** When the $\ell_1/\ell_2$-norm is used as a constraint of the form $\Omega(\boldsymbol{w}) \leq C$, computing the proximal operator amounts to perform an orthogonal projection onto the $\ell_1/\ell_2$-ball of radius $C$. It is easy to show that such a problem can be recast as an orthogonal projection onto the simplex [142]. We know for instance that there exists a parameter $\mu \geq 0$ such that the solution $\boldsymbol{w}^\star$ of the projection is $\text{Prox}_{\mu\Omega}[\boldsymbol{u}]$ whose form is given in Equation (3.7). As a consequence, there exists scalars $z^g \geq 0$ such that $\boldsymbol{w}_g^\star = \frac{z^g}{\|\boldsymbol{u}_g\|_2}\boldsymbol{u}_g$ (where to simplify but without loss of generality we assume all the $\boldsymbol{u}_g$ to be non-zero). By optimizing over the scalars $z^g$, one can equivalently rewrite the projection as

$$\min_{(z^g)_{g\in\mathcal{G}}\in\mathbb{R}_+^{|\mathcal{G}|}} \frac{1}{2}\sum_{g\in\mathcal{G}}(\|\boldsymbol{u}_g\|_2 - z^g)^2 \quad \text{s.t.} \sum_{g\in\mathcal{G}} z^g \leq C,$$

which is a Euclidean projection of the vector $[\|\boldsymbol{u}_g\|_2]_{g\in\mathcal{G}}$ in $\mathbb{R}^{|\mathcal{G}|}$ onto a simplex.[5] The optimization problem above is then solved in linear time using the previously mentioned pivot algorithms [27, 84].

We have shown how to compute the proximal operator of group-norms when the groups form a partition. In general, the case where groups overlap is more complicated because the regularization is no longer separable. Nonetheless, in some cases it is still possible to compute efficiently the proximal operator.

**Hierarchical $\ell_1/\ell_q$-norms.** Hierarchical norms were proposed in [157]. Following [68], we focus on the case of a norm $\Omega\colon \boldsymbol{w} \mapsto \sum_{g\in\mathcal{G}}\|\boldsymbol{w}_g\|_q$, with $q \in \{2,\infty\}$, where the set of groups $\mathcal{G}$ is *tree-structured*, meaning that two groups are either disjoint or one is included in the other. Let $\preceq$ be a total order such that $g_1 \preceq g_2$ if

[5] This result also follows from Lemma 3.3 applied to the computation of the proximal operator of the $\ell_\infty/\ell_2$-norm which is dually related to the projection on the $\ell_1/\ell_2$-norm.

and only if either $g_1 \subset g_2$ or $g_1 \cap g_2 = \emptyset$.[6] Then, if $g_1 \preceq \cdots \preceq g_m$ with $m = |\mathcal{G}|$, and if we define $\Pi_g$ as (a) the proximal operator $\boldsymbol{w}_g \mapsto \text{Prox}_{\mu\|\cdot\|_q}(\boldsymbol{w}_g)$ on the subspace corresponding to group $g$ and (b) the identity on the orthogonal, it can be shown [68] that:

$$\text{Prox}_{\mu\Omega} = \Pi_{g_m} \circ \cdots \circ \Pi_{g_1}. \tag{3.8}$$

In other words, the proximal operator associated with the norm can be obtained as the composition of the proximal operators associated with individual groups provided that the ordering of the groups is well chosen. Note that this result does not hold for $q \notin \{1, 2, \infty\}$ (see [68] for more details). In terms of complexity, $\text{Prox}_{\mu\Omega}$ can be computed in $O(p)$ operations when $q = 2$ and $O(pd)$ when $q = \infty$, where $d$ is the depth of the tree.

**Combined $\ell_1 + \ell_1/\ell_q$-norm ("sparse group Lasso"), with $q \in \{2, \infty\}$.** The possibility of combining an $\ell_1/\ell_q$-norm that takes advantage of sparsity at the group level with an $\ell_1$-norm that induces sparsity within the groups is quite natural [49, 129]. Such regularizations are in fact a special case of the hierarchical $\ell_1/\ell_q$-norms presented above and the corresponding proximal operator is therefore readily computed by applying first the proximal operator associated with the $\ell_1$-norm (soft-thresholding) and the one associated with the $\ell_1/\ell_q$-norm (group soft-thresholding).

**Overlapping $\ell_1/\ell_\infty$-norms.** When the groups overlap but do not have a tree structure, computing the proximal operator has proven to be more difficult, but it can still be done efficiently when $q = \infty$. Indeed, as shown in [87], there exists a dual relation between such an operator and a quadratic min-cost flow problem on a particular graph, which can be tackled using network flow optimization techniques. Moreover, it may be extended to more general situations where structured sparsity is expressed through submodular functions [10].

**Trace norm and spectral functions.** The proximal operator for the trace norm, i.e., the unique minimizer of $\frac{1}{2}\|\boldsymbol{M} - \boldsymbol{N}\|_F^2 + \mu\|\boldsymbol{M}\|_*$

[6] For a tree-structured $\mathcal{G}$ such an order exists.

for a fixed matrix $\boldsymbol{M}$, may be obtained by computing a singular value decomposition of $\boldsymbol{N}$ and then replacing the singular values by their soft-thresholded versions [29]. This result can be easily extended to the case of spectral functions. Assume that the penalty $\Omega$ is of the form $\Omega(\boldsymbol{M}) = \psi(\boldsymbol{s})$ where $\boldsymbol{s}$ is a vector carrying the singular values of $\boldsymbol{M}$ and $\psi$ a convex function which is invariant by permutation of the variables (see, e.g., [20]). Then, it can be shown that $\mathrm{Prox}_{\mu\Omega}[\boldsymbol{N}] = \boldsymbol{U}\,\mathrm{Diag}(\mathrm{Prox}_{\mu\psi}[\boldsymbol{s}])\boldsymbol{V}^\top$, where $\boldsymbol{U}\,\mathrm{Diag}(\boldsymbol{s})\boldsymbol{V}^\top$ is a singular value decomposition of $\boldsymbol{N}$.

## 3.4 Proximal Methods for Structured MKL

In this section we show how proximal methods can be applied to solve multiple kernel learning problems. More precisely, we follow [95] who showed, in the context of plain MKL that proximal algorithms are applicable in an RKHS. We extend and present here this idea to the general case of structured MKL, showing that the proximal operator for the structured RKHS norm may be obtained from the proximal operator of the corresponding subquadratic norms.

Given a collection of reproducing kernel Hilbert spaces $\mathcal{H}_1, \ldots, \mathcal{H}_p$, we consider the Cartesian product $\mathcal{B} := \mathcal{H}_1 \times \cdots \times \mathcal{H}_p$, equipped with the norm $\|h\|_{\mathcal{B}} := (\|h_1\|_{\mathcal{H}_1}^2 + \cdots + \|h_p\|_{\mathcal{H}_p}^2)^{1/2}$, where $h = (h_1, \ldots, h_p)$ with $h_i \in \mathcal{H}_i$.

The set $\mathcal{B}$ is a Hilbert space, in which gradients and subgradients are well defined and in which we can extend some algorithms that we considered in the Euclidean case easily.

In the following, we will consider a *monotonic* norm as defined in Definition 1.1. This is motivated by the fact that a monotonic norm may be composed with norms of elements of Hilbert spaces to defines a norm on $\mathcal{B}$:

---

**Lemma 3.2.** Let $\Omega$ be a *monotonic* norm on $\mathbb{R}^p$ with dual norm $\Omega^*$, then $\Lambda\colon h \mapsto \Omega\big((\|h_1\|_{\mathcal{H}_1}, \ldots, \|h_p\|_{\mathcal{H}_p})\big)$ is a norm on $\mathcal{B}$ whose dual norm is $\Lambda^*\colon g \mapsto \Omega^*\big((\|g_1\|_{\mathcal{H}_1}, \ldots, \|g_p\|_{\mathcal{H}_p})\big)$. Moreover the subgradient of $\Lambda$ is $\partial\Lambda(h) = A(h)$ with $A(h) := \{(u_1 s_1, \ldots, u_p s_p) \mid \boldsymbol{u} \in B(h),\ s_i \in \partial\|\cdot\|_{\mathcal{H}_i}(h_i)\}$ with $B(h) := \partial\Omega\big((\|h_1\|_{\mathcal{H}_1}, \ldots, \|h_p\|_{\mathcal{H}_p})\big)$.

---

*Proof.* It is clear that $\Lambda$ is symmetric, nonnegative definite and homogeneous. The triangle inequality results from the fact that $\Omega$ is monotonic. Indeed the latter property implies that $\Lambda(h+g) = \Omega\big((\|h_i + g_i\|_{\mathcal{H}_i})_{1\leq i\leq p}\big) \leq \Omega\big((\|h_i\|_{\mathcal{H}_i} + \|g_i\|_{\mathcal{H}_i})_{1\leq i\leq p}\big)$ and the result follows from applying the triangle inequality for $\Omega$.

Moreover, we have the generalized Cauchy–Schwarz inequality:

$$\langle h, g\rangle_{\mathcal{B}} = \sum_i \langle h_i, g_i\rangle_{\mathcal{H}_i} \leq \sum_i \|h_i\|_{\mathcal{H}_i}\|g_i\|_{\mathcal{H}_i} \leq \Lambda(h)\ \Lambda^*(g),$$

and it is easy to check that equality is attained if and only if $g \in A(h)$. This simultaneously shows that $\Lambda(h) = \max_{g\in\mathcal{B},\ \Lambda^*(g)\leq 1}\langle h, g\rangle_{\mathcal{B}}$ (as a consequence of Proposition 1.4) and that $\partial\Lambda(h) = A(h)$ (by Proposition 1.2). □

We consider now a learning problem of the form:

$$\min_{h=(h_1,\ldots,h_p)\in\mathcal{B}} f(h_1,\ldots,h_p) + \lambda\,\Lambda(h), \tag{3.9}$$

with, typically, following Section 1.5, $f(h) = \frac{1}{n}\sum_{i=1}^n \ell(y^{(i)}, h(\boldsymbol{x}_i))$. The structured MKL case corresponds more specifically to the case where $f(h) = \frac{1}{n}\sum_{i=1}^n \ell\big(y^{(i)}, h_1(\boldsymbol{x}_i) + \cdots + h_p(\boldsymbol{x}_i)\big)$. Note that the problem we consider here is regularized with $\Lambda$ and not $\Lambda^2$ as opposed to the formulations (1.24) and (1.28) considered in Section 1.5.

To apply the proximal methods introduced in this section using $\|\cdot\|_{\mathcal{B}}$ as the proximal term requires one to be able to solve the proximal problem:

$$\min_{h\in\mathcal{B}} \frac{1}{2}\|h - g\|_{\mathcal{B}}^2 + \mu\,\Lambda(h). \tag{3.10}$$

The following lemma shows that if we know how to compute the proximal operator of $\Omega$ for an $\ell_2$ proximity term in $\mathbb{R}^p$, we can readily compute the proximal operator of $\Lambda$ for the proximity defined by the Hilbert norm on $\mathcal{B}$. Indeed, to obtain the image of $h$ by the proximal operator associated with $\Lambda$, it suffices to apply the proximal operator of $\Omega$ to the vector of norms $(\|h_1\|_{\mathcal{H}_1},\ldots,\|h_p\|_{\mathcal{H}_p})$ which yields a vector $(\boldsymbol{y}_1,\ldots,\boldsymbol{y}_p)$, and to scale in each space $\mathcal{H}_i$, the function $h_i$

by $\boldsymbol{y}_i/\|h_i\|_{\mathcal{H}_i}$. Precisely:

---

**Lemma 3.3.** $\mathrm{Prox}_{\mu\Lambda}(g) = (\boldsymbol{y}_1 s_1, \ldots, \boldsymbol{y}_p s_p)$ where $s_i = 0$ if $g_i = 0$,

$$s_i = \frac{g_i}{\|g_i\|_{\mathcal{H}_i}} \text{ if } g_i \neq 0 \quad \text{and} \quad \boldsymbol{y} = \mathrm{Prox}_{\mu\Omega}\big((\|g_i\|_{\mathcal{H}_i})_{1\leq i\leq p}\big).$$

---

*Proof.* To lighten the notations, we write $\|h_i\|$ for $\|h_i\|_{\mathcal{H}_i}$ if $h_i \in \mathcal{H}_i$. The optimality condition for problem (3.10) is $h - g \in -\mu\partial\Lambda$ so that we have $h_i = g_i - \mu\, s_i \boldsymbol{u}_i$, with $\boldsymbol{u} \in B(h)$, $s_i \in \partial\|\cdot\|_{\mathcal{H}_i}(h_i)$. The last equation implies that $h_i, g_i$ and $s_i$ are colinear. If $g_i = 0$ then the fact that $\Omega$ is monotonic implies that $h_i = s_i = 0$. If on the other hand, $g_i \neq 0$ we have $h_i = g_i\,(1 - \frac{\mu\,\boldsymbol{u}_i}{\|g_i\|})_+$ and thus $\|h_i\| = (\|g_i\| - \mu\,\boldsymbol{u}_i)_+$ and $h_i = s_i\|h_i\|$, but by the optimality conditions of the proximal problem defining $\boldsymbol{y}_i$ we have $\boldsymbol{y}_i = (\|g_i\| - \mu\,\boldsymbol{u}_i)_+$, which shows the result. □

This result shows how to compute the proximal operator at an abstract level. For the algorithm to be practical, we need to show that the corresponding computation can be performed by manipulating a finite number of parameters.

Fortunately, we can appeal to a representer theorem to that end, which leads to the following lemma:

---

**Lemma 3.4.** Assume that for all $i$, $g_i = \sum_{j=1}^n \boldsymbol{\alpha}_{ij} K_i(\boldsymbol{x}_j, \cdot)$. Then the solution of problem (3.10) is of the form $h_i = \sum_{j=1}^n \boldsymbol{\beta}_{ij} K_i(\boldsymbol{x}_j, \cdot)$. Let $\boldsymbol{y} = \mathrm{Prox}_{\mu\Omega}\big((\sqrt{\boldsymbol{\alpha}_k^\top \boldsymbol{K}_k \boldsymbol{\alpha}_k})_{1\leq k\leq p}\big)$. Then if $\boldsymbol{\alpha}_i \neq 0$, $\boldsymbol{\beta}_i = \frac{\boldsymbol{y}_i}{\sqrt{\boldsymbol{\alpha}_i^\top \boldsymbol{K}_i \boldsymbol{\alpha}_i}}\boldsymbol{\alpha}_i$ and otherwise $\boldsymbol{\beta}_i = 0$.

---

*Proof.* We first show that a representer theorem holds. For each $i$ let $h_i^{/\!/}$ be the component of $h_i$ in the span of $K_i(\boldsymbol{x}_j, \cdot)_{1\leq j\leq n}$ and $h_i^\perp = h_i - h_i^{/\!/}$. We can rewrite the objective of problem (3.10) as[7]

$$\frac{1}{2}\sum_{i=1}^p \big[\|h_i^{/\!/}\|^2 + \|h_i^\perp\|^2 - 2\langle h_i^{/\!/}, g_i\rangle_{\mathcal{H}_i} + \|g_i\|^2\big] + \mu\Omega\big((\|h_i^{/\!/} + h_i^\perp\|)_{1\leq i\leq p}\big),$$

[7] We denote again $\|h_i\|$ for $\|h_i\|_{\mathcal{H}_i}$, when the RHKS norm used is implied by the argument.

from which, given that $\Omega$ is assumed monotonic, it is clear that setting $h_i^\perp = 0$ for all $i$ can only decrease the objective. To conclude, the form of the solution in $\boldsymbol{\beta}$ results from the fact that $\|g_i\|^2_{\mathcal{H}_i} = \sum_{1 \leq j,j' \leq n} \boldsymbol{\alpha}_{ij} \boldsymbol{\alpha}_{ij'} \langle K_i(\boldsymbol{x}_j, \cdot), K_i(\boldsymbol{x}_{j'}, \cdot) \rangle_{\mathcal{H}_i}$ and $\langle K_i(\boldsymbol{x}_j, \cdot), K_i(\boldsymbol{x}_{j'}, \cdot) \rangle_{\mathcal{H}_i} = K_i(\boldsymbol{x}_j, \boldsymbol{x}_{j'})$ by the reproducing property, and by identification (note that if the kernel matrix $\boldsymbol{K}_i$ is not invertible the solution might not be unique in $\boldsymbol{\beta}_i$). □

Finally, in the last lemma we assumed that the function $g_i$ in the proximal problem could be represented as a linear combination of the $K_i(\boldsymbol{x}_j, \cdot)$. Since $g_i$ is typically of the form $h_i^t - \frac{1}{L} \frac{\partial}{\partial h_i} f(h_1^t, \ldots, h_p^t)$, when, as in Equation 3.2, we apply the gradient operator after a gradient step then the result follows by linearity if the gradient is in the span of the $\boldsymbol{K}_i(\boldsymbol{x}_j, \cdot)$. The following lemma shows that this is indeed the case:

---

**Lemma 3.5.** For $f(h) = \frac{1}{n} \sum_{j=1}^n \ell(y^{(j)}, h_1(\boldsymbol{x}_j), \ldots, h_p(\boldsymbol{x}_j))$ then

$$\frac{\partial}{\partial h_i} f(h) = \sum_{j=1}^n \boldsymbol{\alpha}_{ij} K_i(\boldsymbol{x}_j, \cdot) \quad \text{for } \boldsymbol{\alpha}_{ij} = \frac{1}{n} \partial_i \ell(y^{(j)}, h_1(\boldsymbol{x}_j), \ldots, h_p(\boldsymbol{x}_j)),$$

where $\partial_i \ell$ denote the partial derivative of $\ell$ w.r.t. to its $(i+1)$th scalar component.

---

*Proof.* This result follows from the rules of composition of differentiation applied to the functions

$$(h_1, \ldots, h_p) \mapsto \ell\big(y^{(j)}, \langle h_1, K_1(x_j, \cdot) \rangle_{\mathcal{H}_1}, \ldots, \langle h_p, K_p(x_j, \cdot) \rangle_{\mathcal{H}_p}\big),$$

and the fact that, since $h_i \mapsto \langle h_i, K_i(\boldsymbol{x}_j, \cdot) \rangle_{\mathcal{H}_i}$ is linear, its gradient in the RKHS $\mathcal{H}_i$ is just $K_i(\boldsymbol{x}_j, \cdot)$. □

# 4

# (Block) Coordinate Descent Algorithms

Coordinate descent algorithms solving $\ell_1$-regularized learning problems go back to [51]. They optimize (exactly or approximately) the objective function with respect to one variable at a time while all others are kept fixed. Note that, in general, coordinate descent algorithms are not necessarily convergent for non-smooth optimization (cf. [18], p. 636); they are however applicable in this setting because of a *separability* property of the nonsmooth regularizer we consider (see end of Section 4.1).

## 4.1 Coordinate Descent for $\ell_1$-Regularization

We consider first the following special case of the one-dimensional $\ell_1$-regularized problem:

$$\min_{w \in \mathbb{R}} \frac{1}{2}(w - w_0)^2 + \lambda |w|. \tag{4.1}$$

As shown in (1.5), the minimizer $w^\star$ can be obtained by *soft-thresholding*:

$$w^\star = \mathrm{Prox}_{\lambda|\cdot|}(w_0) := \left(1 - \frac{\lambda}{|w_0|}\right)_+ w_0. \tag{4.2}$$

**Lasso case.** In the case of the square loss, the minimization with respect to a single coordinate can be written as

$$\min_{\boldsymbol{w}_j \in \mathbb{R}} \nabla_j f(\boldsymbol{w}^t)\,(\boldsymbol{w}_j - \boldsymbol{w}_j^t) + \frac{1}{2}\nabla^2_{jj} f(\boldsymbol{w}^t)(\boldsymbol{w}_j - \boldsymbol{w}_j^t)^2 + \lambda|\boldsymbol{w}_j|,$$

with $\nabla_j f(\boldsymbol{w}) = \frac{1}{n}\boldsymbol{X}_j^\top(\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y})$ and $\nabla^2_{jj} f(\boldsymbol{w}) = \frac{1}{n}\boldsymbol{X}_j^\top \boldsymbol{X}_j$ independent of $\boldsymbol{w}$. Since the above equation is of the form (4.1), it can be solved in closed form:

$$\boldsymbol{w}_j^\star = \mathrm{Prox}_{\frac{\lambda}{\nabla^2_{jj} f}|\cdot|}\big(\boldsymbol{w}_j^t - \nabla_j f(\boldsymbol{w}_j^t)/\nabla^2_{jj} f\big). \tag{4.3}$$

In other words, $\boldsymbol{w}_j^\star$ is obtained by solving the unregularized problem with respect to coordinate $j$ and soft-thresholding the solution.

This is the update proposed in the shooting algorithm of Fu [51], which cycles through all variables in a fixed order.[1] Other cycling schemes are possible (see, e.g., [103]).

An efficient implementation is obtained if the quantity $\boldsymbol{X}\boldsymbol{w}^t - \boldsymbol{y}$ or even better $\nabla f(\boldsymbol{w}^t) = \frac{1}{n}\boldsymbol{X}^\top\boldsymbol{X}\boldsymbol{w}^t - \frac{1}{n}\boldsymbol{X}^\top\boldsymbol{y}$ is kept updated.[2]

**Smooth loss.** For more general smooth losses, like the logistic loss, the optimization with respect to a single variable cannot be solved in closed form. It is possible to solve it numerically using a sequence of modified Newton steps as proposed in [127]. We present here a fast algorithm of Tseng and Yun [140] based on solving just a quadratic approximation of $f$ with an inexact line search at each iteration.

Let $L^t > 0$ be a parameter and let $\boldsymbol{w}_j^\star$ be the solution of

$$\min_{\boldsymbol{w}_j \in \mathbb{R}} \nabla_j f(\boldsymbol{w}^t)\,(\boldsymbol{w}_j - \boldsymbol{w}_j^t) + \frac{1}{2}L^t(\boldsymbol{w}_j - \boldsymbol{w}_j^t)^2 + \lambda|\boldsymbol{w}_j|,$$

Given $d = \boldsymbol{w}_j^\star - \boldsymbol{w}_j^t$ where $\boldsymbol{w}_j^\star$ is the solution of (4.3), the algorithm of Tseng and Yun performs a line search to choose the largest step of

[1] Coordinate descent with a cyclic order is sometimes called a Gauss–Seidel procedure.

[2] In the former case, at each iteration, $\boldsymbol{X}\boldsymbol{w} - \boldsymbol{y}$ can be updated in $O(n)$ operations if $\boldsymbol{w}_j$ changes and $\nabla_{j^{t+1}} f(\boldsymbol{w})$ can always be updated in $O(n)$ operations. The complexity of one cycle is therefore $O(pn)$. However a better complexity is obtained in the latter case, provided the matrix $\boldsymbol{X}^\top\boldsymbol{X}$ is precomputed (with complexity $O(p^2 n)$). Indeed $\nabla f(\boldsymbol{w}^t)$ is updated in $O(p)$ iterations only if $w_j$ does not stay at 0. Otherwise, if $w_j$ stays at 0 the step costs $O(1)$; the complexity of one cycle is therefore $O(ps)$ where $s$ is the number of nonzero variables at the end of the cycle.

the form $\alpha d$ with $\alpha = \alpha_0\beta^k$ and $\alpha_0 > 0, \beta \in (0,1), k \in \mathbb{N}$, such that the following modified Armijo condition is satisfied:

$$F(\boldsymbol{w}^t + \alpha d \boldsymbol{e}_j) - F(\boldsymbol{w}^t) \leq \sigma\alpha\big(\nabla_j f(\boldsymbol{w})d + \gamma L^t d^2 + |\boldsymbol{w}_j^t + d| - |\boldsymbol{w}_j^t|\big),$$

where $F(\boldsymbol{w}) := f(\boldsymbol{w}) + \lambda\Omega(\boldsymbol{w})$, and $0 \leq \gamma \leq 1$ and $\sigma < 1$ are parameters of the algorithm.

Tseng and Yun [140] show that under mild conditions on $f$ the algorithm is convergent and, under further assumptions, asymptotically linear. In particular, if $f$ is of the form $\frac{1}{n}\sum_{i=1}^n \ell(y^{(i)}, \boldsymbol{w}^\top \boldsymbol{x}^{(i)})$ with $\ell(y^{(i)}, \cdot)$ a twice continuously differentiable convex function with strictly positive curvature, the algorithm is asymptotically linear for $L^t = \nabla^2_{jj} f(\boldsymbol{w}_j^t)$. We refer the reader to Section 4.2 and to [140, 150] for results under much milder conditions. It should be noted that the algorithm generalizes to other separable regularizations than the $\ell_1$-norm.

Variants of coordinate descent algorithms have also been considered in [53, 76, 152]. Generalizations based on the Gauss–Southwell rule have been considered in [140].

**Convergence of coordinate descent algorithms.** In general, coordinate descent algorithms are not convergent for non-smooth objectives. Therefore, using such schemes always requires a convergence analysis. In the context of the $\ell_1$-norm regularized smooth objective, the non-differentiability is *separable* (i.e., is a sum of non-differentiable terms that depend on single variables), and this is sufficient for convergence [18, 140]. In terms of convergence rates, coordinate descent behaves in a similar way to first-order methods such as proximal methods, i.e., if the objective function is strongly convex [103, 140], then the convergence is linear, while it is slower if the problem is not strongly convex, e.g., in the learning context, if there are strong correlations between input variables [124].

## 4.2 Block-Coordinate Descent for $\ell_1/\ell_q$-Regularization

When $\Omega(\boldsymbol{w})$ is the $\ell_1/\ell_q$-norm with groups $g \in \mathcal{G}$ forming a partition of $\{1, \ldots, p\}$, the previous methods are generalized by block-coordinate descent (BCD) algorithms, which have been the focus of recent work by

Tseng and Yun [140] and Wright [150]. These algorithms do not attempt to solve exactly a reduced problem on a block of coordinates but rather optimize a surrogate of $F$ in which the function $f$ is substituted by a quadratic approximation.

Specifically, the BCD scheme of [140] solves at each iteration a problem of the form:

$$\min_{\boldsymbol{w}_g \in \mathbb{R}^{|g|}} \nabla_g f(\boldsymbol{w}^t)^\top (\boldsymbol{w}_g - \boldsymbol{w}_g^t) + \frac{1}{2}(\boldsymbol{w}_g - \boldsymbol{w}_g^t)^\top \boldsymbol{H}^t (\boldsymbol{w}_g - \boldsymbol{w}_g^t) + \lambda \|\boldsymbol{w}_g\|_q, \tag{4.4}$$

where the positive semi-definite matrix $\boldsymbol{H}^t \in \mathbb{R}^{|g| \times |g|}$ is a parameter. Note that this may correspond to a richer quadratic approximation around $\boldsymbol{w}_g^t$ than the proximal terms used in Section 3. However, in practice, the above problem is solved in closed form if $\boldsymbol{H}^t = L^t \boldsymbol{I}_{|g|}$ for some scalar $L^t$ and $q \in \{2, \infty\}$.[3] In particular for $q = 2$, the solution $\boldsymbol{w}_g^\star$ is obtained by *group-soft-thresholding*:

$$\boldsymbol{w}_g^\star = \mathrm{Prox}_{\frac{\lambda}{L^t}\|\cdot\|_2}\left(\boldsymbol{w}_g^t - \frac{1}{L^t}\nabla_g f(\boldsymbol{w}_g^t)\right),$$

with

$$\mathrm{Prox}_{\mu\|\cdot\|_2}(\boldsymbol{w}) = \left(1 - \frac{\mu}{\|\boldsymbol{w}\|_2}\right)_+ \boldsymbol{w}.$$

In the case of general smooth losses, the descent direction is given by $\boldsymbol{d} = \boldsymbol{w}_g^\star - \boldsymbol{w}_g^t$ with $\boldsymbol{w}_g^\star$ as above. The next point is of the form $\boldsymbol{w}^t + \alpha \boldsymbol{d}$, where $\alpha$ is a stepsize of the form $\alpha = \alpha_0 \beta^k$, with $\alpha_0 > 0$, $0 < \beta < 1$, $k \in \mathbb{N}$. The parameter $k$ is chosen large enough to satisfy the following modified Armijo condition

$$\begin{aligned} &F(\boldsymbol{w}^t + \alpha \boldsymbol{d}) - F(\boldsymbol{w}^t) \\ &\quad \leq \sigma\alpha\big(\nabla_g f(\boldsymbol{w})^\top \boldsymbol{d} + \gamma\, \boldsymbol{d}^\top \boldsymbol{H}^t \boldsymbol{d} + \|\boldsymbol{w}_g^t + \boldsymbol{d}\|_q - \|\boldsymbol{w}_g^t\|_q\big), \end{aligned}$$

for parameters $0 \leq \gamma \leq 1$ and $\sigma < 1$.

If $f$ is convex continuously differentiable, lower bounded on $\mathbb{R}^p$ and $F$ has a unique minimizer, provided that there exists $\tau, \bar{\tau}$ fixed constants such that for all $t$, $\tau \preceq \boldsymbol{H}^t \preceq \bar{\tau}$, the results of Tseng and Yun show

[3] More generally for $q \geq 1$ and $\boldsymbol{H}^t = L^t \boldsymbol{I}_{|g|}$, it can be solved efficiently coordinate-wise using bisection algorithms.

that the algorithm converges (see Theorem 4.1 in [140] for broader conditions). Wright [150] proposes a variant of the algorithm, in which the line-search on $\alpha$ is replaced by a line search on the parameter $L^t$, similar to the line-search strategies used in proximal methods.

## 4.3 Block-coordinate Descent for MKL

Finally, block-coordinate descent algorithms are also applicable to classical multiple kernel learning. We consider the same setting and notation as in Section 3.4 and we consider specifically the optimization problem:

$$\min_{h \in \mathcal{B}} f(h_1, \ldots, h_p) + \lambda \sum_{i=1}^{p} \|h_i\|_{\mathcal{H}_i}.$$

A block-coordinate algorithm can be applied by considering each RKHS $\mathcal{H}_i$ as one "block"; this type of algorithm was considered in [113]. Applying the Lemmas 3.4 and 3.5 of Section 3.4, we know that $h_i$ can be represented as $h_i = \sum_{j=1}^{n} \boldsymbol{\alpha}_{ij} K_i(\boldsymbol{x}_j, \cdot)$.

The algorithm then consists in performing successively group soft-thresholding in each RKHS $\mathcal{H}_i$. This can be done by working directly with the dual parameters $\boldsymbol{\alpha}_i$, with a corresponding proximal operator in the dual simply formulated as:

$$\mathrm{Prox}_{\mu \|\cdot\|_{\boldsymbol{K}_i}}(\boldsymbol{\alpha}_i) = \Big(1 - \frac{\mu}{\|\boldsymbol{\alpha}_i\|_{\boldsymbol{K}_i}}\Big)_+ \boldsymbol{\alpha}_i.$$

with $\|\boldsymbol{\alpha}\|_{\boldsymbol{K}}^2 = \boldsymbol{\alpha}^\top \boldsymbol{K} \boldsymbol{\alpha}$. The precise equations would be obtained by kernelizing Equation (4.4) (which requires kernelizing the computation of the gradient and the Hessian as in Lemma 3.5). We leave the details to the reader.

# 5

# Reweighted-$\ell_2$ Algorithms

Approximating a nonsmooth or constrained optimization problem by a series of smooth unconstrained problems is common in optimization (see, e.g., [25, 101, 104]). In the context of objective functions regularized by sparsity-inducing norms, it is natural to consider variational formulations of these norms in terms of squared $\ell_2$-norms, since many efficient methods are available to solve $\ell_2$-regularized problems (e.g., linear system solvers for least-squares regression).

## 5.1 Variational Formulations for Grouped $\ell_1$-norms

In this section, we show on our motivating example of sums of $\ell_2$-norms of subsets how such formulations arise (see, e.g., [41, 55, 70, 109, 110]). The variational formulation we have presented in Section 1.4.2 allows us to consider the following function $H(\boldsymbol{w}, \boldsymbol{\eta})$ defined as

$$H(\boldsymbol{w}, \boldsymbol{\eta}) = f(\boldsymbol{w}) + \frac{\lambda}{2} \sum_{j=1}^{p} \left\{ \sum_{g \in \mathcal{G}, j \in g} \eta_g^{-1} \right\} w_j^2 + \frac{\lambda}{2} \sum_{g \in \mathcal{G}} \eta_g.$$

It is jointly convex in $(\boldsymbol{w}, \boldsymbol{\eta})$; the minimization with respect to $\boldsymbol{\eta}$ can be done in closed form, and the optimum is equal to $F(\boldsymbol{w}) = f(\boldsymbol{w}) + \lambda\Omega(\boldsymbol{w})$; as for the minimization with respect to $\boldsymbol{w}$, it is an $\ell_2$-regularized problem.

Unfortunately, the alternating minimization algorithm that is immediately suggested is not convergent in general, because the function $H$ is not continuous (in particular around $\boldsymbol{\eta}$ which has zero coordinates). In order to make the algorithm convergent, two strategies are commonly used:

— **Smoothing**: we can add a term of the form $\frac{\varepsilon}{2}\sum_{g\in\mathcal{G}}\boldsymbol{\eta}_g^{-1}$, which yields a joint cost function with compact level sets on the set of positive numbers. Alternating minimization algorithms are then convergent (as a consequence of general results on block coordinate descent), and have two different iterations: (1) minimization with respect to $\boldsymbol{\eta}$ in closed form, through $\boldsymbol{\eta}_g = \sqrt{\|\boldsymbol{w}_g\|_2^2 + \varepsilon}$, and (2) minimization with respect to $\boldsymbol{w}$, which is an $\ell_2$-regularized problem, which can be, for example, solved in closed form for the square loss. Note however, that the second problem does not need to be exactly optimized at all iterations.

— **First order method in $\boldsymbol{\eta}$**: while the joint cost function $H(\boldsymbol{\eta}, \boldsymbol{w})$ is not continuous, the function $I(\boldsymbol{\eta}) = \min_{\boldsymbol{w}\in\mathbb{R}^p} H(\boldsymbol{w}, \boldsymbol{\eta})$ is continuous, and under general assumptions, continuously differentiable, and is thus amenable to first-order methods (e.g., proximal methods, gradient descent). When the groups in $\mathcal{G}$ do not overlap, one sufficient condition is that the function $f(\boldsymbol{w})$ is of the form $f(\boldsymbol{w}) = \psi(\boldsymbol{X}\boldsymbol{w})$ for $\boldsymbol{X} \in \mathbb{R}^{n\times p}$ any matrix (typically the design matrix) and $\psi$ a strongly convex function on $\mathbb{R}^n$. This strategy is particularly interesting when evaluating $I(\boldsymbol{\eta})$ is computationally cheap.

In theory, the alternating scheme consisting of optimizing alternately over $\boldsymbol{\eta}$ and $\boldsymbol{w}$ can be used to solve learning problems regularized with *any* norms: on top of the subquadratic norms defined in Section 1.4.2, we indeed show in the next section that any norm admits a quadratic variational formulation (potentially defined from a non-diagonal symmetric positive matrix). To illustrate the principle of $\ell_2$-reweighted algorithms, we first consider the special case of multiple kernel learning; in Section 5.2, we consider the case of the trace norm.

**Structured MKL.** Reweighted-$\ell_2$ algorithms are fairly natural for norms which admit a diagonal variational formulation (see Lemma 1.8 and [93]) and for the corresponding multiple kernel learning problem. We consider the structured multiple learning problem presented in Section 1.5.2.

The alternating scheme applied to Equation (1.27) then takes the following form: for $\boldsymbol{\eta}$ fixed, one has to solve a single kernel learning problem with the kernel $K = \sum_i \boldsymbol{\eta}_i K_i$; the corresponding solution in the product of RKHSs $\mathcal{H}_1 \times \cdots \times \mathcal{H}_p$ (see Section 3.4) is of the form $h(\boldsymbol{x}) = h_1(\boldsymbol{x}) + \cdots + h_p(\boldsymbol{x})$ with $h_i(\boldsymbol{x}) = \boldsymbol{\eta}_i \sum_{j=1}^n \boldsymbol{\alpha}_j K_i(\boldsymbol{x}_j, \cdot)$. Since $\|h_i\|_{\mathcal{H}_i}^2 = \boldsymbol{\eta}_i^2 \boldsymbol{\alpha}^\top \boldsymbol{K}_i \boldsymbol{\alpha}$, for fixed $\boldsymbol{\alpha}$, the update in $\boldsymbol{\eta}$ then takes the form:

$$\boldsymbol{\eta}^{t+1} \leftarrow \operatorname*{argmin}_{\boldsymbol{\eta} \in H} \sum_{i=1}^p \frac{(\boldsymbol{\eta}_i^t)^2 {\boldsymbol{\alpha}^t}^\top \boldsymbol{K}_i \boldsymbol{\alpha}^t + \varepsilon}{\boldsymbol{\eta}_i}.$$

Note that these updates produce a non-increasing sequence of values of the primal objective. Moreover, this MKL optimization scheme uses a potentially much more compact parameterization than proximal methods since in addition to the variational parameter $\boldsymbol{\eta} \in \mathbb{R}^p$ a single vector of parameters $\boldsymbol{\alpha} \in \mathbb{R}^n$ is needed as opposed to up to one such vector for each kernel in the case of proximal methods. MKL problems can also be tackled using first order methods in $\boldsymbol{\eta}$ described above: we refer the reader to [110] for an example in the case of classical MKL.

## 5.2 Quadratic Variational Formulation for General Norms

We now investigate a general variational formulation of norms that naturally leads to a sequence of reweighted $\ell_2$-regularized problems. The formulation is based on approximating the unit ball of a norm $\Omega$ with enclosing ellipsoids. See Figure 5.1. The following proposition shows that all norms may be expressed as a minimum of Euclidean norms:

---

**Proposition 5.1.** Let $\Omega : \mathbb{R}^p \to \mathbb{R}$ be a norm on $\mathbb{R}^p$, then there exists a function $g$ defined on the cone of positive semi-definite matrices $\mathbf{S}_p^+$, such that $g$ is convex, strictly positive except at zero, positively homogeneous and such that for all $\boldsymbol{w} \in \mathbb{R}^p$,

$$\Omega(\boldsymbol{w}) = \min_{\boldsymbol{\Lambda} \in \mathbf{S}_p^+,\ g(\boldsymbol{\Lambda}) \leqslant 1} \sqrt{\boldsymbol{w}^\top \boldsymbol{\Lambda}^{-1} \boldsymbol{w}} = \frac{1}{2} \min_{\boldsymbol{\Lambda} \in \mathbf{S}_p^+} \big\{ \boldsymbol{w}^\top \boldsymbol{\Lambda}^{-1} \boldsymbol{w} + g(\boldsymbol{\Lambda}) \big\}. \quad (5.1)$$

---

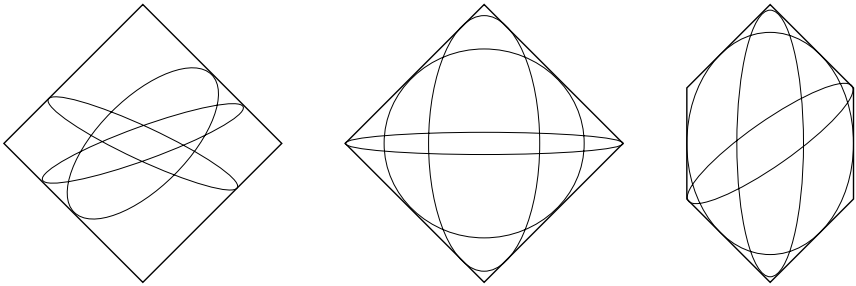


Fig. 5.1. Example of a sparsity-inducing ball in two dimensions, with enclosing ellipsoids. Left: ellipsoids with general axis for the $\ell_1$-norm; middle: ellipsoids with horizontal and vertical axis for the $\ell_1$-norm; right: ellipsoids for another polyhedral norm.

*Proof.* Let $\Omega^*$ be the dual norm of $\Omega$, defined as $\Omega^*(\boldsymbol{s}) = \max_{\Omega(\boldsymbol{w}) \leqslant 1} \boldsymbol{w}^\top \boldsymbol{s}$ [25]. Let $g$ be the function defined through $g(\boldsymbol{\Lambda}) = \max_{\Omega^*(\boldsymbol{s}) \leqslant 1} \boldsymbol{s}^\top \boldsymbol{\Lambda} \boldsymbol{s}$. This function is well-defined as the maximum of a continuous function over a compact set; moreover, as a maximum of linear functions, it is convex and positive homogeneous. Also, for nonzero $\boldsymbol{\Lambda}$, the quadratic form $\boldsymbol{s} \mapsto \boldsymbol{s}^\top \boldsymbol{\Lambda} \boldsymbol{s}$ is not identically zero around $\boldsymbol{s} = 0$, hence the strict positivity of $g$.

Let $\boldsymbol{w} \in \mathbb{R}^p$ and $\boldsymbol{\Lambda} \in \mathbf{S}_p^+$; there exists $\boldsymbol{s}$ such that $\Omega^*(\boldsymbol{s}) = 1$ and $\boldsymbol{w}^\top \boldsymbol{s} = \Omega(\boldsymbol{w})$. We then have

$$\Omega(\boldsymbol{w})^2 = (\boldsymbol{w}^\top \boldsymbol{s})^2 \leqslant (\boldsymbol{w}^\top \boldsymbol{\Lambda}^{-1} \boldsymbol{w})(\boldsymbol{s}^\top \boldsymbol{\Lambda} \boldsymbol{s}) \leqslant g(\boldsymbol{\Lambda})(\boldsymbol{w}^\top \boldsymbol{\Lambda}^{-1} \boldsymbol{w}).$$

This shows that $\Omega(\boldsymbol{w}) \leqslant \min_{\boldsymbol{\Lambda} \in \mathbf{S}_p^+,\ g(\boldsymbol{\Lambda}) \leqslant 1} \sqrt{\boldsymbol{w}^\top \boldsymbol{\Lambda}^{-1} \boldsymbol{w}}$. Proving the other direction can be done using the following limiting argument. Given $\boldsymbol{w}_0 \in \mathbb{R}^p$, consider $\boldsymbol{\Lambda}(\varepsilon) = (1 - \varepsilon) \boldsymbol{w}_0 \boldsymbol{w}_0^\top + \varepsilon (\boldsymbol{w}_0^\top \boldsymbol{w}_0) I$. We have $\boldsymbol{w}_0^\top \boldsymbol{\Lambda}(\varepsilon)^{-1} \boldsymbol{w}_0 = 1$ and $g(\boldsymbol{\Lambda}(\varepsilon)) \to g(\boldsymbol{w}_0 \boldsymbol{w}_0^\top) = \Omega(\boldsymbol{w}_0)^2$. Thus, for $\tilde{\boldsymbol{\Lambda}}(\varepsilon) = \boldsymbol{\Lambda}(\varepsilon) / g(\boldsymbol{\Lambda}(\varepsilon))$, we have that $\sqrt{\boldsymbol{w}_0^\top \tilde{\boldsymbol{\Lambda}}(\varepsilon)^{-1} \boldsymbol{w}_0}$ tends to $\Omega(\boldsymbol{w}_0)$, thus $\Omega(\boldsymbol{w}_0)$ must be no smaller than the minimum over all $\boldsymbol{\Lambda}$. The right-hand side of Equation (5.1) can be obtained by optimizing over the scale of $\boldsymbol{\Lambda}$. □

Note that while the proof provides a closed-form expression for a candidate function $g$, it is not unique, as can be seen in the following examples. The domain of $g$ (matrices so that $g$ is finite) may be reduced (in particular to diagonal matrices for the $\ell_1$-norm and more generally

the sub-quadratic norms defined in Section 1.4.2):

- For the $\ell_1$-norm: using the candidate from the proof, we have $g(\boldsymbol{\Lambda}) = \max_{\|\boldsymbol{s}\|_\infty \leq 1} \boldsymbol{s}^\top \boldsymbol{\Lambda} \boldsymbol{s}$, but we could use $g(\boldsymbol{\Lambda}) = \operatorname{Tr} \boldsymbol{\Lambda}$ if $\boldsymbol{\Lambda}$ is diagonal and $+\infty$ otherwise.
- For subquadratic norms (Section 1.4.2), we can take $g(\boldsymbol{\Lambda})$ to be $+\infty$ for non-diagonal $\boldsymbol{\Lambda}$, and either equal to the gauge function of the set $H$, i.e. the function $\boldsymbol{s} \mapsto \inf\{\nu \in \mathbb{R}_+ \mid \boldsymbol{s} \in \nu H\}$, or equal to the function $\bar{\Omega}$ defined in Lemma 1.9, both applied to the diagonal of $\boldsymbol{\Lambda}$.
- For the $\ell_2$-norm: $g(\boldsymbol{\Lambda}) = \lambda_{\max}(\boldsymbol{\Lambda})$ but we could of course use $g(\boldsymbol{\Lambda}) = 1$ if $\boldsymbol{\Lambda} = I$ and $+\infty$ otherwise.
- For the trace norm: $\boldsymbol{w}$ is assumed to be of the form $\boldsymbol{w} = \operatorname{vect}(\boldsymbol{W})$ and the trace norm of $\boldsymbol{W}$ is regularized. The trace norm admits the variational form (see [6]):

$$\|\boldsymbol{W}\|_* = \frac{1}{2} \inf_{\boldsymbol{D} \succeq 0} \operatorname{tr}(\boldsymbol{W}^\top \boldsymbol{D}^{-1} \boldsymbol{W} + \boldsymbol{D}), \quad \text{s.t.} \quad \boldsymbol{D} \succ 0. \quad (5.2)$$

But $\operatorname{tr}(\boldsymbol{W}^\top \boldsymbol{D}^{-1} \boldsymbol{W}) = \boldsymbol{w}^\top (\boldsymbol{I} \otimes \boldsymbol{D})^{-1} \boldsymbol{w}$, which shows that the regularization by the trace norm takes the form of Equation (5.1) in which we can choose $g(\boldsymbol{\Lambda})$ equal to $\operatorname{tr}(\boldsymbol{D})$ if $\boldsymbol{\Lambda} = \boldsymbol{I} \otimes \boldsymbol{D}$ for some $\boldsymbol{D} \succ 0$ and $+\infty$ otherwise.

The solution of the above optimization problem is given by $\boldsymbol{D}^\star = (\boldsymbol{W}\boldsymbol{W}^\top)^{1/2}$ which can be computed via a singular value decomposition of $\boldsymbol{W}$. The reweighted-$\ell_2$ algorithm to solve

$$\min_{\boldsymbol{W} \in \mathbb{R}^{p \times k}} f(\boldsymbol{W}) + \lambda \|\boldsymbol{W}\|_*$$

therefore consists of iterating between the two updates (see, e.g., [6] for more details):

$$\boldsymbol{W} \leftarrow \operatorname*{argmin}_{\boldsymbol{W}} f(\boldsymbol{W}) + \frac{\lambda}{2} \operatorname{Tr}(\boldsymbol{W}^\top \boldsymbol{D}^{-1} \boldsymbol{W}) \quad \text{and}$$

$$\boldsymbol{D} \leftarrow (\boldsymbol{W}\boldsymbol{W}^\top + \varepsilon \boldsymbol{I}_k)^{1/2},$$

where $\varepsilon$ is a smoothing parameter that arises from adding a term $\frac{\varepsilon\lambda}{2} \operatorname{Tr}(\boldsymbol{D}^{-1})$ to Equation (5.2) and prevents the matrix from becoming singular.

# 6

# Working-Set and Homotopy Methods

In this section, we consider methods that explicitly take into account the fact that the solutions are sparse, namely working set methods and homotopy methods.

## 6.1 Working-Set Techniques

Working-set algorithms address optimization problems by solving an increasing sequence of small subproblems of (1.1), which we recall can be written as

$$\min_{\boldsymbol{w} \in \mathbb{R}^p} f(\boldsymbol{w}) + \lambda \Omega(\boldsymbol{w}),$$

where $f$ is a convex smooth function and $\Omega$ a sparsity-inducing norm. The working set, which we denote by $J$, refers to the subset of variables involved in the optimization of these subproblems.

Working-set algorithms proceed as follows: after computing a solution to the problem restricted to the variables in $J$ (while setting the variables in $J^c$ to zero), global optimality is checked to determine whether the algorithm has to continue. If this is the case, new variables enter the working set $J$ according to a strategy that has to be

defined. Note that we only consider *forward* algorithms, i.e., where the working set grows monotonically. In other words, there are no *backward* steps where variables would be allowed to leave the set $J$. Provided this assumption is met, it is easy to see that these procedures stop in a finite number of iterations.

This class of algorithms is typically applied to linear programming and quadratic programming problems (see, e.g., [104]), and here takes specific advantage of sparsity from a computational point of view [9, 66, 79, 106, 116, 121, 132], since the subproblems that need to be solved are typically much smaller than the original one.

Working-set algorithms require three ingredients:

• **Inner-loop solver**: At each iteration of the working-set algorithm, problem (1.1) has to be solved on $J$, i.e., subject to the additional equality constraint that $\boldsymbol{w}_j = 0$ for all $j$ in $J^c$:

$$\min_{\boldsymbol{w}\in\mathbb{R}^p} f(\boldsymbol{w}) + \lambda\Omega(\boldsymbol{w}), \quad \text{such that } \boldsymbol{w}_{J^c} = 0. \tag{6.1}$$

The computation can be performed by any of the methods presented in this monograph. Working-set algorithms should therefore be viewed as "meta-algorithms". Since solutions for successive working sets are typically close to each other the approach is efficient if the method chosen can use *warm-restarts*.

• **Computing the optimality conditions**: Given a solution $\boldsymbol{w}^\star$ of problem (6.1), it is then necessary to check whether $\boldsymbol{w}^\star$ is also a solution for the original problem (1.1). This test relies on the duality gaps of problems (6.1) and (1.1). In particular, if $\boldsymbol{w}^\star$ is a solution of problem (6.1), it follows from Proposition 1.6 in Section 1.4 that

$$f(\boldsymbol{w}^\star) + \lambda\Omega(\boldsymbol{w}^\star) + f^*(\nabla f(\boldsymbol{w}^\star)) = 0.$$

In fact, the Lagrangian parameter associated with the equality constraint ensures the feasibility of the dual variable formed from the gradient of $f$ at $\boldsymbol{w}^\star$. In turn, this guarantees that the duality gap of problem (6.1) vanishes. The candidate $\boldsymbol{w}^\star$ is now a solution of the full problem (1.1), i.e., without the equality constraint $\boldsymbol{w}_{J^c} = 0$, if and only if

$$\Omega^*(\nabla f(\boldsymbol{w}^\star)) \leq \lambda. \tag{6.2}$$

Condition (6.2) points out that the dual norm $\Omega^*$ is a key quantity to monitor the progress of the working-set algorithm [66]. In simple settings, for instance when $\Omega$ is the $\ell_1$-norm, checking condition (6.2) can be easily computed since $\Omega^*$ is just the $\ell_\infty$-norm. In this case, condition (6.2) becomes

$$|[\nabla f(\boldsymbol{w}^\star)]_j| \leq \lambda, \quad \text{for all } j \text{ in } \{1, \ldots, p\}.$$

Note that by using the optimality of problem (6.1), the components of the gradient of $f$ indexed by $J$ are already guaranteed to be no greater than $\lambda$.

• **Strategy for the growth of the working set**: If condition (6.2) is not satisfied for the current working set $J$, some inactive variables in $J^c$ have to become active. This point raises the questions of *how many* and *how* these variables should be chosen. First, depending on the structure of $\Omega$, a *single* or a *group* of inactive variables have to be considered to enter the working set. Furthermore, one natural way to proceed is to look at the variables that violate condition (6.2) most. In the example of $\ell_1$-regularized least squares regression with normalized predictors, this strategy amounts to selecting the inactive variable that has the highest correlation with the current residual.

The working-set algorithms we have described so far aim at solving problem (1.1) for a fixed value of the regularization parameter $\lambda$. However, for specific types of loss and regularization functions, the set of solutions of problem (1.1) can be obtained efficiently for all possible values of $\lambda$, which is the topic of the next section.

## 6.2 Homotopy Methods

We present in this section an active-set[1] method for solving the Lasso problem of Equation (1.8). We recall the Lasso formulation:

$$\min_{\boldsymbol{w} \in \mathbb{R}^p} \frac{1}{2n} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}\|_2^2 + \lambda \|\boldsymbol{w}\|_1, \tag{6.3}$$

where $\boldsymbol{y}$ is in $\mathbb{R}^n$, and $\boldsymbol{X}$ is a design matrix in $\mathbb{R}^{n \times p}$. Even though generic working-set methods introduced above could be used to solve

[1] Active-set and working-set methods are very similar; they differ in that active-set methods allow (or sometimes require) variables returning to zero to exit the set.

this formulation (see, e.g., [79]), a specific property of the $\ell_1$-norm associated with a quadratic loss makes it possible to address it more efficiently.

Under mild assumptions (which we will detail later), the solution of Equation (6.3) is unique, and we denote it by $\boldsymbol{w}^\star(\lambda)$ for a given regularization parameter $\lambda > 0$. We use the name *regularization path* to denote the function $\lambda \mapsto \boldsymbol{w}^\star(\lambda)$ that associates to a regularization parameter $\lambda$ the corresponding solution. We will show that this function is piecewise linear, a behavior illustrated in Figure 6.1, where the entries of $\boldsymbol{w}^\star(\lambda)$ for a particular instance of the Lasso are represented as functions of $\lambda$.

An efficient algorithm can thus be constructed by choosing a particular value of $\lambda$, for which finding this solution is trivial, and by following the piecewise affine path, computing the directions of the current affine parts, and the points where the direction changes (also known as kinks). This piecewise linearity was first discovered and exploited in [90] in the context of portfolio selection, revisited in [108] describing a *homotopy* algorithm, and studied in [44] with the LARS algorithm.[2]

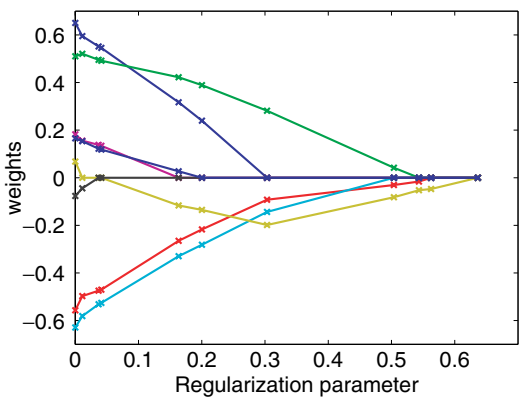



Fig. 6.1. The weights $\boldsymbol{w}^\star(\lambda)$ are represented as functions of the regularization parameter $\lambda$. When $\lambda$ increases, more and more coefficients are set to zero. These functions are all piecewise affine. Note that some variables (here one) may enter and leave the regularization path.

[2] Even though the basic version of LARS is a bit different from the procedure we have just described, it is closely related, and indeed a simple modification makes it possible to obtain the full regularization path of Equation (1.8).

Similar ideas also appear early in the optimization literature: Finding the full regularization path of the Lasso is in fact a particular instance of a *parametric quadratic programming* problem, for which path following algorithms have been developed [114].

Let us show how to construct the path. From the optimality conditions we have presented in Equation (1.9), denoting by $J := \{j; |\boldsymbol{X}_j^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}^\star)| = n\lambda\}$ the set of active variables, and defining the vector $\boldsymbol{t}$ in $\{-1; 0; 1\}^p$ as $\boldsymbol{t} := \operatorname{sign}\big(\boldsymbol{X}^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}^\star)\big)$, we have the following closed-form expression

$$\begin{cases} \boldsymbol{w}_J^\star(\lambda) = (\boldsymbol{X}_J^\top \boldsymbol{X}_J)^{-1}(\boldsymbol{X}_J^\top \boldsymbol{y} - n\lambda \boldsymbol{t}_J) \\ \boldsymbol{w}_{J^c}^\star(\lambda) = 0, \end{cases}$$

where we have assumed the matrix $\boldsymbol{X}_J^\top \boldsymbol{X}_J$ to be invertible (which is a sufficient condition to guarantee the uniqueness of $\boldsymbol{w}^\star$). This is an important point: if one knows in advance the set $J$ and the signs $\boldsymbol{t}_J$, then $\boldsymbol{w}^\star(\lambda)$ admits a simple closed-form. Moreover, when $J$ and $\boldsymbol{t}_J$ are fixed, the function $\lambda \mapsto (\boldsymbol{X}_J^\top \boldsymbol{X}_J)^{-1}(\boldsymbol{X}_J^\top \boldsymbol{y} - n\lambda \boldsymbol{t}_J)$ is affine in $\lambda$. With this observation in hand, we can now present the main steps of the path-following algorithm. It basically starts from a trivial solution of the regularization path, follows the path by exploiting this formula, updating $J$ and $\boldsymbol{t}_J$ whenever needed so that optimality conditions (1.9) remain satisfied. This procedure requires some assumptions — namely that (a) the matrix $\boldsymbol{X}_J^\top \boldsymbol{X}_J$ is always invertible, and (b) that updating $J$ along the path consists of adding or removing from this set a single variable at the same time. Concretely, we proceed as follows:

(1) Set $\lambda$ to $\frac{1}{n}\|\boldsymbol{X}^\top \boldsymbol{y}\|_\infty$ for which it is easy to show from Equation (1.9) that $\boldsymbol{w}^\star(\lambda) = 0$ (trivial solution on the regularization path).
(2) Set $J := \{j; |\boldsymbol{X}_j^\top \boldsymbol{y}| = n\lambda\}$.
(3) Follow the regularization path by decreasing the value of $\lambda$, with the formula $\boldsymbol{w}_J^\star(\lambda) = (\boldsymbol{X}_J^\top \boldsymbol{X}_J)^{-1}(\boldsymbol{X}_J^\top \boldsymbol{y} - n\lambda \boldsymbol{t}_J)$ keeping $\boldsymbol{w}_{J^c}^\star = 0$, until one of the following events (kink) occurs
    - There exists $j$ in $J^c$ such that $|\boldsymbol{X}_j^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}^\star)| = n\lambda$. Then, add $j$ to the set $J$.

- There exists $j$ in $J$ such that a non-zero coefficient $\boldsymbol{w}_j^\star$ hits zero. Then, remove $j$ from $J$.

We assume that only one of such events can occur at the same time (b). It is also easy to show that the value of $\lambda$ corresponding to the next event can be obtained in closed form such that one can "jump" from a kink to another.

(4) Go back to 3.

Let us now briefly discuss assumptions (a) and (b). When the matrix $\boldsymbol{X}_J^\top \boldsymbol{X}_J$ is not invertible, the regularization path is non-unique, and the algorithm fails. This can easily be fixed by addressing instead a slightly modified formulation. It is possible to consider instead the elastic-net formulation [159] that uses $\Omega(\boldsymbol{w}) = \lambda\|\boldsymbol{w}\|_1 + \frac{\gamma}{2}\|\boldsymbol{w}\|_2^2$. Indeed, it amounts to replacing the matrix $\boldsymbol{X}_J^\top \boldsymbol{X}_J$ by $\boldsymbol{X}_J^\top \boldsymbol{X}_J + n\gamma \boldsymbol{I}$, which is positive definite and therefore always invertible, and to apply the same algorithm. The second assumption (b) can be unsatisfied in practice because of the machine precision. To the best of our knowledge, the algorithm will fail in such cases, but we consider this scenario unlikely with real data, though possible when the Lasso/basis pursuit is used multiple times such as in dictionary learning, presented in Section 7.3. In such situations, a proper use of optimality conditions can detect such problems and more stable algorithms such as proximal methods may then be used.

The complexity of the above procedure depends on the number of kinks of the regularization path (which is also the number of iterations of the algorithm). Even though it is possible to build examples where this number is large, we often observe in practice that the event where one variable gets out of the active set is rare. The complexity also depends on the implementation. By maintaining the computations of $\boldsymbol{X}_j^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}^\star)$ and a Cholesky decomposition of $(\boldsymbol{X}_J^\top \boldsymbol{X}_J)^{-1}$, it is possible to obtain an implementation in $O(psn + ps^2 + s^3)$ operations, where $s$ is the sparsity of the solution when the algorithm is stopped (which we approximately consider as equal to the number of iterations). The product $psn$ corresponds to the computation of the matrices $\boldsymbol{X}_J^\top \boldsymbol{X}_J$, $ps^2$ to the updates of the correlations $\boldsymbol{X}_j^\top(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}^\star)$ along the path, and $s^3$ to the Cholesky decomposition.

# 7

# Sparsity and Nonconvex Optimization

In this section, we consider alternative approaches to sparse modelling, which are not based in convex optimization, but often use convex optimization problems in inner loops.

## 7.1 Greedy Algorithms

We consider the $\ell_0$-constrained signal decomposition problem

$$\min_{\boldsymbol{w}\in\mathbb{R}^p} \frac{1}{2n}\|\boldsymbol{y}-\boldsymbol{X}\boldsymbol{w}\|_2^2, \quad \text{s.t.} \quad \|\boldsymbol{w}\|_0 \leq s, \tag{7.1}$$

where $s$ is the desired number of non-zero coefficients of the solution, and we assume for simplicity that the columns of $\boldsymbol{X}$ have unit norm. Even though this problem can be shown to be NP-hard [96], greedy procedures can provide an approximate solution. Under some assumptions on the matrix $\boldsymbol{X}$, they can also be shown to have some optimality guarantees [136].

Several variants of these algorithms with different names have been developed both by the statistics and signal processing communities. In a nutshell, they are known as forward selection techniques in statistics (see [146]), and matching pursuit algorithms in signal

processing [89]. All of these approaches start with a null vector $\boldsymbol{w}$, and iteratively add non-zero variables to $\boldsymbol{w}$ until the threshold $s$ is reached.

The algorithm dubbed *matching pursuit*, was introduced in the 1990s in [89], and can be seen as a non-cyclic coordinate descent procedure for minimizing Equation (7.1). It selects at each step the column $\boldsymbol{x}^{\hat{\imath}}$ that is the most correlated with the residual according to the formula

$$\hat{\imath} \leftarrow \operatorname*{argmin}_{i \in \{1,\ldots,p\}} |\boldsymbol{r}^\top \boldsymbol{x}^i|,$$

where $\boldsymbol{r}$ denotes the residual $\boldsymbol{y} - \boldsymbol{X}\boldsymbol{w}$. Then, the residual is projected on $\boldsymbol{x}^{\hat{\imath}}$ and the entry $\boldsymbol{w}_{\hat{\imath}}$ is updated according to

$$\begin{aligned} \boldsymbol{w}_{\hat{\imath}} &\leftarrow \boldsymbol{w}_{\hat{\imath}} + \boldsymbol{r}^\top \boldsymbol{x}^{\hat{\imath}} \\ \boldsymbol{r} &\leftarrow \boldsymbol{r} - (\boldsymbol{r}^\top \boldsymbol{x}^{\hat{\imath}}) \boldsymbol{x}^{\hat{\imath}}. \end{aligned}$$

Such a simple procedure is guaranteed to decrease the objective function at each iteration, but is not to converge in a finite number of steps (the same variable can be selected several times during the process). Note that such a scheme also appears in statistics in boosting procedures [50].

*Orthogonal matching pursuit* [89] was proposed as a major variant of matching pursuit that ensures the residual of the decomposition to be always *orthogonal to all previously selected columns of* $\boldsymbol{X}$. Such technique existed in fact in the statistics literature under the name *forward selection* [146], and a particular implementation exploiting a QR matrix factorization also appears in [96]. More precisely, the algorithm is an active set procedure, which sequentially adds one variable at a time to the active set, which we denote by $J$. It provides an approximate solution of Equation (7.1) for every value $s' \leq s$, and stops when the desired level of sparsity is reached. Thus, it builds a regularization path, and shares many similarities with the homotopy algorithm for solving the Lasso [44], even though the two algorithms address different optimization problems. These similarities are also very strong in terms of implementation: identical tricks as those described in Section 6.2 for the homotopy algorithm can be used, and in fact both algorithms have roughly the same complexity (if most variables do not leave the path

once they have entered it). At each iteration, one has to choose which new predictor should enter the active set $J$. A possible choice is to look for the column of $\boldsymbol{X}$ most correlated with the residual as in the matching pursuit algorithm, but another criterion is to select the one that helps most reducing the objective function

$$\hat{\imath} \leftarrow \operatorname*{argmin}_{i \in J^c} \min_{\boldsymbol{w}' \in \mathbb{R}^{|J|+1}} \frac{1}{2n} \|\boldsymbol{y} - \boldsymbol{X}_{J \cup \{i\}} \boldsymbol{w}'\|_2^2.$$

whereas this choice seem at first sight computationally expensive since it requires solving $|J^c|$ least-squares problems, the solution can in fact be obtained efficiently using a Cholesky matrix decomposition of the matrix $\boldsymbol{X}_J^\top \boldsymbol{X}_J$ and basic linear algebra, which we will not detail here for simplicity reasons (see [40] for more details).

After this step, the active set is updated $J \leftarrow J \cup \{\hat{\imath}\}$, and the corresponding residual $\boldsymbol{r}$ and coefficients $\boldsymbol{w}$ are

$$\begin{aligned} \boldsymbol{w} &\leftarrow (\boldsymbol{X}_J^\top \boldsymbol{X}_J)^{-1} \boldsymbol{X}_J^\top \boldsymbol{y}, \\ \boldsymbol{r} &\leftarrow (\boldsymbol{I} - \boldsymbol{X}_J (\boldsymbol{X}_J^\top \boldsymbol{X}_J)^{-1} \boldsymbol{X}_J^\top) \boldsymbol{y}, \end{aligned}$$

where $\boldsymbol{r}$ is the residual of the orthogonal projection of $\boldsymbol{y}$ onto the linear subspace spanned by the columns of $\boldsymbol{X}_J$. It is worth noticing that one does not need to compute these two quantities in practice, but only updating the Cholesky decomposition of $(\boldsymbol{X}_J^\top \boldsymbol{X}_J)^{-1}$ and computing directly $\boldsymbol{X}^\top \boldsymbol{r}$, via simple linear algebra relations.

For simplicity, we have chosen to present matching pursuit algorithms for solving the $\ell_0$-sparse approximation problem, but they admit several variants (see [98] for example), or extensions when the regularization is more complex than the $\ell_0$-penalty or for other loss functions than the square loss. For instance, they are used in the context of non-convex group-sparsity in [137], or with structured sparsity formulations [15, 62].

We also remark that other possibilities than greedy methods exist for optimizing Equation (7.1). One can notably use the algorithm ISTA (i.e., the non-accelerated proximal method) presented in Section 3 when the function $f$ is convex and its gradient Lipschitz continuous. Under this assumption, it is easy to see that ISTA can iteratively decrease the value of the nonconvex objective function. Such proximal gradient

algorithms when $\Omega$ is the $\ell_0$-penalty often appear under the name of iterative hard-thresholding methods [59].

## 7.2 Reweighted-$\ell_1$ Algorithms with DC-Programming

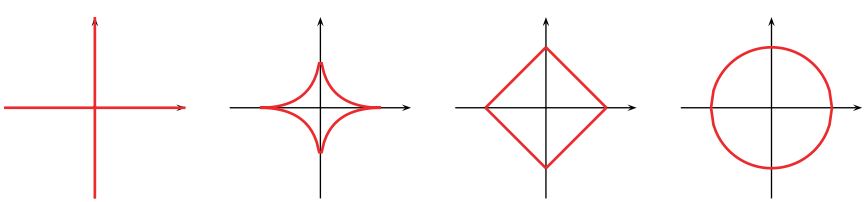
We focus in this section on optimization schemes for a certain type of nonconvex regularization functions. More precisely, we consider problem (1.1) when $\Omega$ is a nonconvex separable penalty that can be written as $\Omega(\boldsymbol{w}) := \sum_{i=1}^{p} \zeta(|\boldsymbol{w}_i|)$, where $\boldsymbol{w}$ is in $\mathbb{R}^p$, and $\zeta : \mathbb{R}_+ \to \mathbb{R}_+$ is a *concave* nondecreasing differentiable function. In other words, we address

$$\min_{\boldsymbol{w}\in\mathbb{R}^p} f(\boldsymbol{w}) + \lambda \sum_{i=1}^{p} \zeta(|\boldsymbol{w}_i|), \tag{7.2}$$

where $f$ is a convex smooth function. Examples of such penalties include variants of the $\ell_q$-penalties for $q < 1$ defined as $\zeta : t \mapsto (|t| + \varepsilon)^q$, log-penalties $\zeta : t \mapsto \log(|t| + \varepsilon)$, where $\varepsilon > 0$ makes the function $\zeta$ differentiable at 0. Other nonconvex regularization functions have been proposed in the statistics community, such as the SCAD penalty [47].

The main motivation for using such penalties is that they induce more sparsity than the $\ell_1$-norm, while they can be optimized with continuous optimization as opposed to greedy methods. The unit balls corresponding to the $\ell_q$-pseudo-norms and norms are illustrated in Figure 7.1, for several values of $q$. When $q$ decreases, the $\ell_q$-ball approaches in a sense the $\ell_0$-ball, which allows to induce sparsity more aggressively.

Even though the optimization problem (7.2) is not convex and not smooth, it is possible to iteratively decrease the value of the objective function by solving a sequence of convex problems. Algorithmic schemes



(a) $\ell_0$-ball. (b) $\ell_{0.5}$-ball. (c) $\ell_1$-ball. (d) $\ell_2$-ball.

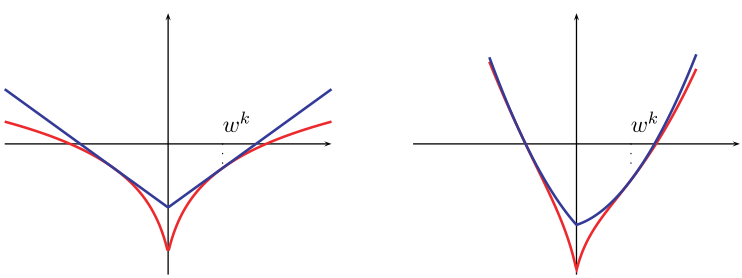
Fig. 7.1. Unit balls in 2D corresponding to $\ell_q$-penalties.

of this form appear early in the optimization literature in a more general framework for minimizing the difference of two convex functions (or equivalently the sum of a convex and a concave function), which is called DC programming (see [52] and references therein). Even though the objective function of Equation (7.2) is not exactly a difference of convex functions (it is only the case on $\mathbb{R}^p_+$), the core idea of DC programming can be applied. We note that these algorithms were recently revisited under the particular form of reweighted-$\ell_1$ algorithms [30]. The idea is relatively simple. We denote by $g\colon \mathbb{R}^p \to \mathbb{R}$ the objective function which can be written as $g(\boldsymbol{w}) := f(\boldsymbol{w}) + \lambda \sum_{i=1}^p \zeta(|\boldsymbol{w}_i|)$ for a vector $\boldsymbol{w}$ in $\mathbb{R}^p$. This optimization scheme consists in minimizing, at iteration $k$ of the algorithm, a convex upper bound of the objective function $g$, which is tangent to the graph of $g$ at the current estimate $\boldsymbol{w}^k$.

A surrogate function with these properties is obtained easily by exploiting the concavity of the functions $\zeta$ on $\mathbb{R}_+$, which always lie below their tangents, as illustrated in Figure 7.2. The iterative scheme can then be written as:

$$\boldsymbol{w}^{k+1} \leftarrow \operatorname*{argmin}_{\boldsymbol{w}\in\mathbb{R}^p} f(\boldsymbol{w}) + \lambda \sum_{i=1}^{p} \zeta'(|\boldsymbol{w}_i^k|)|\boldsymbol{w}_i|,$$

which is a reweighted-$\ell_1$ sparse decomposition problem [30]. To initialize the algorithm, the first step is usually a simple Lasso, with no




(a) red: $\zeta(|w|) = \log(|w| + \varepsilon)$.
blue: convex surrogate $\zeta'(|w^k|)|w| + C$.

(b) red: $f(w) + \zeta(|w|)$.
blue: $f(w) + \zeta'(|w^k|)|w| + C$.

Fig. 7.2. Functions and their surrogates involved in the reweighted-$\ell_1$ optimization scheme in the one dimensional case ($p = 1$). The function $\zeta$ can be written here as $\zeta(|w|) = \log(|w| + \varepsilon)$ for a scalar $w$ in $\mathbb{R}$, and the function $f$ is quadratic. The graph of the nonconvex functions are represented in red and their convex "tangent" surrogates in blue.

weights. In practice, the effect of the weights $\zeta'(|\boldsymbol{w}_i^k|)$ is to push to zero the smallest non-zero coefficients from iteration $k-1$, and two or three iterations are usually enough to obtain the desired sparsifying effect. Linearizing iteratively concave functions to obtain convex surrogates is the main idea of DC programming, which readily applies here to the functions $w \mapsto \zeta(|w|)$.

For simplicity we have presented these reweighted-$\ell_1$ algorithms when $\Omega$ is separable. We note however that these optimization schemes are far more general and readily apply to nonconvex versions of most of the penalties we have considered in this monograph. For example, when the penalty $\Omega$ has the form

$$\Omega(\boldsymbol{w}) = \sum_{g \in \mathcal{G}} \zeta(\|\boldsymbol{w}_g\|),$$

where $\zeta$ is concave and differentiable on $\mathbb{R}_+$, $\mathcal{G}$ is a set of (possibly overlapping) groups of variables and $\|.\|$ is any norm. The idea is then similar, iteratively linearizing for each group $g$ the functions $\zeta$ around $\|\boldsymbol{w}_g\|$ and minimizing the resulting convex surrogate (see an application to structured sparse principal component analysis in [70]).

Finally, another possible approach to solve optimization problems regularized by nonconvex penalties of the type presented in this section is to use the reweighted-$\ell_2$ algorithms of Section 5 based on quadratic variational forms of such penalties (see, e.g., [70]).

## 7.3 Sparse Matrix Factorization and Dictionary Learning

Sparse linear models for regression in statistics and machine learning assume a linear relation $\boldsymbol{y} \approx \boldsymbol{X}\boldsymbol{w}$, where $\boldsymbol{y}$ in $\mathbb{R}^n$ is a vector of observations, $\boldsymbol{X}$ in $\mathbb{R}^{n \times p}$ is a design matrix whose rows can be interpreted as features, and $\boldsymbol{w}$ is a weight vector in $\mathbb{R}^p$. Similar models are used in the signal processing literature, where $\boldsymbol{y}$ is a signal approximated by a linear combination of columns of $\boldsymbol{X}$, which are called dictionary elements, or basis element when $\boldsymbol{X}$ is orthogonal.

Whereas a lot of attention has been devoted to cases where $\boldsymbol{X}$ is fixed and pre-defined, another line of work considered the problem of learning $\boldsymbol{X}$ from training data. In the context of sparse linear models, this problem was first introduced in the neuroscience community by

Olshausen and Field [107] to model the spatial receptive fields of simple cells in the mammalian visual cortex. Concretely, given a training set of $q$ signals $\boldsymbol{Y} = [\boldsymbol{y}^1, \ldots, \boldsymbol{y}^q]$ in $\mathbb{R}^{n \times q}$, one would like to find a dictionary matrix $\boldsymbol{X}$ in $\mathbb{R}^{n \times p}$ and a coefficient matrix $\boldsymbol{W} = [\boldsymbol{w}^1, \ldots, \boldsymbol{w}^q]$ in $\mathbb{R}^{p \times q}$ such that each signal $\boldsymbol{y}^i$ admits a sparse approximation $\boldsymbol{X}\boldsymbol{w}^i$. In other words, we want to learn a dictionary $\boldsymbol{X}$ and a sparse matrix $\boldsymbol{W}$ such that $\boldsymbol{Y} \approx \boldsymbol{X}\boldsymbol{W}$.

A natural formulation is the following non-convex matrix factorization problem:

$$\min_{\boldsymbol{X} \in \mathcal{X}, \boldsymbol{W} \in \mathbb{R}^{n \times q}} \frac{1}{q} \sum_{i=1}^{q} \frac{1}{2} \|\boldsymbol{y}^i - \boldsymbol{X}\boldsymbol{w}^i\|_2^2 + \lambda \Omega(\boldsymbol{w}^i), \tag{7.3}$$

where $\Omega$ is a sparsity-inducing penalty function, and $\mathcal{X} \subseteq \mathbb{R}^{n \times p}$ is a convex set, which is typically the set of matrices whose columns have $\ell_2$-norm less than or equal to 1. Without any sparse prior (i.e., for $\lambda = 0$), the solution of this factorization problem is obtained through principal component analysis (PCA) (see, e.g., [28] and references therein). However, when $\lambda > 0$, the solution of Equation (7.3) has a different behavior, and may be used as an alternative to PCA for unsupervised learning.

A successful application of this approach is when the vectors $\boldsymbol{y}^i$ are small natural image patches, for example, of size $n = 10 \times 10$ pixels. A typical setting is to have an overcomplete dictionary — that is, the number of dictionary elements can be greater than the signal dimension but small compared to the number of training signals, for example $p = 200$ and $q = 100{,}000$. For this sort of data, dictionary learning finds linear subspaces of small dimension where the patches live, leading to effective applications in image processing [45]. Examples of a dictionary for image patches is given in Figure 7.3.

In terms of optimization, Equation (7.3) is nonconvex and no known algorithm has a guarantee of providing a global optimum in general, whatever the choice of penalty $\Omega$ is. A typical approach to find a local minimum is to use a block-coordinate scheme, which optimizes $\boldsymbol{X}$ and $\boldsymbol{W}$ in turn, while keeping the other one fixed [46]. Other alternatives include the K-SVD algorithm [3] (when $\Omega$ is the $\ell_0$-penalty), and online learning techniques [86, 107] that have proven to be particularly

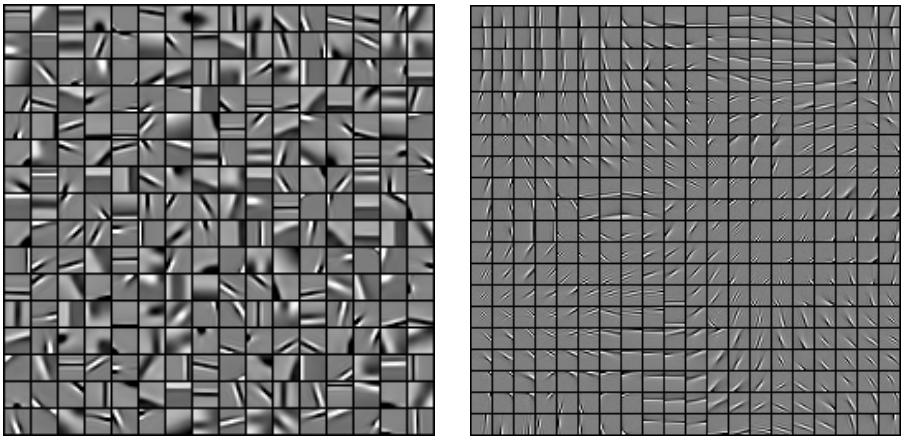


Fig. 7.3. Left: Example of dictionary with $p = 256$ elements, learned on a database of natural $12 \times 12$ image patches when $\Omega$ is the $\ell_1$-norm. Right: Dictionary with $p = 400$ elements, learned with a structured sparsity-inducing penalty $\Omega$ (see [88]).

efficient when the number of signals $q$ is large.[1] Convex relaxations of dictionary learning have also been proposed in [14, 26].

## 7.4 Bayesian Methods

While the focus of this monograph is on frequentist approaches to sparsity, and particularly on approaches that minimize a regularized empirical risk, there naturally exist several Bayesian[2] approaches to sparsity.

As a first remark, regularized optimization can be viewed as solving a maximum a posteriori (MAP) estimation problem if the loss $\ell$ (cf. Section 1.2) defining $f$ can be interpreted as a log-likelihood and the norm as certain log-prior. Typically, the $\ell_1$-norm can for instance be interpreted as the logarithm of a product of independent Laplace priors on the loading vectors $\mathbf{w}$ (see, e.g., [123]). However, the Laplace distribution is actually not a sparse prior, in the sense that it is a continuous

[1] Such efficient algorithms are freely available in the open-source software package SPAMS `http://www.di.ens.fr/willow/SPAMS/`.

[2] Bayesian methods can of course not be reduced to nonconvex optimization, but given that they are often characterized by multimodality and that corresponding variational formulations are typically nonconvex, we conveniently discuss them here.

distribution whose samples are thus nonzero almost surely. Besides the fact that MAP estimation is generally not considered as a Bayesian method (the fundamental principle of Bayesian method is to integrate over the uncertainty and avoid point estimates), evidence in the literature suggests that MAP is not a good principle to yield estimators that are adapted to the corresponding prior. In particular, the Lasso does in fact not provide a good algorithm to estimate vectors whose coefficients follow a Laplace distribution [56]!

To obtain exact zeros in a Bayesian setting, one must use so called "spike and slab" priors [63]. Inference with such priors leads to nonconvex optimization problems, and sampling methods, while also simple to implement, do not come with any guarantees, in particular in high-dimensional settings.

In reality, while obtaining exact zeros can be valuable from a computational point of view, it is a priori not necessary to obtain theoretical guarantees associated with sparse methods. In fact, sparsity should be understood as the requirement or expectation that a few coefficients are significantly larger than most of the rest, an idea which is somehow formalized as *compressibility* in the compressed sensing literature, and which inspires *automatic relevance determination* methods (ARD) and the use of *heavy-tail priors* among Bayesian approaches [97, 148]. Using heavy-tailed prior distribution on $\mathbf{w}_i$ allows to obtain posterior estimates with many small values and a few large values, this effect being stronger when the tails are heavier, in particular with Student's $t$-distribution. Heavy-tailed distributions and ARD are very related since these prior distributions can be expressed as scaled mixture of Gaussians [5, 31]. This is of interest computationally, in particular for variational methods.

Variable selection is also obtained in a Bayesian setting by optimizing the marginal likelihood of the data over the hyper-parameters, that is using *empirical Bayes* estimation; in that context iterative methods based on DC programming may be efficiently used [147].

It should be noted that the heavy-tail prior formulation points to an interesting connection between sparsity and the notion of robustness in statistics, in which a sparse subset of the data is allowed to take large values. This is also suggested by works such as [149, 154].

# 8

# Quantitative Evaluation

To illustrate and compare the methods presented in this monograph, we consider in this section three benchmarks. These benchmarks are chosen to be representative of problems regularized with sparsity-inducing norms, involving different norms and different loss functions. To make comparisons that are as fair as possible, each algorithm is implemented in `C/C++`, using efficient BLAS and LAPACK libraries for basic linear algebra operations. Most of these implementations have been made available in the open-source software SPAMS.[1] All subsequent simulations are run on a single core of a 3.07 GHz CPU, with 8 GB of memory. In addition, we take into account several criteria which strongly influence the convergence speed of the algorithms. In particular, we consider

(a) different problem scales,
(b) different levels of correlations between input variables,
(c) different strengths of regularization.

We also show the influence of the required precision by monitoring the time of computation as a function of the objective function.

[1] `http://www.di.ens.fr/willow/SPAMS/`.

For the convenience of the reader, we list here the algorithms compared and the acronyms we use to refer to them throughout this section: the homotopy/LARS algorithm (LARS), coordinate-descent (CD), reweighted-$\ell_2$ schemes (Re-$\ell_2$), simple proximal method (ISTA) and its accelerated version (FISTA). Note that all methods except the working set methods are very simple to implement as each iteration is straightforward (for proximal methods such as FISTA or ISTA, as long as the proximal operator may be computed efficiently). On the contrary, as detailed in Section 6.2, homotopy methods require some care in order to achieve the performance we report in this section.

We also include in the comparisons generic algorithms such as a subgradient descent algorithm (SG), and a commercial software[2] for cone (CP), quadratic (QP) and second-order cone programming (SOCP) problems.

## 8.1 Speed Benchmarks for Lasso

We first present a large benchmark evaluating the performance of various optimization methods for solving the Lasso.

We perform small-scale ($n = 200, p = 200$) and medium-scale ($n = 2000, p = 10{,}000$) experiments. We generate design matrices as follows. For the scenario with low correlations, all entries of $\boldsymbol{X}$ are independently drawn from a Gaussian distribution $\mathcal{N}(0, 1/n)$, which is a setting often used to evaluate optimization algorithms in the literature. For the scenario with large correlations, we draw the rows of the matrix $\boldsymbol{X}$ from a multivariate Gaussian distribution for which the *average absolute value* of the correlation between two different columns is eight times the one of the scenario with low correlations. Test data vectors are taken of the form $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w} + \boldsymbol{n}$ where $\boldsymbol{w}$ are randomly generated, with two levels of sparsity to be used with the two different levels of regularization; $\boldsymbol{n}$ is a noise vector whose entries are i.i.d. samples from a Gaussian distribution $\mathcal{N}(0, 0.01\|\boldsymbol{X}\boldsymbol{w}\|_2^2/n)$. In the low regularization setting the sparsity of the vectors $\boldsymbol{w}$ is $s = 0.5\min(n, p)$, and, in the high regularization one, $s = 0.01\min(n, p)$, corresponding to fairly sparse vectors.

[2] Mosek, available at `http://www.mosek.com/`.

For SG, we take the step size to be equal to $a/(k+b)$, where $k$ is the iteration number, and $(a,b)$ are the best[3] parameters selected on a logarithmic grid $(a,b) \in \{10^3,\dots,10\} \times \{10^2,10^3,10^4\}$; we proceeded this way not to disadvantage SG by an arbitrary choice of stepsize.

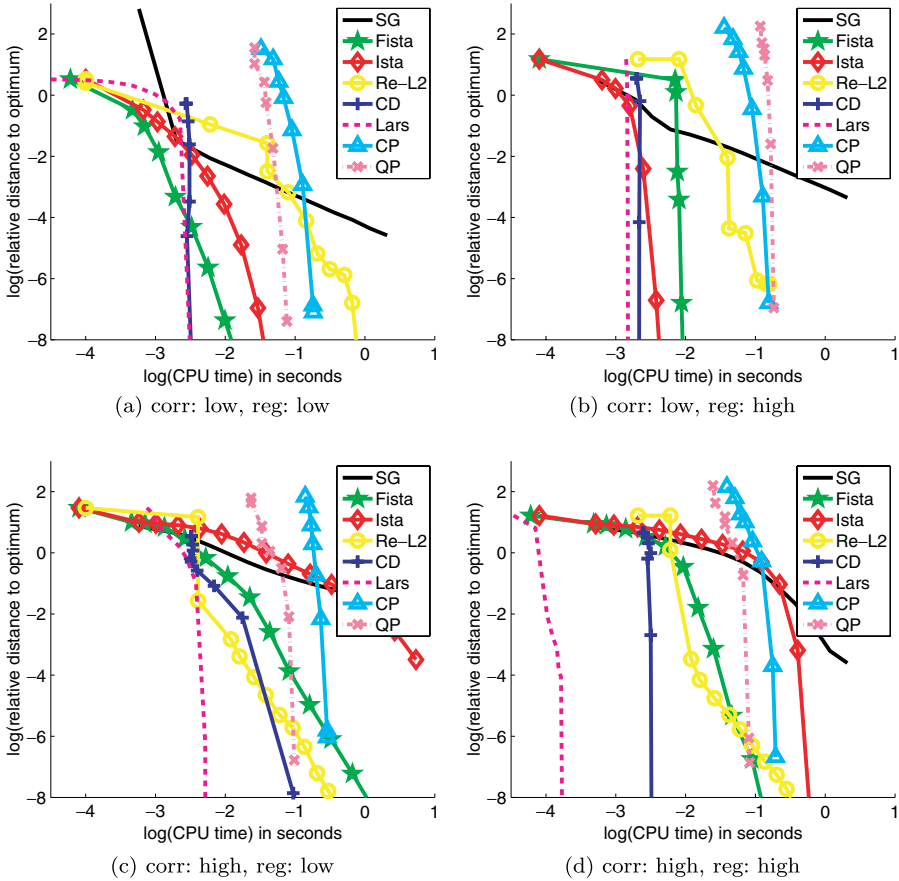
To sum up, we make a comparison for eight different conditions (2 scales $\times$ 2 levels of correlation $\times$ 2 levels of regularization). All results are reported on Figures 8.1, 8.2, by averaging 5 runs for each experiment. Interestingly, we observe that the relative performance of the different methods change significantly with the scenario.

Our conclusions for the different methods are as follows:

- **LARS/homotopy methods**: For the small-scale problem, LARS outperforms all other methods for almost every scenario and precision regime. It is therefore *definitely the right choice for the small-scale setting.* Unlike first-order methods, its performance does not depend on the correlation of the design matrix $\boldsymbol{X}$, but rather on the sparsity $s$ of the solution. In our larger scale setting, it has been competitive either when the solution is *very sparse* (high regularization), or when there is *high correlation* in $\boldsymbol{X}$ (in that case, other methods do not perform as well). More importantly, the homotopy algorithm gives an exact solution and computes the regularization path.
- **Proximal methods (ISTA, FISTA)**: FISTA outperforms ISTA in all scenarios but one. Both methods are close for high regularization or low correlation, but FISTA is significantly better for high correlation or/and low regularization. These methods are almost always outperformed by LARS in the small-scale setting, except for *low precision and low correlation.*

  Both methods *suffer from correlated features*, which is consistent with the fact that their convergence rate depends on the correlation between input variables (convergence as a geometric sequence when the correlation matrix is invertible,

[3] "The best step size" is understood here as being the step size leading to the smallest objective function after 500 iterations.



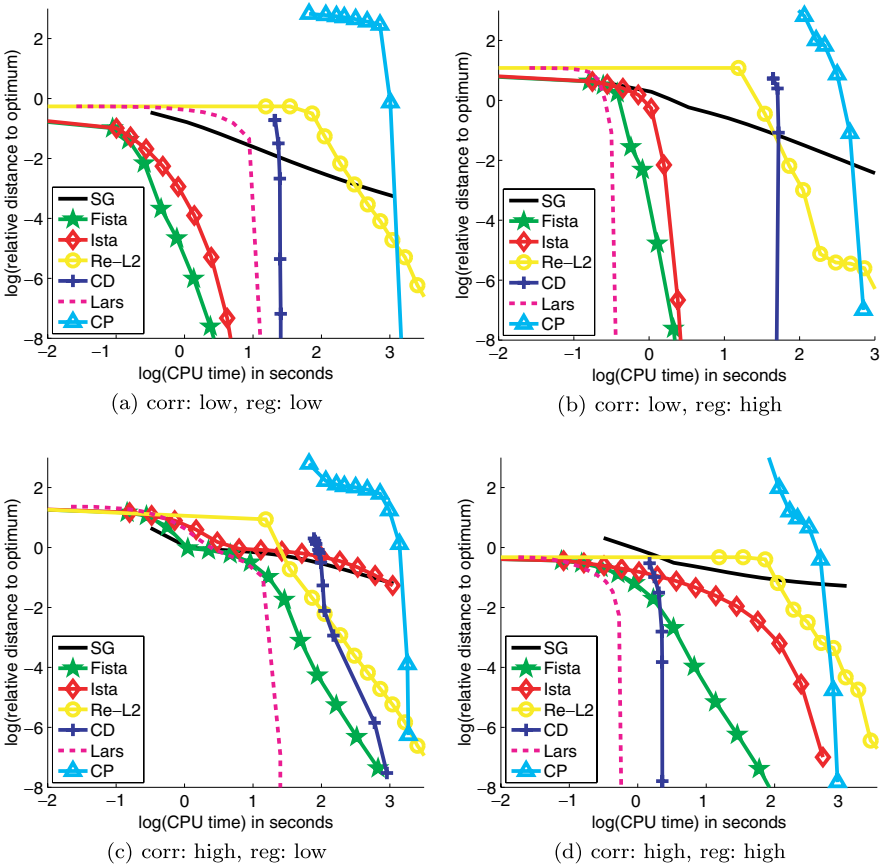

(a) corr: low, reg: low (b) corr: low, reg: high

(c) corr: high, reg: low (d) corr: high, reg: high

Fig. 8.1. Benchmark for solving the Lasso for the small-scale experiment ($n = 200$, $p = 200$), for the two levels of correlation and two levels of regularization, and eight optimization methods (see main text for details). The curves represent the relative value of the objective function as a function of the computational time in second on a $\log_{10}/\log_{10}$ scale.

and as the inverse of a degree-two polynomial otherwise). They are *well adapted to large-scale settings, with low or medium correlation.*

- **Coordinate descent (CD)**: The theoretical analysis of these methods suggest that they behave in a similar way to proximal methods [103, 124]. However, empirically, we have observed that the behavior of CD often consists of a first "warm-up" stage followed by a fast convergence phase.




(a) corr: low, reg: low (b) corr: low, reg: high

(c) corr: high, reg: low (d) corr: high, reg: high

Fig. 8.2. Benchmark for solving the Lasso for the medium-scale experiment $n = 2{,}000$, $p = 10{,}000$, for the two levels of correlation and two levels of regularization, and eight optimization methods (see main text for details). The curves represent the relative value of the objective function as a function of the computational time in second on a $\log_{10}/\log_{10}$ scale.

Its performance in the *small-scale setting is competitive* (even though always behind LARS), but *less efficient in the large-scale one*. For a reason we cannot explain, *it suffers less than proximal methods from correlated features.*

- **Reweighted-$\ell_2$**: This method was outperformed in all our experiments by other dedicated methods.[4] Note that we

[4] Note that the reweighted-$\ell_2$ scheme requires solving iteratively large-scale linear system that are badly conditioned. Our implementation uses LAPACK Cholesky decompositions,

considered only the smoothed alternating scheme of Section 5 and not first order methods in $\boldsymbol{\eta}$ such as that of [110]. A more exhaustive comparison should include these as well.

- **Generic Methods (SG, QP, CP)**: As expected, generic methods are not adapted for solving the Lasso and are always outperformed by dedicated ones such as LARS.

Among the methods that we have presented, some require an initial overhead computation of the Gram matrix $\boldsymbol{X}^\top \boldsymbol{X}$: this is the case for coordinate descent and reweighted-$\ell_2$ methods. We took into account this overhead time in all figures, which explains the behavior of the corresponding convergence curves. Like homotopy methods, these methods could also benefit from an offline pre-computation of $\boldsymbol{X}^\top \boldsymbol{X}$ and would therefore be more competitive if the solutions corresponding to several values of the regularization parameter have to be computed.

We have considered in the above experiments the case of the square loss. Obviously, some of the conclusions drawn above would not be valid for other smooth losses. On the one hand, the LARS is no longer applicable; on the other hand, proximal methods are clearly still available and coordinate descent schemes, which were dominated by the LARS in our experiments, turn out to be very good contenders in that setting as we illustrate in the next Section.

## 8.2 Group-Sparsity for Multi-Task Learning

For $\ell_1$-regularized least-squares regression, homotopy methods have appeared in the previous section as one of the best techniques, in almost all the experimental conditions.

This second speed benchmark explores a setting where this homotopy approach cannot be applied anymore. In particular, we consider a multi-class classification problem in the context of cancer diagnosis. We address this problem from a multi-task viewpoint [106]. To this end, we take the regularizer to be $\ell_1/\ell_2$- and $\ell_1/\ell_\infty$-norms, with (nonoverlapping) groups of variables penalizing features across all classes [82, 106].

but a better performance might be obtained using a pre-conditioned conjugate gradient, especially in the very large scale setting.

As a data-fitting term, we now choose a simple "1-vs-all" logistic loss function.

We focus on two multi-class classification problems in the "small $n$, large $p$" setting, based on two datasets[5] of gene expressions. The medium-scale dataset contains $n = 83$ observations, $p = 2308$ variables and 4 classes, while the large-scale one contains $n = 308$ samples, $p = 15{,}009$ variables and 26 classes. Both datasets exhibit highly correlated features.

In addition to ISTA, FISTA, and SG, we consider here the block coordinate-descent (BCD) from [140] presented in Section 4. We also consider a working-set strategy on top of BCD, that optimizes over the full set of features (including the non-active ones) only once every four iterations. As further discussed in Section 4, it is worth mentioning that the multi-task setting is well suited for the method of [140] since an appropriate approximation of the Hessian can be easily computed.

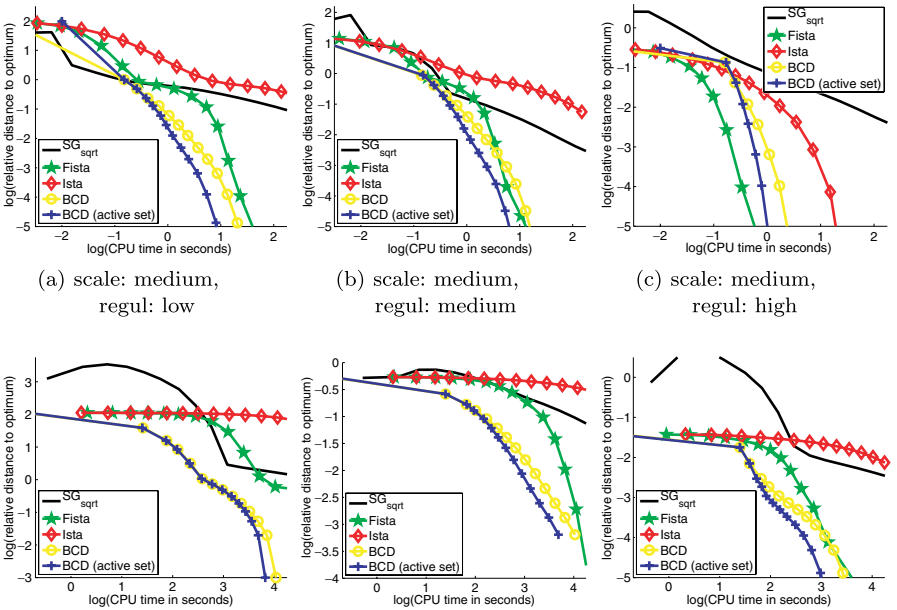
All the results are reported in Figures 8.3 and 8.4. As expected in the light of the benchmark for the Lasso, BCD appears as the best option, regardless of the sparsity/scale conditions.

## 8.3 Structured Sparsity

In this second series of experiments, the optimization techniques of the previous sections are further evaluated when applied to other types of loss and sparsity-inducing functions. Instead of the $\ell_1$-norm previously studied, we focus on the particular *hierarchical* $\ell_1/\ell_2$-norm $\Omega$ introduced in Section 3.

From an optimization standpoint, although $\Omega$ shares some similarities with the $\ell_1$-norm (e.g., the convexity and the nonsmoothness), it differs in that it cannot be decomposed into independent parts (because of the overlapping structure of $\mathcal{G}$). Coordinate descent schemes hinge on this property and as a result, cannot be straightforwardly applied in this case.

[5] The two datasets we use are *SRBCT* and *14_Tumors*, which are freely available at `http://www.gems-system.org/`.




(a) scale: medium, regul: low

(b) scale: medium, regul: medium

(c) scale: medium, regul: high

(d) scale: large, regul: low

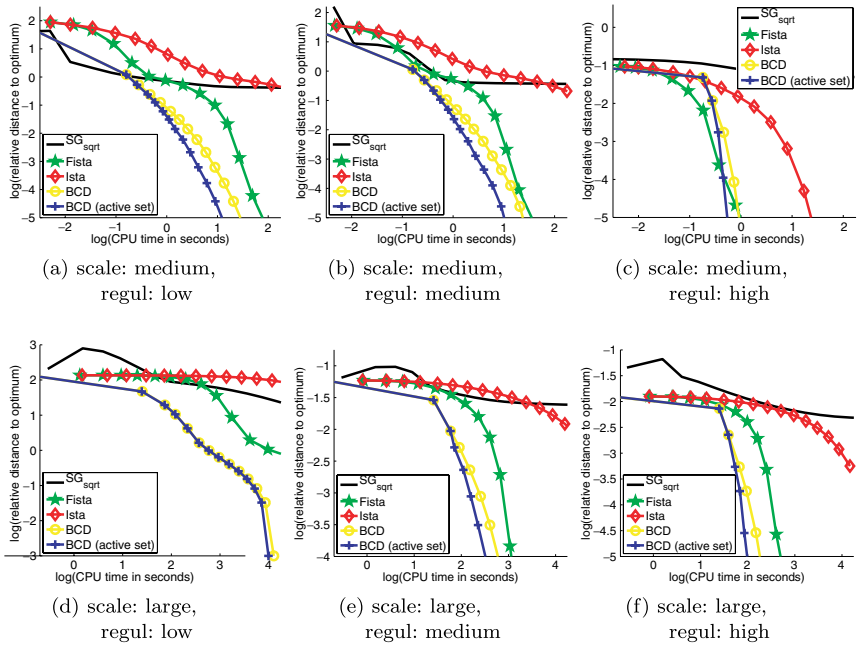
(e) scale: large, regul: medium

(f) scale: large, regul: high

Fig. 8.3. Medium- and large-scale multi-class classification problems with an $\ell_1/\ell_2$-regularization, for three optimization methods (see details about the datasets and the methods in the main text). Three levels of regularization are considered. The curves represent the relative value of the objective function as a function of the computation time in second on a $\log_{10}/\log_{10}$ scale. In the highly regularized setting, the tuning of the step-size for the subgradient turned out to be difficult, which explains the behavior of SG in the first iteration.

### 8.3.1 Denoising of Natural Image Patches

In this first benchmark, we consider a least-squares regression problem regularized by $\Omega$ that arises in the context of the denoising of natural image patches [68]. In particular, based on a hierarchical set of features that accounts for different types of edge orientations and frequencies in natural images, we seek to reconstruct noisy $16 \times 16$-patches. Although the problem involves a small number of variables (namely $p = 151$), it has to be solved repeatedly for thousands of patches, at moderate precision. It is therefore crucial to be able to solve this problem efficiently.



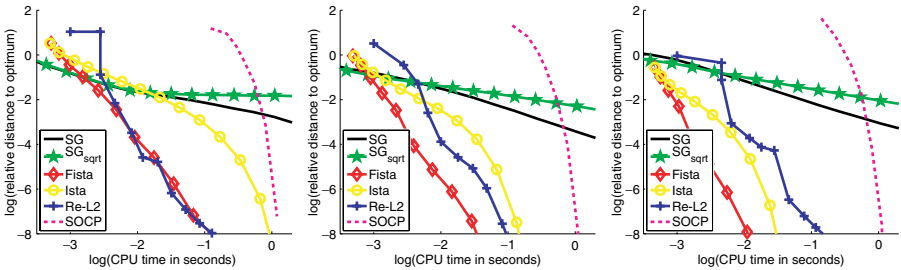

(a) scale: medium, regul: low (b) scale: medium, regul: medium (c) scale: medium, regul: high

(d) scale: large, regul: low (e) scale: large, regul: medium (f) scale: large, regul: high

Fig. 8.4. Medium- and large-scale multi-class classification problems with an $\ell_1/\ell_\infty$-regularization for three optimization methods (see details about the datasets and the methods in the main text). Three levels of regularization are considered. The curves represent the relative value of the objective function as a function of the computation time in second on a $\log_{10}/\log_{10}$ scale. In the highly regularized setting, the tuning of the step-size for the subgradient turned out to be difficult, which explains the behavior of SG in the first iterations.

The algorithms that take part in the comparisons are ISTA, FISTA, Re-$\ell_2$, SG, and SOCP. All results are reported in Figure 8.5, by averaging 5 runs.

We can draw several conclusions from the simulations. First, we observe that across all levels of sparsity, the accelerated proximal scheme performs better, or similarly, than the other approaches. In addition, as opposed to FISTA, ISTA seems to suffer in nonsparse scenarios. In the least sparse setting, the reweighted-$\ell_2$ scheme matches the performance of FISTA. However this scheme does not yield truly sparse solutions, and would therefore require a subsequent thresholding operation, which can be difficult to motivate in a principled way. As expected, the generic techniques such as SG and SOCP do not compete with the dedicated algorithms.



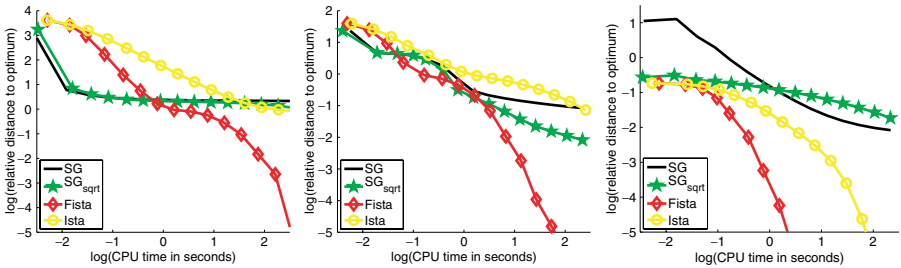

(a) scale: small, regul: low (b) scale: small, regul: medium (c) scale: small, regul: high

Fig. 8.5. Benchmark for solving a least-squares regression problem regularized by the hierarchical norm $\Omega$. The experiment is small scale, $n = 256, p = 151$, and shows the performances of five optimization methods (see main text for details) for three levels of regularization. The curves represent the relative value of the objective function as a function of the computational time in second on a $\log_{10} / \log_{10}$ scale.

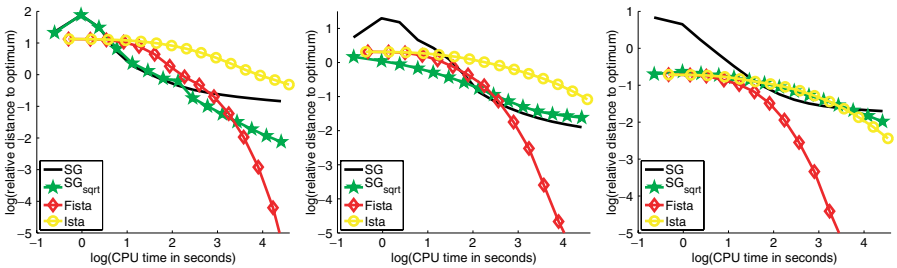
### 8.3.2 Multi-class Classification of Cancer Diagnosis

This benchmark focuses on multi-class classification of cancer diagnosis and reuses the two datasets from the multi-task problem of Section 8.2. Inspired by [73], we build a tree-structured set of groups of features $\mathcal{G}$ by applying Ward's hierarchical clustering [71] on the gene expressions. The norm $\Omega$ built that way aims at capturing the hierarchical structure of gene expression networks [73]. For more details about this construction, see [67] in the context of neuroimaging. The resulting datasets with tree-structured sets of features contain $p = 4615$ and $p = 30{,}017$ variables, for respectively the medium- and large-scale datasets.

Instead of the square loss function, we consider the multinomial logistic loss function, which is better suited for multi-class classification problems. As a direct consequence, the algorithms whose applicability crucially depends on the choice of the loss function are removed from the benchmark. This is for instance the case for reweighted-$\ell_2$ schemes that have closed-form updates available only with the square loss (see Section 5). Importantly, the choice of the multinomial logistic loss function requires one to optimize over a matrix with dimensions $p$ times the number of classes (i.e., a total of $4615 \times 4 \approx 18{,}000$ and $30{,}017 \times 26 \approx 780{,}000$ variables). Also, for lack of scalability, generic interior point solvers could not be considered here. To summarize, the




(a) scale: medium, regul: low (b) scale: medium, regul: medium (c) scale: medium, regul: high




(d) scale: large, regul: low (e) scale: large, regul: medium (f) scale: large, regul: high

Fig. 8.6. Medium- and large-scale multi-class classification problems for three optimization methods (see details about the datasets and the methods in the main text). Three levels of regularization are considered. The curves represent the relative value of the objective function as a function of the computation time in second on a $\log_{10}/\log_{10}$ scale. In the highly regularized setting, the tuning of the step-size for the subgradient turned out to be difficult, which explains the behavior of SG in the first iterations.

following comparisons involve ISTA, FISTA, and SG; for subgradient descent, we set the step size either to $a/(k+b)$ or to $a/(\sqrt{k}+b)$ (in which case we refer to it in figure legends respectively as SG and $\text{SG}_{\text{sqrt}}$), where $k$ is the iteration number, and $(a, b)$ are best parameters selected on a logarithmic grid....

All the results are reported in Figure 8.6. The benchmark especially points out that the accelerated proximal scheme performs overall better than the two other methods. Again, it is important to note that both proximal algorithms yield sparse solutions, which is not the case for SG. More generally, this experiment illustrates the flexibility of proximal algorithms with respect to the choice of the loss function.

### 8.3.3 General Overlapping Groups of Variables

We consider a structured sparse decomposition problem with overlapping groups of $\ell_\infty$-norms, and compare the proximal gradient algorithm FISTA [17] consider the proximal operator presented in Section 3.3 (referred to as ProxFlow [87]). Since, the norm we use is a sum of several simple terms, we can bring to bear other optimization techniques which are dedicated to this situation, namely proximal splitting method known as alternating direction method of multipliers (ADMM) (see, e.g., [24, 38]). We consider two variants, (ADMM) and (Lin-ADMM) — see more details in [88].

We consider a design matrix $\boldsymbol{X}$ in $\mathbb{R}^{n\times p}$ built from overcomplete dictionaries of discrete cosine transforms (DCT), which are naturally organized on one- or two-dimensional grids and display local correlations. The following families of groups $\mathcal{G}$ using this spatial information are thus considered: (1) every contiguous sequence of length 3 for the one-dimensional case, and (2) every $3\times 3$-square in the two-dimensional setting. We generate vectors $\boldsymbol{y}$ in $\mathbb{R}^n$ according to the linear model $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{w}_0 + \boldsymbol{\varepsilon}$, where $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, 0.01\|\boldsymbol{X}\boldsymbol{w}_0\|_2^2)$. The vector $\boldsymbol{w}_0$ has about 20% nonzero components, randomly selected, while respecting the structure of $\mathcal{G}$, and uniformly generated in $[-1, 1]$.

In our experiments, the regularization parameter $\lambda$ is chosen to achieve the same level of sparsity (20%). For SG, ADMM and Lin-ADMM, some parameters are optimized to provide the lowest value of the objective function after 1000 iterations of the respective algorithms. For SG, we take the step size to be equal to $a/(k+b)$, where $k$ is the iteration number, and $(a, b)$ are the pair of parameters selected in $\{10^{-3}, \ldots, 10\} \times \{10^2, 10^3, 10^4\}$. The parameter $\gamma$ for ADMM is selected in $\{10^{-2}, \ldots, 10^2\}$. The parameters $(\gamma, \delta)$ for Lin-ADMM are selected in $\{10^{-2}, \ldots, 10^2\} \times \{10^{-1}, \ldots, 10^8\}$. For interior point methods, since problem (1.1) can be cast either as a quadratic (QP) or as a conic program (CP), we show in Figure 8.7 the results for both formulations. On three problems of different sizes, with $(n, p) \in \{(100, 10^3), (1024, 10^4), (1024, 10^5)\}$, our algorithms ProxFlow, ADMM and Lin-ADMM compare favorably with the other methods, (see Figure 8.7), except for ADMM in the large-scale setting which yields

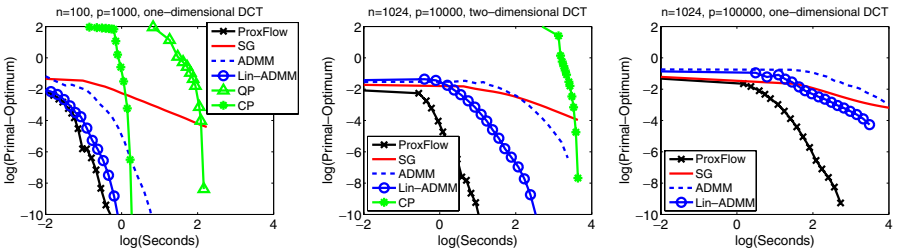



Fig. 8.7. Speed comparisons: distance to the optimal primal value versus CPU time (log–log scale). Due to the computational burden, QP and CP could not be run on every problem.

an objective function value similar to that of SG after $10^4$ seconds. Among ProxFlow, ADMM and Lin-ADMM, ProxFlow is consistently better than Lin-ADMM, which is itself better than ADMM. Note that for the small scale problem, the performance of ProxFlow and Lin-ADMM is similar. In addition, note that QP, CP, SG, ADMM and Lin-ADMM do not obtain sparse solutions, whereas ProxFlow does.

## 8.4 General Comments

We conclude this section by a couple of general remarks on the experiments that we presented. First, the use of proximal methods is often advocated because of their optimal worst case complexities in $O(\frac{1}{t^2})$ (where $t$ is the number of iterations). In practice, in our experiments, these and several other methods exhibit empirically convergence rates that are at least linear, if not better, which suggests that the adaptivity of the method (e.g., its ability to take advantage of local curvature) might be more crucial to its practical success. Second, our experiments concentrated on regimes that are of interest for sparse methods in machine learning where typically $p$ is larger than $n$ and where it is possible to find good sparse solutions. The setting where $n$ is much larger than $p$ was out of scope here, but would be worth a separate study, and should involve methods from stochastic optimization. Also, even though it might make sense from an optimization viewpoint, we did not consider problems with low levels of sparsity, that is with more dense solution vectors, since it would be a more difficult regime for many of the algorithms that we presented (namely LARS, CD or proximal methods).

# 9

# Extensions

We obviously could not cover exhaustively the literature on algorithms for sparse methods in this monograh.

Surveys and comparisons of algorithms for sparse methods have been proposed in [118] and [155]. These articles present quite a few algorithms, but focus essentially on $\ell_1$-regularization and unfortunately do not consider proximal methods. Also, it is not clear that the metrics used to compare the performance of various algorithms is the most relevant to machine learning; in particular, we present the full convergence curves that we believe are more informative than the ordering of algorithms at fixed precision.

Beyond the material presented here, there a few topics that we did not develop and that are worth mentioning.

In the section on proximal methods, we presented the proximal methods called forward–backward splitting methods. We applied them to objectives which are the sum of two terms: a differentiable function with Lipschitz-continuous gradients and a norm. More generally these methods apply to the sum of two semi-lower continuous (l.s.c.), proper, convex functions with non-empty domain, and where one element is assumed differentiable with Lipschitz-continuous gradient [38].

The proximal operator itself dates back to [94] and proximal methods themselves date back to [81, 91]. As of today, they have been extended to various settings [36, 38, 39, 138]. In particular, instances of proximal methods are still applicable if the smoothness assumptions that we made on the loss are relaxed. For example, the Douglas–Rachford splitting algorithm applies as soon as the objective function to minimize is only assumed l.s.c. proper convex, without any smoothness properties (although a l.s.c. convex function is continuous inside of its domain). The augmented Lagrangian techniques (see [24, 38, 54] and numerous references therein) and more precisely their variants known as alternating-direction methods of multipliers are related to proximal methods via duality. These methods are in particular applicable to cases where several regularizations and constraints are mixed [88, 135].

For certain combination of losses and regularizations, dedicated methods have been proposed. This is the case for linear regression with the least absolute deviation (LAD) loss (also called $\ell_1$-loss) with an $\ell_1$-norm regularizer, which leads to a linear program [152]. This is also the case for algorithms designed for classical multiple kernel learning when the regularizer is the squared norm [110, 128, 131]; these methods are therefore not exactly comparable to the MKL algorithms presented in this monograph which apply to objective regularized by the unsquared norm (except for reweighted $\ell_2$-schemes, based on variational formulations for the squared norm).

In the context of proximal methods, the metric used to define the proximal operator can be modified by judicious rescaling operations, in order to fit better the geometry of the data [43]. Moreover, they can be mixed with Newton and quasi-Newton methods, for further acceleration (see, e.g., [119]).

Finally, from a broader outlook, our — *a priori* deterministic — optimization problem (1.1) may also be tackled with stochastic optimization approaches, which has been the focus of much recent research [21, 23, 43, 60, 125, 153].

# 10

# Conclusions

We have tried to provide in this monograph a unified view of sparsity and structured sparsity as it can emerge when convex analysis and convex optimization are used as the conceptual basis to formalize respectively problems and algorithms. In that regard, we did not aim at exhaustivity and other paradigms are likely to provide complementary views.

With convexity as a requirement however, using non-smooth norms as regularizers is arguably the most natural way to encode sparsity constraints. A main difficulty associated with these norms is that they are intrinsically non-differentiable; they are however fortunately also structured, so that a few concepts can be leveraged to manipulate and solve problems regularized with them. To summarize:

— Fenchel–Legendre duality and the dual norm allow to compute subgradients, duality gaps and are also key to exploit sparsity algorithmically via working set methods. More trivially, duality also provides an alternative formulation to the initial problem which is sometimes more tractable.

— The proximal operator, when it can be computed efficiently (exactly or approximately), allows one to treat the optimization problem as if it were a smooth problem.
— Quadratic variational formulations provide an alternative way to decouple the difficulties associated with the loss and the nondifferentiability of the norm.

Leveraging these different tools led us to present and compare four families of algorithms for sparse methods: proximal methods, block-coordinate descent algorithms, reweighted-$\ell_2$ schemes and the LARS/homotopy algorithms that are representative of the state of the art. The properties of these methods can be summarized as follows:

— Proximal methods provide efficient and scalable algorithms that are applicable to a wide family of loss functions, that are simple to implement, compatible with many sparsity-inducing norms and often competitive with the other methods considered.
— For the square loss, the homotopy method remains the fastest algorithm for (a) small and medium scale problems, since its complexity depends essentially on the size of the active sets, (b) cases with very correlated designs. It computes the whole path up to a certain sparsity level. Its main drawback is that it is difficult to implement efficiently, and it is subject to numerical instabilities. On the other hand, coordinate descent and proximal algorithms are trivial to implement.
— For smooth losses, block-coordinate descent provides one of the fastest algorithms but it is limited to separable regularizers.
— For the square-loss and possibly sophisticated sparsity inducing regularizers, reweighted-$\ell_2$ schemes provide generic algorithms, that are still pretty competitive compared to subgradient and interior point methods. For general losses, these methods currently require to solve iteratively $\ell_2$-regularized problems and it would be desirable to relax this constraint.

Of course, many learning problems are by essence nonconvex and several approaches to inducing (sometimes more aggressively) sparsity are also nonconvex. Beyond providing an overview of these methods to the reader as a complement to the convex formulations, we have tried to suggest that faced with nonconvex nondifferentiable and therefore potentially quite hard problems to solve, a good strategy is to try and reduce the problem to solving iteratively convex problems, since more stable algorithms are available and progress can be monitored with duality gaps.

Last but not least, duality suggests strongly that multiple kernel learning is in a sense the dual view to sparsity, and provides a natural way, via the "kernel trick", to extend sparsity to reproducing kernel Hilbert spaces. We have therefore illustrated throughout the text that rather than being a vague connection, this duality can be exploited both conceptually, leading to the idea of structured MKL, and algorithmically to kernelize all of the algorithms we considered so as to apply them in the MKL and RKHS settings.

# Acknowledgments

Francis Bach, Rodolphe Jenatton and Guillaume Obozinski are supported in part by ANR under grant MGA ANR-07-BLAN-0311 and the European Research Council (SIERRA Project). Julien Mairal is supported by the NSF grant SES-0835531 and NSF award CCF-0939370. All the authors would like to thank the anonymous reviewers, whose comments have greatly contributed to improve the quality of this monograph.

# References


[1] J. Abernethy, F. Bach, T. Evgeniou, and J.-P. Vert, "A new approach to collaborative filtering: Operator estimation with spectral regularization," *Journal of Machine Learning Research*, vol. 10, pp. 803–826, 2009.

[2] J. Aflalo, A. Ben-Tal, C. Bhattacharyya, J. S. Nath, and S. Raman, "Variable sparsity kernel learning," *Journal of Machine Learning Research*, vol. 12, pp. 565–592, 2011.

[3] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.

[4] Y. Amit, M. Fink, N. Srebro, and S. Ullman, "Uncovering shared structures in multiclass classification," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2007.

[5] C. Archambeau and F. Bach, "Sparse probabilistic projections," in *Advances in Neural Information Processing Systems (NIPS)*, 2008.

[6] A. Argyriou, T. Evgeniou, and M. Pontil, "Convex multi-task feature learning," *Machine Learning*, vol. 73, no. 3, pp. 243–272, 2008.

[7] F. Bach, "Consistency of the group lasso and multiple kernel learning," *Journal of Machine Learning Research*, vol. 9, pp. 1179–1225, 2008.

[8] F. Bach, "Consistency of trace norm minimization," *Journal of Machine Learning Research*, vol. 9, pp. 1019–1048, 2008.

[9] F. Bach, "Exploring large feature spaces with hierarchical multiple kernel learning," in *Advances in Neural Information Processing Systems (NIPS)*, 2008.

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

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

[12] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski, "Convex optimization with sparsity-inducing norms," in *Optimization for Machine Learning*, (S. Sra, S. Nowozin, and S. J. Wright, eds.), MIT Press, 2011.

[13] F. Bach, G. R. G. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality, and the SMO algorithm," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2004.

[14] F. Bach, J. Mairal, and J. Ponce, "Convex sparse matrix factorizations," *Preprint arXiv:0812.1869v1*, 2008.

[15] R. G. Baraniuk, V. Cevher, M. Duarte, and C. Hegde, "Model-based compressive sensing," *IEEE Transactions on Information Theory*, vol. 56, no. 4, pp. 1982–2001, 2010.

[16] F. L. Bauer, J. Stoer, and C. Witzgall, "Absolute and monotonic norms," *Numerische Mathematik*, vol. 3, no. 1, pp. 257–264, 1961.

[17] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[18] D. P. Bertsekas, *Nonlinear Programming*. Athena Scientific, 2nd ed., 1999.

[19] P. Bickel, Y. Ritov, and A. Tsybakov, "Simultaneous analysis of Lasso and Dantzig selector," *Annals of Statistics*, vol. 37, no. 4, pp. 1705–1732, 2009.

[20] J. M. Borwein and A. S. Lewis, *Convex Analysis and Nonlinear Optimization: Theory and Examples*. Springer-Verlag, 2006.

[21] L. Bottou, "Online algorithms and stochastic approximations," *Online Learning and Neural Networks*, vol. 5, 1998.

[22] L. Bottou and O. Bousquet, "The tradeoffs of large scale learning," in *Advances in Neural Information Processing Systems (NIPS)*, 2008.

[23] L. Bottou and Y. LeCun, "Large scale online learning," in *Advances in Neural Information Processing Systems (NIPS)*, 2004.

[24] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1–124, 2011.

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

[26] D. M. Bradley and J. A. Bagnell, "Convex coding," in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI)*, 2009.

[27] P. Brucker, "An $O(n)$ algorithm for quadratic knapsack problems," *Operations Research Letters*, vol. 3, no. 3, pp. 163–166, 1984.

[28] C. Burges, "Dimension reduction: A guided tour," *Machine Learning*, vol. 2, no. 4, pp. 275–365, 2009.

[29] J. F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, pp. 1956–1982, 2010.

[30] E. J. Candès, M. Wakin, and S. Boyd, "Enhancing sparsity by reweighted L1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 877–905, 2008.

[31] F. Caron and A. Doucet, "Sparse Bayesian nonparametric regression," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2008.

[32] V. Cehver, M. Duarte, C. Hedge, and R. G. Baraniuk, "Sparse signal recovery using markov random fields," in *Advances in Neural Information Processing Systems (NIPS)*, 2008.

[33] A. Chambolle, "Total variation minimization and a class of binary MRF models," in *Proceedings of the fifth International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, 2005.

[34] A. Chambolle and J. Darbon, "On total variation minimization and surface evolution using parametric maximum flows," *International Journal of Computer Vision*, vol. 84, no. 3, pp. 288–307, 2009.

[35] V. Chandrasekaran, B. Recht, P. A. Parrilo, and A. S. Willsky, "The convex geometry of linear inverse problems," *Preprint arXiv:1012.0621*, 2010.

[36] G. H. G. Chen and R. T. Rockafellar, "Convergence rates in forward-backward splitting," *SIAM Journal on Optimization*, vol. 7, no. 2, pp. 421–444, 1997.

[37] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 20, pp. 33–61, 1999.

[38] P. L. Combettes and J.-C. Pesquet, *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*. Chapter Proximal Splitting Methods in Signal Processing. Springer-Verlag, 2011.

[39] P. L. Combettes and V. R. Wajs, "Signal recovery by proximal forward-backward splitting," *SIAM Multiscale Modeling and Simulation*, vol. 4, no. 4, pp. 1168–1200, 2006.

[40] S. F. Cotter, J. Adler, B. Rao, and K. Kreutz-Delgado, "Forward sequential algorithms for best basis selection," in *IEEE Proceedings of Vision Image and Signal Processing*, pp. 235–244, 1999.

[41] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk, "Iteratively reweighted least squares minimization for sparse recovery," *Communications on Pure and Applied Mathematics*, vol. 63, no. 1, pp. 1–38, 2010.

[42] D. L. Donoho and I. M. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage," *Journal of the American Statistical Association*, vol. 90, no. 432, pp. 1200–1224, 1995.

[43] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2011.

[44] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Annals of Statistics*, vol. 32, no. 2, pp. 407–499, 2004.

[45] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3736–3745, 2006.

[46] K. Engan, S. O. Aase, and H. Husoy et al., "Method of optimal directions for frame design," in *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1999.

[47] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its oracle properties," *Journal of the American Statistical Association*, vol. 96, no. 456, pp. 1348–1360, 2001.

[48] M. Fazel, H. Hindi, and S. Boyd, "A rank minimization heuristic with application to minimum order system approximation," in *Proceedings of the American Control Conference*, vol. 6, pp. 4734–4739, 2001.

[49] J. Friedman, T. Hastie, and R. Tibshirani, "A note on the group lasso and a sparse group lasso," *Preprint arXiv:1001:0736v1*, 2010.

[50] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[51] W. J. Fu, "Penalized regressions: The bridge versus the lasso," *Journal of Computational and Graphical Statistics*, vol. 7, no. 3, pp. 397–416, 1998.

[52] G. Gasso, A. Rakotomamonjy, and S. Canu, "Recovering sparse signals with non-convex penalties and DC programming," *IEEE Transactions on Signal Processing*, vol. 57, no. 12, pp. 4686–4698, 2009.

[53] A. Genkin, D. D. Lewis, and D. Madigan, "Large-scale bayesian logistic regression for text categorization," *Technometrics*, vol. 49, no. 3, pp. 291–304, 2007.

[54] R. Glowinski and P. Le Tallec, *Augmented Lagrangian and Operator-splitting Methods in Nonlinear Mechanics*. Society for Industrial Mathematics, 1989.

[55] Y. Grandvalet and S. Canu, "Outcomes of the equivalence of adaptive ridge with least absolute shrinkage," in *Advances in Neural Information Processing Systems (NIPS)*, 1999.

[56] R. Gribonval, V. Cevher, and M. E. Davies, "Compressible distributions for high-dimensional statistics," *preprint arXiv:1102.1249v2*, 2011.

[57] Z. Harchaoui, "Méthodes à Noyaux pour la Détection," PhD thesis, Télécom ParisTech, 2008.

[58] Z. Harchaoui and C. Lévy-Leduc, "Catching change-points with Lasso," in *Advances in Neural Information Processing Systems (NIPS)*, 2008.

[59] K. K. Herrity, A. C. Gilbert, and J. A. Tropp, "Sparse approximation via iterative thresholding," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing, (ICASSP)*, 2006.

[60] C. Hu, J. Kwok, and W. Pan, "Accelerated gradient methods for stochastic optimization and online learning," in *Advances in Neural Information Processing Systems (NIPS)*, 2009.

[61] J. Huang and T. Zhang, "The benefit of group sparsity," *Annals of Statistics*, vol. 38, no. 4, pp. 1978–2004, 2010.

[62] J. Huang, Z. Zhang, and D. Metaxas, "Learning with structured sparsity," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2009.

[63] H. Ishwaran and J. S. Rao, "Spike and slab variable selection: frequentist and Bayesian strategies," *Annals of Statistics*, vol. 33, no. 2, pp. 730–773, 2005.

[64] L. Jacob, G. Obozinski, and J.-P. Vert, "Group Lasso with overlaps and graph Lasso," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2009.

[65] R. Jenatton, "Structured sparsity-inducing norms: Statistical and algorithmic properties with applications to neuroimaging," PhD thesis, Ecole Normale Supérieure de Cachan, 2012.

[66] R. Jenatton, J.-Y. Audibert, and F. Bach, "Structured variable selection with sparsity-inducing norms," *Journal of Machine Learning Research*, vol. 12, pp. 2777–2824, 2011.

[67] R. Jenatton, A. Gramfort, V. Michel, G. Obozinski, F. Bach, and B. Thirion, "Multi-scale mining of fMRI data with hierarchical structured sparsity," in *International Workshop on Pattern Recognition in Neuroimaging (PRNI)*, 2011.

[68] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach, "Proximal methods for sparse hierarchical dictionary learning," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2010.

[69] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach, "Proximal methods for hierarchical sparse coding," *Journal of Machine Learning Research*, vol. 12, pp. 2297–2334, 2011.

[70] R. Jenatton, G. Obozinski, and F. Bach, "Structured sparse principal component analysis," in *Proceedings of International Workshop on Artificial Intelligence and Statistics (AISTATS)*, 2010.

[71] S. C. Johnson, "Hierarchical clustering schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, 1967.

[72] K. Kavukcuoglu, M. A. Ranzato, R. Fergus, and Y. LeCun, "Learning invariant features through topographic filter maps," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[73] S. Kim and E. P. Xing, "Tree-guided group lasso for multi-task regression with structured sparsity," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2010.

[74] G. S. Kimeldorf and G. Wahba, "Some results on Tchebycheffian spline functions," *Journal of Mathematical Analysis and Applications*, vol. 33, pp. 82–95, 1971.

[75] K. Koh, S. J. Kim, and S. Boyd, "An interior-point method for large-scale l1-regularized logistic regression," *Journal of Machine Learning Research*, vol. 8, no. 8, pp. 1519–1555, 2007.

[76] B. Krishnapuram, L. Carin, M. A. T. Figueiredo, and A. J. Hartemink, "Sparse multinomial logistic regression: Fast algorithms and generalization bounds," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 957–968, 2005.

[77] G. R. G. Lanckriet, N. Cristianini, L. El Ghaoui, P. Bartlett, and M. I. Jordan, "Learning the kernel matrix with semidefinite programming," *Journal of Machine Learning Research*, vol. 5, pp. 27–72, 2004.

[78] G. R. G. Lanckriet, T. De Bie, N. Cristianini, M. I. Jordan, and W. S. Noble, "A statistical framework for genomic data fusion," *Bioinformatics*, vol. 20, pp. 2626–2635, 2004.

[79] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Advances in Neural Information Processing Systems (NIPS)*, 2007.

[80] A. Lefèvre, F. Bach, and C. Févotte, "Itakura-Saito nonnegative matrix factorization with group sparsity," in *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011.

[81] P. L. Lions and B. Mercier, "Splitting algorithms for the sum of two nonlinear operators," *SIAM Journal on Numerical Analysis*, vol. 16, no. 6, pp. 964–979, 1979.

[82] H. Liu, M. Palatucci, and J. Zhang, "Blockwise coordinate descent procedures for the multi-task lasso, with applications to neural semantic basis discovery," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2009.

[83] K. Lounici, M. Pontil, A. B. Tsybakov, and S. van de Geer, "Taking advantage of sparsity in multi-task learning," *Preprint arXiv:0903.1468*, 2009.

[84] N. Maculan and G. Galdino de Paula, "A linear-time median-finding algorithm for projecting a vector on the simplex of $\mathbb{R}^n$," *Operations Research Letters*, vol. 8, no. 4, pp. 219–222, 1989.

[85] J. Mairal, "Sparse coding for machine learning, image processing and computer vision," PhD thesis, Ecole Normale Supérieure de Cachan, `http://tel.archives-ouvertes.fr/tel-00595312`, 2010.

[86] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *Journal of Machine Learning Research*, vol. 11, pp. 19–60, 2010.

[87] J. Mairal, R. Jenatton, G. Obozinski, and F. Bach, "Network flow algorithms for structured sparsity," in *Advances in Neural Information Processing Systems (NIPS)*, 2010.

[88] J. Mairal, R. Jenatton, G. Obozinski, and F. Bach, "Convex and network flow optimization for structured sparsity," *Journal of Machine Learning Research*, vol. 12, pp. 2681–2720, 2011.

[89] S. Mallat and Z. Zhang, "Matching pursuit in a time-frequency dictionary," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.

[90] H. Markowitz, "Portfolio selection," *Journal of Finance*, vol. 7, no. 1, pp. 77–91, 1952.

[91] B. Martinet, "Régularisation d'inéquations variationnelles par approximations successives," *Revue franaise d'informatique et de recherche opérationnelle, série rouge*, 1970.

[92] A. F. T. Martins, N. A. Smith, P. M. Q. Aguiar, and M. A. T. Figueiredo, "Structured sparsity in structured prediction," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2011.

[93] C. A. Micchelli, J. M. Morales, and M. Pontil, "Regularizers for structured sparsity," *Preprint arXiv:1010.0556v2*, 2011.

[94] J. J. Moreau, "Fonctions convexes duales et points proximaux dans un espace hilbertien," *Comptes-Rendus de l'Académie des Sciences de Paris, Série A, Mathématiques*, vol. 255, pp. 2897–2899, 1962.

[95] S. Mosci, L. Rosasco, M. Santoro, A. Verri, and S. Villa, "Solving structured sparsity regularization with proximal methods," *Machine Learning and Knowledge Discovery in Databases*, pp. 418–433, 2010.

[96] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM Journal on Computing*, vol. 24, pp. 227–234, 1995.

[97] R. M. Neal, *Bayesian Learning for Neural Networks*, vol. 118. Springer Verlag, 1996.

[98] D. Needell and J. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.

[99] S. Negahban, P. Ravikumar, M. J. Wainwright, and B. Yu, "A unified framework for high-dimensional analysis of M-estimators with decomposable regularizers," in *Advances in Neural Information Processing Systems (NIPS)*, 2009.

[100] Y. Nesterov, *Introductory Lectures on Convex Optimization: A Basic Course*. Kluwer Academic Publishers, 2004.

[101] Y. Nesterov, "Smooth minimization of non-smooth functions," *Mathematical Programming*, vol. 103, no. 1, pp. 127–152, 2005.

[102] Y. Nesterov, "Gradient methods for minimizing composite objective function," Technical Report, Center for Operations Research and Econometrics (CORE), Catholic University of Louvain, Discussion paper, 2007.

[103] Y. Nesterov, "Efficiency of coordinate descent methods on huge-scale optimization problems," Technical Report, Center for Operations Research and Econometrics (CORE), Catholic University of Louvain, Discussion paper, 2010.

[104] J. Nocedal and S. J. Wright, *Numerical Optimization*. Springer Verlag, 2nd ed., 2006.

[105] G. Obozinski, L. Jacob, and J.-P. Vert, "Group Lasso with overlaps: the Latent Group Lasso approach," *preprint HAL — inria-00628498*, 2011.

[106] G. Obozinski, B. Taskar, and M. I. Jordan, "Joint covariate selection and joint subspace selection for multiple classification problems," *Statistics and Computing*, vol. 20, no. 2, pp. 231–252, 2009.

[107] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, pp. 607–609, 1996.

[108] M. R. Osborne, B. Presnell, and B. A. Turlach, "On the Lasso and its dual," *Journal of Computational and Graphical Statistics*, vol. 9, no. 2, pp. 319–337, 2000.

[109] M. Pontil, A. Argyriou, and T. Evgeniou, "Multi-task feature learning," in *Advances in Neural Information Processing Systems (NIPS)*, 2007.

[110] A. Rakotomamonjy, F. Bach, S. Canu, and Y. Grandvalet, "SimpleMKL," *Journal of Machine Learning Research*, vol. 9, pp. 2491–2521, 2008.

[111] N. S. Rao, R. D. Nowak, S. J. Wright, and N. G. Kingsbury, "Convex approaches to model wavelet sparsity patterns," in *International Conference on Image Processing (ICIP)*, 2011.

[112] F. Rapaport, E. Barillot, and J.-P. Vert, "Classification of arrayCGH data using fused SVM," *Bioinformatics*, vol. 24, no. 13, pp. 375–382, 2008.

[113] P. Ravikumar, J. Lafferty, H. Liu, and L. Wasserman, "Sparse additive models," *Journal of the Royal Statistical Society. Series B, statistical methodology*, vol. 71, pp. 1009–1030, 2009.

[114] K. Ritter, "Ein verfahren zur lösung parameterabhängiger, nichtlinearer maximum-probleme," *Mathematical Methods of Operations Research*, vol. 6, no. 4, pp. 149–166, 1962.

[115] R. T. Rockafellar, *Convex Analysis*. Princeton University Press, 1997.

[116] V. Roth and B. Fischer, "The Group-Lasso for generalized linear models: uniqueness of solutions and efficient algorithms," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2008.

[117] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.

[118] M. Schmidt, G. Fung, and R. Rosales, "Fast optimization methods for L1 regularization: A comparative study and two new approaches," in *Proceedings of the European Conference on Machine Learning (ECML)*, 2007.

[119] M. Schmidt, D. Kim, and S. Sra, "Projected Newton-type methods in machine learning," in *Optimization for Machine Learning*, (S. Sra, S. Nowozin, and S. J. Wright, eds.), MIT Press, 2011.

[120] M. Schmidt, N. Le Roux, and F. Bach, "Convergence rates of inexact proximal-gradient methods for convex optimization," in *Advances in Neural Information Processing Systems (NIPS)*, 2011.

[121] M. Schmidt and K. Murphy, "Convex structure learning in log-linear models: Beyond pairwise potentials," in *Proceedings of International Workshop on Artificial Intelligence and Statistics (AISTATS)*, 2010.

[122] B. Schölkopf and A. J. Smola, *Learning with Kernels*. MIT Press, 2001.

[123] M. W. Seeger, "Bayesian inference and optimal design for the sparse linear model," *Journal of Machine Learning Research*, vol. 9, pp. 759–813, 2008.

[124] S. Shalev-Shwartz and A. Tewari, "Stochastic methods for $\ell_1$-regularized loss minimization," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2009.

[125] A. Shapiro, D. Dentcheva, and A. P. Ruszczyński, *Lectures on Stochastic Programming: Modeling and Theory*. Society for Industrial Mathematics, 2009.

[126] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[127] S. K. Shevade and S. S. Keerthi, "A simple and efficient algorithm for gene selection using sparse logistic regression," *Bioinformatics*, vol. 19, no. 17, pp. 2246–2253, 2003. Oxford Univ Press.

[128] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf, "Large scale multiple kernel learning," *Journal of Machine Learning Research*, vol. 7, pp. 1531–1565, 2006.

[129] P. Sprechmann, I. Ramirez, G. Sapiro, and Y. C. Eldar, "C-HiLasso: A collaborative hierarchical sparse modeling framework," *IEEE Transactions on Signal Processing*, vol. 59, no. 9, pp. 4183–4198, 2011.

[130] N. Srebro, J. D. M. Rennie, and T. S. Jaakkola, "Maximum-margin matrix factorization," in *Advances in Neural Information Processing Systems (NIPS)*, 2005.

[131] T. Suzuki and R. Tomioka, "SpicyMKL: A fast algorithm for multiple kernel learning with thousands of kernels," *Machine Learning*, vol. 85, pp. 77–108, 2011.

[132] M. Szafranski, Y. Grandvalet, and P. Morizet-Mahoudeaux, "Hierarchical penalization," in *Advances in Neural Information Processing Systems (NIPS)*, 2007.

[133] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *Journal of the Royal Statistical Society Series B*, vol. 58, no. 1, pp. 267–288, 1996.

[134] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused Lasso," *Journal of the Royal Statistical Society Series B*, vol. 67, no. 1, pp. 91–108, 2005.

[135] R. Tomioka, T. Suzuki, and M. Sugiyama, "Augmented Lagrangian methods for learning, selecting and combining features," in *Optimization for Machine Learning*, (S. Sra, S. Nowozin, and S. J. Wright, eds.), MIT Press, 2011.

[136] J. A. Tropp, "Greed is good: Algorithmic results for sparse approximation," *IEEE Transactions on Signal Processing*, vol. 50, no. 10, pp. 2231–2242, 2004.

[137] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Algorithms for simultaneous sparse approximation. Part I: Greedy pursuit," *Signal Processing, Special Issue "Sparse Approximations in Signal and Image Processing"*, vol. 86, pp. 572–588, 2006.

[138] P. Tseng, "Applications of a splitting algorithm to decomposition in convex programming and variational inequalities," *SIAM Journal on Control and Optimization*, vol. 29, pp. 119–138, 1991.

[139] P. Tseng, "On accelerated proximal gradient methods for convex-concave optimization," *submitted to SIAM Journal on Optimization*, 2008.

[140] P. Tseng and S. Yun, "A coordinate gradient descent method for nonsmooth separable minimization," *Mathematical Programming*, vol. 117, no. 1, pp. 387–423, 2009.

[141] B. A. Turlach, W. N. Venables, and S. J. Wright, "Simultaneous variable selection," *Technometrics*, vol. 47, no. 3, pp. 349–363, 2005.

[142] E. Van den Berg, M. Schmidt, M. P. Friedlander, and K. Murphy, "Group sparsity via linear-time projections," Technical report, University of British Columbia, Technical Report number TR-2008-09, 2008.

[143] G. Varoquaux, R. Jenatton, A. Gramfort, G. Obozinski, B. Thirion, and F. Bach, "Sparse structured dictionary learning for brain resting-state activity modeling," in *NIPS Workshop on Practical Applications of Sparse Modeling: Open Issues and New Directions*, 2010.

[144] J.-P. Vert and K. Bleakley, "Fast detection of multiple change-points shared by many signals using group LARS," in *Advances in Neural Information Processing Systems (NIPS)*, 2010.

[145] M. J. Wainwright, "Sharp thresholds for noisy and high-dimensional recovery of sparsity using $\ell_1$-constrained quadratic programming," *IEEE Transactions on Information Theory*, vol. 55, no. 5, pp. 2183–2202, 2009.

[146] S. Weisberg, *Applied Linear Regression*. Wiley, 1980.

[147] D. P. Wipf and S. Nagarajan, "A new view of automatic relevance determination," *Advances in Neural Information Processing Systems (NIPS)*, 2008.

[148] D. P. Wipf and B. D. Rao, "Sparse bayesian learning for basis selection," vol. 52, no. 8, pp. 2153–2164, 2004.

[149] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

[150] S. J. Wright, "Accelerated block-coordinate relaxation for regularized optimization," Technical report, Technical report, University of Wisconsin-Madison, 2010.

[151] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo, "Sparse reconstruction by separable approximation," *IEEE Transactions on Signal Processing*, vol. 57, no. 7, pp. 2479–2493, 2009.

[152] T. T. Wu and K. Lange, "Coordinate descent algorithms for lasso penalized regression," *Annals of Applied Statistics*, vol. 2, no. 1, pp. 224–244, 2008.

[153] L. Xiao, "Dual averaging methods for regularized stochastic learning and online optimization," *Journal of Machine Learning Research*, vol. 9, pp. 2543–2596, 2010.

[154] H. Xu, C. Caramanis, and S. Sanghavi, "Robust PCA via outlier pursuit," *Preprint arXiv:1010.4237*, 2010.

[155] G. X. Yuan, K. W. Chang, C. J. Hsieh, and C. J. Lin, "A comparison of optimization methods for large-scale l1-regularized linear classification," Technical Report, Department of Computer Science, National University of Taiwan, 2010.

[156] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society Series B*, vol. 68, pp. 49–67, 2006.

[157] P. Zhao, G. Rocha, and B. Yu, "The composite absolute penalties family for grouped and hierarchical variable selection," *Annals of Statistics*, vol. 37, no. 6A, pp. 3468–3497, 2009.

[158] P. Zhao and B. Yu, "On model selection consistency of Lasso," *Journal of Machine Learning Research*, vol. 7, pp. 2541–2563, 2006.

[159] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society Series B*, vol. 67, no. 2, pp. 301–320, 2005.