Learning with Structured Sparsity: From Discrete to Convex and Back

THÈSE N^{O} 8516 (2018)

PRÉSENTÉE LE 22 JUIN 2018

À LA FACULTÉ DES SCIENCES ET TECHNIQUES DE L'INGÉNIEUR
LABORATOIRE DE SYSTÈMES D'INFORMATION ET D'INFÉRENCE
PROGRAMME DOCTORAL EN INFORMATIQUE ET COMMUNICATIONS

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Marwa EL HALABI

acceptée sur proposition du jury:

Prof. P. Vandergheynst, président du jury
Prof. V. Cevher, directeur de thèse
Prof. F. Bach, rapporteur
Prof. R. Baraniuk, rapporteur
Prof. M. Jaggi, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2018

To my sister, Mira

Acknowledgements

This thesis would not have been possible without the support of many people. First, I would like to express my gratitude to my advisor Volkan Cevher, for his guidance and support throughout my PhD. Thank you Volkan for introducing me to the interesting areas of machine learning and optimization, for your enthusiasm for research and your optimism, and for being always available for discussions about research, career advice, or just TV series and funny YouTube videos.

I was honored to have Francis Bach, Richard Baraniuk, Martin Jaggi as jury members of my thesis, and Pierre Vandergheynst as president of the jury. I am grateful for their time, their kindness, and for the insightful discussions during and outside the private defense.

I would also like to thank Francis for hosting me during the Fall of 2017 at INRIA, and for providing a welcoming and stimulating working environment. I truly enjoyed and learned a lot from our collaboration, which resulted in the third chapter of this thesis. Thanks for the bright and kind members of SIERRA and WILLOW teams for making my stay in Paris fun and memorable. Many thanks to Andreas Krause for hosting me for a month at ETH, and to Josip Djolonga for instructive research discussions, both during my stay at ETH and throughout our PhDs.

I was fortunate to be surrounded by very kind and smart colleagues during my PhD. I am grateful for all the current and past members of LIONS who made working in the lab much more enjoyable, and long nights before deadlines more bearable. Thanks Gosia for your invaluable help in all sorts of administrative and every-day details, and for the nice coffee chats; Ya-Ping for letting me badger you with research questions, for your help and advice, and for sharing your music playlists, your fancy coffee and whiskey (sorry for not appreciating it enough); Yu-Chun for being one of the most honest people I ever met; Alp and Ahmet[1] for all the fun outings and trips, and for checking how my thesis is going; Ilija for fun climbing sessions and movies/series recommendations; Luca for our collaboration in the beginning of my PhD, your support at the end of it, and for introducing me to Le Cube; and Tasos for your generous and helpful advice from the moment I applied to LIONS, until now when I am pondering my next step.

I am indebted to several teachers for paving my path to EPFL. I deeply thank AUB's professors, in particular Louay Bazzi for the best courses I took at AUB which cultivated my interest in theoretical computer science, for introducing us to EPFL, and encouraging us to pursue graduate studies, Fadi Zaraket for his support and encouragement, and my mathematics teacher in high-school, Mostafa Chall, who by being a dedicated, passionate, and caring teacher, reinforced my love for math.

[1] Sorry for trolling you before that deadline!

Acknowledgements

Many thanks for all my friends for the great times we shared and for keeping me sane (more or less) during this journey. I was very lucky to have several close friends from Lebanon move to Lausanne at the same time as me. A special thanks to my best friend Rafah, for all the fun, for letting me nag and always knowing how to cheer me up. Spending time with you is never dull! Thanks Ibrahim (otherwise known as Isha or Philippo), for moving at the right time to Lausanne to replace Rafah ☺, for interesting discussions, for making me appreciate Lebanese artists more, and of course for letting me nag. Having two close friends, Farah and Abbas, as my flatmates turned our apartment into a home. Thank you Farah for the bubbling energy you bring to my life, and Abbas for the sarcastic one. Thanks Ghid and Dan (my favorite couple) for fun brunches and dinners, fascinating philosophical discussions, for lending me books, and for letting me play with your adorable Dalia.

I am fortunate to have met several awesome people at Lausanne. Thanks Renata (whose cheerfulness is contagious), Ajay (who tried, and failed, to teach me to be chill), and Artem, for all the fun trips, hikes, ski weekends, and parties. Thanks Ersi (who is always generous with her compliments), Elena, Manos, Marco, Beril, Mireille, Eda, Andreas, Betül, and Timo for all the nice moments we shared. Thanks to the Lebanese gang at Lausanne for keeping me connected to home: Elie, Hiba, Raed, Ahmad, Serj, Sahar, Rajai, Dia, Elio, Amer, and Hani. Thanks also to the friends with whom I managed stay connected despite the distance: Maya, Sireen, Zahi, Dana, and my childhood friend Jihane.

I discovered climbing in Lausanne and became obsessed with it. Thanks to all my climbing partners and friends, and in particular Justin, Paola, and Aaron for the weekly climbing sessions at Le Cube and the following fun discussions around beer and Hummus; and to the Club Montagne at EPFL for organizing exciting outdoors outings.

Last but not least, I want to thank my family for their unconditional love and support. Thank you Mom and Jeddo for all your sacrifices for my education, my brother Mohamad for your encouragements; and my sister Mira, with whom I can be completely myself, for always being there for me (literally ☺).

Lausanne, 6 June 2018 M. E.

Abstract

In modern-data analysis applications, the abundance of data makes extracting meaningful information from it challenging, in terms of computation, storage, and interpretability. In this setting, exploiting *sparsity* in data has been essential to the development of scalable methods to problems in machine learning, statistics and signal processing. However, in various applications, the input variables exhibit structure beyond simple sparsity. This motivated the introduction of *structured sparsity* models, which capture such sophisticated structures, leading to significant performance gains and better interpretability. Structured sparsity approaches have been successfully applied in a variety of domains including computer vision, text and audio processing, medical imaging, and bioinformatics.

The goal of this thesis is to improve on these methods and expand their success to a wider range of applications. We thus develop novel methods to incorporate general structure a priori in learning problems, which balance computational and statistical efficiency trade-offs. To achieve this, our results bring together tools from discrete and convex optimization.

Applying structured sparsity approaches in general is challenging because structures encountered in practice are naturally combinatorial. An effective approach to circumvent this computational challenge is to employ continuous convex relaxations. We thus start by introducing a new class of structured sparsity models, able to capture a large range of structures, which admit tight convex relaxations amenable to efficient optimization. We then present an in-depth study of the geometric and statistical properties of convex relaxations of general combinatorial structures. In particular, we characterize which structure is lost by imposing convexity and which is preserved.

We then focus on the optimization of the convex composite problems that result from the convex relaxations of structured sparsity models. We develop efficient algorithmic tools to solve these problems in a non-Euclidean setting, leading to faster convergence in some cases.

Finally, to handle structures that do not admit meaningful convex relaxations, we propose to use, as a heuristic, a non-convex proximal gradient method, efficient for several classes of structured sparsity models. We further extend this method to address a probabilistic structured sparsity model, which we introduce to model approximately sparse signals.

Key Words: Structured sparsity, high-dimensional learning, convex relaxations, convex composite minimization, integer and linear programming, submodularity.

Résumé

Dans l'exercice qu'est analyse des données modernes, l'abondance et le volume de ces dernières rend l'extraction d'informations significatives difficile, tant sur le plan de calcul, du stockage ou encore de l'interprétabilité. Dans ce contexte, utiliser la *parcimonie* (*sparsity*) du problème a été essentielle au développement de méthodes supportant de grandes quantités de données, que ce soit pour des problèmes d'apprentissage automatique, de statistiques ou encore de traitement du signal. Cependant, dans diverses applications, les variables d'entrée présentent une structure au-delà de la simple parcimonie. Ceci a motivé l'introduction de modèles de *parcimonie structurée*, qui rendent compte de ces structures sophistiquées, conduisant à des gains de performance significatifs ainsi qu'une meilleure interprétation. Les approches de parcimonie structurée ont été appliquées avec succès dans divers domaines, dont la vision par ordinateur, le traitement de texte et audio ou encore l'imagerie médicale et la bio-informatique.

Le but de cette thèse est d'améliorer ces méthodes et d'étendre leur succès à un plus large éventail d'applications. Nous développons ainsi des méthodes d'apprentissage qui permettent d'exploiter ces structures en généralité, tout en équilibrant les différents compromis entre efficacité statistique et algorithmique. Pour ce faire, nos résultats rassemblent des outils issus de l'optimisation discrète et convexe.

En raison de la nature combinatoire des problèmes, l'application des approches de parcimonie structurée en général est difficile. Une approche efficace pour contourner cette difficulté consiste à utiliser des relaxations convexes continues. Nous commençons donc par introduire une nouvelle classe de modèles de parcimonie structurée, capables d'exprimer une large gamme de structures, et qui admettent des relaxations convexes n'induisant que peu de pertes et pouvant être optimisées efficacement. Nous présentons ensuite une étude approfondie des propriétés géométriques et statistiques des relaxations convexes de structures combinatoires générales. En particulier, nous donnons une caractérisation des structures qui sont perdues en imposant la convexité, et de celles qui sont préservées.

Nous nous concentrons ensuite sur l'optimisation des problèmes convexes qui résultent des relaxations convexes des modèles de parcimonie structurée. Nous développons des outils algorithmiques efficaces pour résoudre ces problèmes dans un contexte non-Euclidien, ce qui conduit dans certains cas à une convergence plus rapide de nos algorithmes.

Enfin, pour gérer des structures qui n'admettent pas de bonnes relaxations convexes, nous proposons d'utiliser, comme heuristique, une méthode de gradient proximal non-convexe, efficace pour plusieurs classes de modèles de parcimonie structurée. Nous étendons davantage cette méthode pour traiter un modèle probabiliste de parcimonie structurée, que nous introduisons pour

modéliser des signaux approximativement parcimonieux.

Key Words : Parcimonie structurée, apprentissage en haute dimension, relaxations convexes, minimisation convexe composée, programmation en nombres entiers et linéaire, sous-modularité.

Contents

1 Introduction

Learning problems are ubiquitous in machine learning, signal processing and statistics applications, where given some data, we are interested in learning the underlying parameter vector. Depending on the application, the objective can be to estimate the parameter vector, or to use it for prediction or classification. In the presence of large and complicated data, solving such tasks becomes challenging, without a priori model on the data source.

Such models are particularly important in the high-dimensional setting, where the number of variables exceeds the number of observations. This setting naturally arises in modern data analysis problems, where the current trend of systematic data collection leads to a large ambient dimension. Moreover, many applications are intrinsically high-dimensional, due to observations being expensive (cost or time-wise). Without further assumptions, the learning problem in this setting is ill-posed (it admits infinitely many solutions). Fortunately, the relevant information of real-world data typically lies in a low-dimensional space. For example, in machine learning and statistics, only a small number of features are usually relevant. Similarly, in signal processing, signals can often be approximated by a small collection of basis or dictionary vectors. This idea that only few elements out of many are important is known as *sparsity*, and has been key to the development of scalable methods that circumvent the curse of dimensionality.

While sparse modeling is powerful, it does not account for potential relationships that may exist between the variables. Indeed in many applications, the data source naturally exhibit additional structure beyond sparsity. For example, in computer vision, the pixels corresponding to the foreground of an image are expected to be *clustered* together (see Figure 1.1), and the coefficients of the wavelet transform of an image are naturally organized on a *tree* (see Figure 1.2); in genomics, gene expression patterns are better explained by *groups* of genes sharing a common biological function [STM+05]. Moreover, it is sometimes advantageous to enforce additional structure. For example, in deep learning, a neural network with a compact structure, where only a few *groups* of weights in adjacent memory space are active, is desirable to reduce computation, especially in resource constrained devices [WWW+16]. *Structured sparsity* models capture such sophisticated structures. Incorporating such models into the learning process leads to

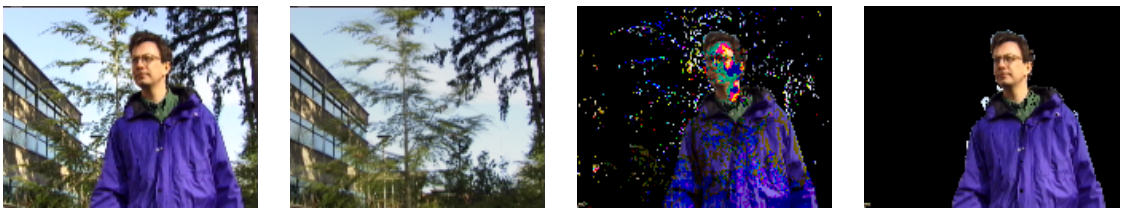


Figure 1.1: Background subtraction task: Given a sequence of frames, the goal is to segment out foreground objects in a new image. From left to right: original image; estimated background; foreground estimated with sparsity model; foreground estimated with a structured sparsity model (clustered support). This figure is taken from [MJBO10].

Figure 1.2: Image inpainting task: The goal is to reconstruct the missing pixels of an image. From left to right: original image (256×256); its wavelet transform; image with 50% missing pixels; image estimated with sparsity model; image estimated with a structured sparsity model (tree support).

significant improvements in the estimation performance, as illustrated for example in Figures 1.1 and 1.2. It also leads to better noise robustness, better interpretability and allows recovery with fewer observations [EM09, BD09, CHDB09, BCDH10, HZ10, JOB10, RRN12]. To highlight the importance of the last two properties, we note that for example, obtaining more interpretable results in gene analysis, by focusing on groups of genes instead of single genes, allows biologists to identify relevant biological pathways in cancer-related data sets [STM$^+$05]. Also, in the case of Magnetic Resonance Imaging (MRI), reducing the number of measurements allows the procedure to be shorter and thus less uncomfortable for patients [LDP07]. Moreover, sometimes observations are simply not available, like in the image inpainting task (see Figure 1.2).

The main goal of this thesis is to improve on existing structured sparsity methods and expand their success to a wider range of applications. In particular, we are interested in developing effective methods to exploit available a priori knowledge, which address the following three concerns.

- Computational efficiency: How to efficiently solve the underlying optimization problem?
- Statistical efficiency: How to reduce the number of samples needed for accurate solutions?
- Generality: How to handle a wide range of structures?

In what follows, we will first present the notation used throughout the thesis in Section 1.1, before introducing more formally the problem set-up of structured sparse learning in Section 1.2. We follow up with an overview of related work, where we identify some gaps and open problems in

the state-of-the-art for structured sparsity, and point out how the results presented in this thesis fill some of these gaps. In particular, we review in Section 1.3 convex approaches to structured sparsity, and in Section 1.4 two convex optimization methods for solving the resulting convex problems. In Section 1.5, we briefly review non-convex approaches to structured sparsity. We conclude the introduction with an overview of the main contributions made by this thesis.

Some parts of the related work sections are based on the book chapter [KBEH$^+$15], coauthored with Anastasios Kyrillidis, Luca Baldassarre, Quoc Tran-Dinh, and Volkan Cevher.

1.1 Notation, terminology and prerequisites

We introduce here the notation we will use throughout the thesis, and some basic terminology.

We denote scalars by lowercase letters, vectors by lowercase boldface letters, matrices by boldface uppercase letters, and sets by uppercase letters.

Real-valued functions: The set of real numbers is denoted by $\mathbb{R}$ and the set of non-negative real numbers by $\mathbb{R}_+$. We write $\overline{\mathbb{R}}$ for $\mathbb{R} \cup \{+\infty\}$. Given an extended real-valued function $f : \mathbb{R}^d \to \overline{\mathbb{R}}$, we denote its domain by $\mathrm{dom}(f) := \{\boldsymbol{x} \in \mathbb{R}^d : f(\boldsymbol{x}) < +\infty\}$, its epigraph by $\mathrm{epi}(f) := \{(\boldsymbol{x}, t) \in \mathbb{R}^d \times \mathbb{R} : f(\boldsymbol{x}) \leq t\}$. We say f is *proper* if its domain is non-empty, and *lower semi-continuous*, or *closed*, if its epigraph is a closed set. We say f is *positively homogeneous* if $f(\alpha\boldsymbol{x}) = \alpha f(\boldsymbol{x}), \forall \boldsymbol{x} \in \mathbb{R}^d, \forall \alpha > 0$. We denote the Fenchel conjugate of f by f^*, defined as $f^*(\boldsymbol{x}) = \sup_{\boldsymbol{y} \in \mathbb{R}^d} \boldsymbol{x}^\top \boldsymbol{y} - f(\boldsymbol{y})$. If f is differentiable, we denote its gradient by ∇f, and if it is non-differentiable we denote by ∂f its subdifferential set. Given a set $C \subseteq \mathbb{R}^d$, we will denote by $\iota_C(\boldsymbol{x})$ the indicator function of the set C, taking value 0 on the set C and $+\infty$ outside it.

Convex sets and functions: A subset $C \subseteq \mathbb{R}^d$ is *convex*, if for any two choices $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, the line segment that connects $\boldsymbol{x}$ and $\boldsymbol{y}$ also belongs to C, i.e., $\forall \lambda \in [0, 1], \lambda\boldsymbol{x} + (1 - \lambda)\boldsymbol{y} \in C$. A function $f : \mathbb{R}^d \to \overline{\mathbb{R}}$ is convex, if its domain is convex, and $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathrm{dom}(f), \forall \lambda \in [0, 1], f(\lambda\boldsymbol{x} + (1 - \lambda)\boldsymbol{y}) \leq \lambda f(\boldsymbol{x}) + (1 - \lambda) f(\boldsymbol{y})$. If f is convex, then $-f$ is called *concave*.

Set-valued functions: We consider the ground set $V = \{1, \ldots, d\}$, and its power set $2^V = \{S | S \subseteq V\}$ composed of the 2^d subsets of V. Given a set $S \subseteq V$, the notation S^c denotes the set complement of S with respect to V, and $|S|$ its cardinality. Given a set function $F : 2^V \to \overline{\mathbb{R}}$, we say F is proper if its domain $\mathcal{D} := \{S : F(S) < +\infty\} \neq \emptyset$ is non-empty, and *monotone* if $\forall A \subseteq B \subseteq V, F(A) \leq F(B)$.

Submodular set functions: A finite-valued set function $F : 2^V \to \mathbb{R}$ is *submodular* if and only if $\forall A \subseteq B \subseteq V, \forall i \in B^c, F(B \cup \{i\}) - F(B) \leqslant F(A \cup \{i\}) - F(A)$. If F is submodular, then $-F$ is called *supermodular*. If F is both submodular and supermodular, it is called *modular*.

Vector notation: The i-th entry of a vector $\boldsymbol{x}$ is denoted as x_i. In iterative algorithms, we use

superscripts $\boldsymbol{x}^k$ to denote the k-th vector in a sequence of vectors $\boldsymbol{x}^1, \boldsymbol{x}^2, \cdots, \boldsymbol{x}^k$. Given $\boldsymbol{x} \in \mathbb{R}^d$ and a set $S \subseteq V$, $\boldsymbol{x}_S$ denotes the vector in $\mathbb{R}^d$ s.t., $[\boldsymbol{x}_S]_i = x_i, \forall i \in S$ and $[\boldsymbol{x}_S]_i = 0, \forall i \notin S$. $\boldsymbol{Q}_{SS}$ is defined similarly for a matrix $\boldsymbol{Q} \in \mathbb{R}^{d \times d}$. We let $\mathbb{1}_d, \mathbf{0}_d$ be the vectors in R^d of all ones and all zeros, respectively, and $\boldsymbol{I}_d$ the $d \times d$ identity matrix. We drop subscripts whenever the dimensions are clear from the context. Accordingly, we let $\mathbb{1}_S$ be the indicator vector of the set S. We drop the subscript for $S = V$, so that $\mathbb{1}_V = \mathbb{1}$ denotes the vector of all ones.

We call the set of non-zero elements of a vector $\boldsymbol{x}$ the support, denoted by $\operatorname{supp}(\boldsymbol{x}) = \{i : x_i \neq 0\}$. We use the notation from submodular analysis, where a vector $\boldsymbol{x} \in \mathbb{R}^d$ also denotes the modular set-function defined as $\boldsymbol{x}(S) = \sum_{i \in S} x_i$. The symbol $\circ$ denotes the coordinate-wise multiplication, i.e., $[\boldsymbol{x} \circ \boldsymbol{y}]_i = x_i y_i$. Similarly the operations $|\boldsymbol{x}|$, $\boldsymbol{x} \geq \boldsymbol{y}$ and $\operatorname{sign}(\boldsymbol{x})$ are applied element-wise, i.e., $[|\boldsymbol{x}|]_i = |x_i|$, $\boldsymbol{x} \geq \boldsymbol{y}$ iff $x_i \geq y_i, \forall i \in V$ and $[\operatorname{sign}(\boldsymbol{x})]_i = \pm 1$ is the sign of x_i with $\operatorname{sign}(0) = 0$. The vector containing the positive part of $\boldsymbol{x}$ is denoted by $\boldsymbol{x}_+ = \max\{\boldsymbol{x}, \mathbf{0}\}$ (maximum taken element wise).

Inner product and norms: The inner product between two vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$ is denoted by $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \boldsymbol{x}^\top \boldsymbol{y} = \sum_{i=1}^d x_i y_i$. A norm of a vector $\boldsymbol{x}$ is denoted by $\|\boldsymbol{x}\|$ and its dual by $\|\boldsymbol{x}\|_* := \max_{\|\boldsymbol{y}\| \leq 1} \boldsymbol{y}^\top \boldsymbol{x}$. For $p > 0$, the ℓ_p-quasi-norm is given by $\|\boldsymbol{x}\|_p = (\sum_{i=1}^d |x_i|^p)^{1/p}$. $\|\cdot\|_p$ becomes a norm, if $p \geq 1$, and $\|\boldsymbol{x}\|_\infty = \max_i |x_i|$. The ℓ_0-pseudo-norm is defined as: $\|\boldsymbol{x}\|_0 := |\operatorname{supp}(x)|$. For $p \in [1, \infty]$, we define the conjugate $q \in [1, \infty]$ via $\frac{1}{p} + \frac{1}{q} = 1$.

Prerequisites: Throughout the thesis, we make extensive use of concepts from submodular analysis. We review the relevant notions in Appendix A.

1.2 Learning with structured sparsity

We present in this section the formal set-up of the learning problems we consider in this thesis.

1.2.1 Problem set-up

In a structured sparsity learning problem, we are interested in learning a parameter vector $\boldsymbol{x}^\natural \in \mathbb{R}^d$ from some noisy observations $\boldsymbol{y} \in \mathbb{R}^n$ that depend on $\boldsymbol{x}^\natural$, where $\boldsymbol{x}^\natural$ is assumed to satisfy some structure, e.g, sparsity. A vector $\boldsymbol{x} \in \mathbb{R}^d$ is said to be s-sparse, if it has only $s < d$ non-zero coefficients. In the commonly used linear model, $\boldsymbol{y}$ and $\boldsymbol{x}^\natural$ are related by $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}^\natural + \boldsymbol{\varepsilon}$, where $\boldsymbol{A} \in \mathbb{R}^{n \times d}$ is a known data matrix and $\boldsymbol{\varepsilon} \in \mathbb{R}^n$ is an unknown noise vector. The high-dimensional setting corresponds to the case $n < d$, which in the linear model example implies that $\boldsymbol{A}$ has a nontrivial nullspace, hence the impossibility to learn $\boldsymbol{x}^\natural$, even in the absence of noise, without further assumptions.

Data fidelity is typically measured by a smooth and convex loss function $f : \mathbb{R}^d \to \mathbb{R}_+$, which corresponds to an empirical risk in machine learning and a data fitting term in signal processing.

Examples of smooth loss functions include the square loss $f(\boldsymbol{x}) = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2$ in regression problems, and the logistic loss $f(\boldsymbol{x}) = \sum_{i=1}^n \log(1 + \exp(-y_i \boldsymbol{x}^T \boldsymbol{a}_i))$ in classification problems.

Structured sparsity models are inherently combinatorial, and can thus be naturally encoded by set functions $F : 2^V \to \mathbb{R} \cup \{+\infty\}$ defined on the support $\operatorname{supp}(\boldsymbol{x}) = \{i : x_i \neq 0\}$. Incorporating such prior information in learning problems leads then to the following problem:

$$\min_{\boldsymbol{x} \in \mathbb{R}^d} f(\boldsymbol{x}) + \lambda F(\operatorname{supp}(\boldsymbol{x})), \tag{1.1}$$

where $\lambda \geq 0$ is a regularization parameter that controls the trade-off between data-fitting and regularization. $F(\operatorname{supp}(\boldsymbol{x}))$ then favors certain *supports*, or *non-zero patterns*, over others. For example, to favor sparse supports, the ℓ_0-pseudo-norm $F(\operatorname{supp}(\boldsymbol{x})) = |\operatorname{supp}(\boldsymbol{x})|$ can be used. To enforce hard constraints, F can be chosen to be an indicator function over a set of allowed supports; e.g., $F(\operatorname{supp}(\boldsymbol{x})) = \iota_{|\operatorname{supp}(\boldsymbol{x})| \leq s}(\boldsymbol{x})$. Problem (1.1) is computationally intractable[1] in general (see e.g., [Nat95]). Two main approaches, each with its own merits and shortcomings, have been adopted in the literature to confront this computational challenge. One is via *non-convex approaches* that provide approximate solutions directly to (1.1). The other is based on continuous *convex relaxations* where $F(\operatorname{supp}(\boldsymbol{x}))$ is replaced by a convex surrogate $g : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$, yielding the following *composite convex minimization* problem:

$$\min_{\boldsymbol{x} \in \mathbb{R}^d} f(\boldsymbol{x}) + \lambda g(\boldsymbol{x}), \tag{1.2}$$

The main benefit of non-convex approaches is that, by maintaining the combinatorial term in problem (1.1), they preserve the true structure model. This is particularly important in the case of structures that have no meaningful convex relaxations (such cases are identified in Chapter 3). Existing non-convex methods are based on iterative greedy algorithms, which are guaranteed to return approximate solutions to (1.1), but are only known to be tractable in special cases of structures. See Section 1.5 for further details.

Convex methods on the other hand can utilize a rich set of algorithmic tools guaranteed to return solutions of arbitrary accuracy to the relaxed convex problem (1.2), and analysis tools for characterizing the statistical efficiency of the resulting estimator. They also tend to be more robust to *model misspecifications*, which is likely to occur in practice.

The challenge in this approach resides in finding a convex surrogate, which can be efficiently optimized, while still preserving the structure encoded by F, which is crucial to guarantee statistical efficiency. We present some of the convex surrogates proposed in the literature to achieve this in Section 1.3, and review some methods to optimize the resulting convex problems in Section 1.4.

We will adopt the convex approach to structured sparsity throughout most of the thesis, except

[1] Throughout the thesis, we will use "intractable" to mean NP-Hard.

for the last two chapters, where we turn to non-convex approaches to handle structures with no meaningful convex relaxations.

1.2.2 Performance criteria

As we mentioned earlier, throughout the thesis, when considering an approach for learning with structured sparsity, we will be concerned with three factors: Computational efficiency, statistical efficiency, and generality. We now clarify what is meant by the first two notions.

Given a choice of F or g, and a solution $\hat{\boldsymbol{x}}$ returned by a proposed algorithm solving the corresponding optimization problem. Statistical efficiency is concerned with the number of samples required for $\hat{\boldsymbol{x}}$ to estimate $\boldsymbol{x}^\natural$ and its support, up to some target accuracy, while computational efficiency is concerned with the time needed to achieve this.

Let $\boldsymbol{x}^\star$ be a minimizer[2] of problem (1.1) for non-convex methods, and of problem (1.2) for convex ones, and $\mathcal{L}^\star$ the corresponding optimal objective value, e.g., $\mathcal{L}^\star = f(\boldsymbol{x}^\star) + \lambda g(\boldsymbol{x}^\star)$ in the convex approach case. We will split the discussion into the following two parts, as is often done in classical convex approaches[3].

Optimization performance: Given a proposed iterative[4] algorithm, let $\boldsymbol{x}^k$ be the solution obtained at an iteration k of the algorithm. We are interested in assessing the scalability of the algorithm with respect to the following, often conflicting, criteria:

- **Computational cost per iteration:** The performance of an iterative algorithm highly depends on the cost of computing $\boldsymbol{x}^k$ at each iteration, in terms of dependence on the dimensions d and n of the problem.
- **Number of iterations:** The performance of an iterative algorithm also depends on the number of iterations required to obtain a target numerical accuracy $\epsilon > 0$, either with respect to the objective error, i.e., $\mathcal{L}(\boldsymbol{x}^k) - \mathcal{L}^\star \leq \epsilon$ or the distance to the optimal point (if unique), i.e., $\|\boldsymbol{x}^k - \boldsymbol{x}^\star\| \leq \epsilon$. The number of iterations typically depends both on the accuracy ϵ and the dimensions d and n of the problem.

Note that, in large scale optimization problems, such as the ones considered in this thesis, the dependence in the above two criteria on the ambient dimension d is particularly important.

Statistical performance: We are interested in studying the performance of $\boldsymbol{x}^\star$ as an estimator of $\boldsymbol{x}^\natural$ in terms of the following criteria, where the probability is with respect to all random

[2] Without further assumptions, $\boldsymbol{x}^\star$ is not necessarily unique.
[3] Modern stochastic methods in machine learning tackle the computation and analysis of $\hat{\boldsymbol{x}}$ simultaneously.
[4] Optimization algorithms considered in this thesis are all iterative.

elements in the problem (e.g., noise and design matrix). For formal definitions, see, e.g., [Liu10].

- **Estimation consistency:** We say that $\boldsymbol{x}^\star$ is estimation consistent, if the *estimation error*, i.e., $\|\boldsymbol{x}^\star - \boldsymbol{x}^\natural\|$ converges in probability to zero.
- **Model selection consistency:** This criterion is also called sparsistency. We say that $\boldsymbol{x}^\star$ is sparsistent, if the *support recovery error*, i.e., $\|\mathbb{1}_{\mathrm{supp}(\boldsymbol{x}^\star)} - \mathbb{1}_{\mathrm{supp}(\boldsymbol{x}^\natural)}\|_0$ converges in probability to zero.

In the asymptotic regime, d is finite and fixed, and convergence in the above two criteria is with respect to $n \to \infty$. This regime forbids the high-dimensional setting, and is thus less interesting in the context of structured sparsity. Nevertheless, studying it typically requires simpler analysis, and is helpful to develop insights which contribute to the understanding of the high-dimensional setting. In the *non-asymptotic* regime, we are interested in the rate of convergence of the errors in the above two criteria, as a function of the ambient dimension d, the number of samples n, and the sparsity $s = |\,\mathrm{supp}(\boldsymbol{x}^\natural)|$ (or more generally the "complexity" of $\boldsymbol{x}^\natural$ under the assumed structured sparsity model). In particular, the number of samples required to recover (up to some accuracy) $\boldsymbol{x}^\natural$, as a function of d and s is called *sample complexity*. Other performance criteria, such as prediction error, are also of interest, but we will focus on these two criteria in our discussion.

1.2.3 Penalized and constrained formulations

Note that by allowing F and g to take infinite values, problems (1.1) and (1.2) include two regularization variants. Given our convex loss function f and a regularizer $\Omega : \mathbb{R}^d \to \mathbb{R}$, the penalized variant, also known as the Lagrangian form, is given by:

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x}) + \lambda\Omega(\boldsymbol{x}), \tag{1.3}$$

while the constrained variant is given by:

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} \{f(\boldsymbol{x}) : \Omega(\boldsymbol{x}) \leq \tau\}. \tag{1.4}$$

If Ω is convex then, under some mild conditions, the two variants are equivalent, in the sense that $\boldsymbol{x}^\star$ is a solution of problem (1.3) for some $\lambda > 0$, if and only if, it is a solution of problem (1.4) for some $\tau > 0$ [BL10, Sect. 4.3]. However, the exact relation between λ and τ is not known.

In practice, the choice between one or the other variant, depends on factors such as computational complexity, stability, and robustness. For example, depending on Ω, one variant can be easier to solve than the other. Moreover, the solution of problem (1.3) is less sensitive to small changes in λ, than the solution of problem (1.4) to small changes in τ, which makes tuning λ easier than tuning τ, in practice. Also, when f is the least squares loss, the penalized formulation can be more robust to model misspecifications [LST13]. In these cases, the penalized formulation is

thus preferable. On the other hand, if one is interested in enforcing a fixed bound τ on $\Omega(\boldsymbol{x}) \leq \tau$, the constrained formulation is then preferable. This is particularly relevant in the non-convex setting, where we might be interested for example in obtaining solutions that are exactly $\boldsymbol{s}$-sparse.

1.3 Convex approaches

In this section, we present several approaches adopted in the literature to design convex surrogates of structured sparsity models. In each case, we will pay attention to the structures that can be expressed by the presented convex penalty and its statistical properties, if known. We defer the discussion of how to optimize the resulting convex problems to Section 1.4.

1.3.1 Popular structured sparsity-inducing norms

A classical approach for choosing a convex surrogate consists of designing a norm that leads to the desired set of non-zero patterns in a "reverse-engineering" manner. This approach led to the design of several interesting structured sparsity-inducing norms. We outline below some of them

Lasso (ℓ_1-norm)

We start with the most popular example of sparsity-inducing norms, the ℓ_1-norm, which is used as a convex surrogate of the ℓ_0-pseudo-norm. In this case, when f is the least-squares loss, problem (1.2) reduces to the following formulation, known as basis pursuit denoising (BPDN)[5] [CDS98].

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{x}\|_1. \tag{1.5}$$

Another closely related formulation is known as the least absolute shrinkage and selection operator (Lasso) [Tib96]:

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} \{\frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 : \|\boldsymbol{x}\|_1 \leq \tau\}. \tag{1.6}$$

In Sections 1.3.3 and 1.3.4, we present a formal justification as to why the ℓ_1-norm is the "best" convex surrogate for the ℓ_0-pseudo-norm. We present here some intuitive reasons for why the ℓ_1-norm induces sparsity. From an analytical perspective, we can see this by considering the simple denoising example, where $\boldsymbol{A} = \boldsymbol{I}$ in the BPDN formulation (1.5). The solution in this case is given by the soft-thresholding operator, introduced by [DJ95], $\boldsymbol{x}^\star(\lambda) = \mathrm{sign}(\boldsymbol{y}) \circ \max\{|\boldsymbol{y}| - \lambda, 0\}$. For large enough λ, $\boldsymbol{x}^\star(\lambda)$ is sparse, since all coefficients $|y_i| \leq \lambda$ are set to zero. This behavior mimics the solution obtained by regularizing instead with the ℓ_0-pseudo-norm, $\boldsymbol{x}^\star(\lambda) = \boldsymbol{y} \circ \mathbb{1}_{\{i:|y_i|\geq\sqrt{2\lambda}\}}$, where all coefficients $|y_i| \leq \sqrt{2\lambda}$ are set to zero. The difference between the

[5] BPDN is also sometimes used to refer to the following formulation: $\min_{\boldsymbol{x}\in\mathbb{R}^d}\{\|\boldsymbol{x}\|_1 : \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 \leq \tau\}$.

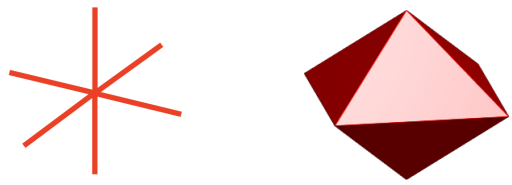


Figure 1.3: Unit balls of ℓ_0-"norm", restricted to the unit ℓ_∞-ball (left), and ℓ_1-norm (right).

two solutions is the additional shrinkage effect imposed in the ℓ_1-solution.

From a geometric perspective, it is easy to see that the unit ℓ_1-ball is the convex hull of the standard basis vectors, which are one-sparse, or equivalently of the unit ℓ_0-ball, when restricted inside the unit ℓ_∞-ball, i.e., the set $\{\boldsymbol{x} : \|\boldsymbol{x}\|_0 \leq 1, \|\boldsymbol{x}\|_\infty \leq 1\}$ (see Figure 1.3). In fact, from this perspective, ℓ_1-norm is a special case of a class of norms defined as convex hulls of vectors whose support satisfy the desired structure, which we discuss in Section 1.3.2.

The convex approach proved to be successful in this case. Indeed, replacing the ℓ_0-pseudo-norm with the ℓ_1-norm, allows efficient robust recovery of any s-sparse vector $\boldsymbol{x}^\natural \in \mathbb{R}^d$, in the linear model case, using only $n = O(s \log(d/s))$ samples, under some assumptions on $\boldsymbol{A}$ (e.g., restricted isometry property) [CT05, Don06].This sample complexity can be significantly smaller than the classical Shannon-Nyquist sampling bound, which dictates uniformly sampling a signal at a rate at least twice its highest frequency in the Fourier domain [Sha49].

In the past decade, substantial work was done towards extending this success to more involved structures, with various convex penalties proposed in the literature (see [OB16] and [KBEH+15] for an overview).

Group Lasso

A simple extension of the Lasso is the group Lasso [YL06], also called ℓ_1/ℓ_p-norm, defined as:

$$\Omega_p^{\cap}(\boldsymbol{x}) = \sum_{G \in \mathfrak{G}} d_G \|\boldsymbol{x}_G\|_p, \tag{1.7}$$

where $\mathfrak{G}$ is a collection of non-overlapping groups that partition V, $(d_G)_{G \in \mathfrak{G}}$ are positive weights, and where $p \geq 1$, with $p \in \{2, \infty\}$ being popular choices in practice.

$\Omega_p^{\cap}$ acts as an ℓ_1-norm (when weights are equal) over the terms $\|\boldsymbol{x}_G\|_p$, and hence it promotes sparsity on the group level. This structure, dubbed as block-sparsity, was shown to improve the estimation performance over standard Lasso, when $\boldsymbol{x}^\natural$ is block-sparse, both in terms of sampling complexity and noise robustness [SPH09, EB09, HZ10]. Block-sparsity arises in applications such as DNA microarrays [PVMH08], equalization of communication channels [CR02], multi-task learning [OTJ10], and multiple kernel learning [Bac08].

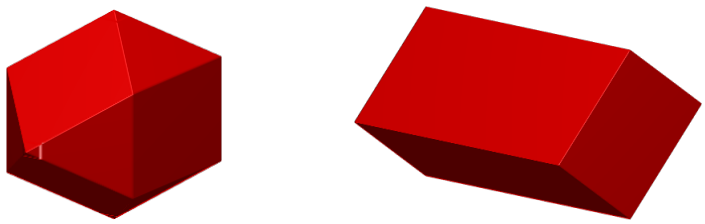


Figure 1.4: Unit ball of ℓ_1/ℓ_∞-norm (left) and ℓ_∞-LGL norm (right), for $\mathfrak{G} = \{\{1,2\},\{2,3\}\}$.

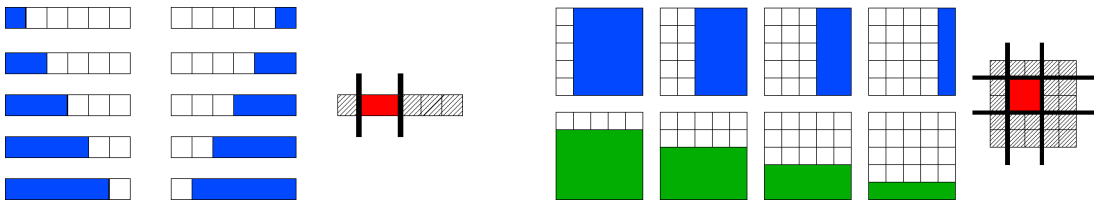


Figure 1.5: The sets in blue or green are the groups to include in $\mathfrak{G}$, along with their complements, to select interval (left) or rectangular (right) patterns, as proposed in [JAB11]. The sets in red are examples of the corresponding induced non-zero patterns. This figure is taken from [OB16].

Group Lasso was further generalized to the case of *overlapping* groups in [ZRY09, JOV09, JAB11, MJOB11]. Figure 1.4 (left) displays the unit ℓ_1/ℓ_∞-norm ball, for an example of overlapping groups. This norm was shown, in [JAB11], to induce supports corresponding to the *intersection* of a sub-collection of the complements of groups in $\mathfrak{G}$. Conversely, given a *intersection-closed*[6] set of non-zero patterns, it is possible to engineer the groups in $\mathfrak{G}$ in order to favor these patterns via $\Omega_p^{\cap}$.

For example, [JAB11] showed that using the groups displayed in Figure 1.5 (left) induces *interval* patterns; a structure desirable in applications such as time series, or cancer diagnosis [RBV08]. Similarly, using the groups displayed in Figure 1.5 (right) induces rectangular patterns; a structure desirable in applications such as background subtraction [CHDB09, MJBO10], dictionary learning [MJOB11] and face recognition [JOB10]. $\Omega_p^{\cap}$ can also be used to induce hierarchical structures which we discuss below.

The statistical properties of $\Omega_p^{\cap}$ were studied in [JAB11], with conditions for consistent estimation of the non-zero patterns presented, both in low and high-dimensional settings. In Section 1.3.3, we see why the overlapping ℓ_1/ℓ_∞-norm is the "best" convex surrogate, in some sense, for group-intersection structures.

Latent group Lasso

As mentioned above, overlapping ℓ_1/ℓ_p-norm can induce supports belonging to an intersection-closed set of supports, while in several applications, non-zero patterns corresponding to the

[6] A set $\mathcal{S}$ is intersection-closed if $\forall S_1, S_2 \in \mathcal{S}, S_1 \cap S_2 \in \mathcal{S}$.

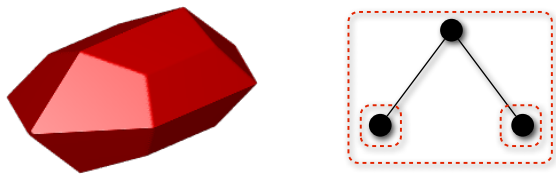


Figure 1.6: Unit ball of $\Omega_\infty^\cap$ (left), and corresponding groups $\mathfrak{G}_H = \{\{1,2,3\},\{2\},\{3\}\}$ (right).

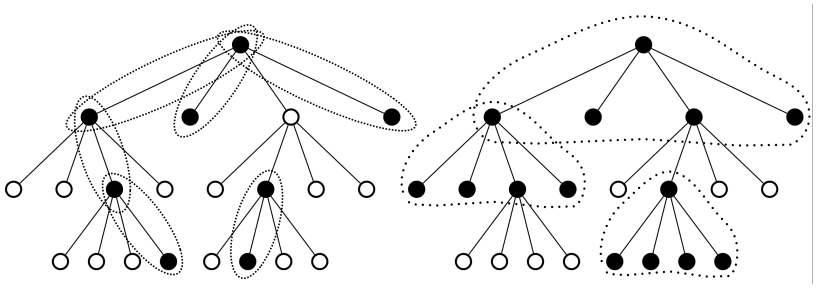


Figure 1.7: Examples of parent-child and family models. Active groups are indicated by dotted ellipses. The support (black nodes) is given by the union of the active groups. (Left) Parent-child model. (Right) Family model.

union of a sub-collection of groups in $\mathfrak{G}$ are more desirable. This is particularly important, in applications such as cancer prognosis from high-dimensional gene expression data, where the groups are naturally predefined, e.g., genes involving the same biological function should be grouped together, as opposed to being manually chosen. This motivated another generalization of group Lasso to overlapping groups, given by the latent group Lasso (LGL), introduced by [JOV09] (see also [OJV11]). Given a collection of groups $\mathfrak{G}$, and associated positive weights $(d_G)_{G\in\mathfrak{G}}$, the LGL norm is defined as;

$$\Omega_p^\cup(\boldsymbol{x}) = \min_{\boldsymbol{v}\in\mathbb{R}^{d\times|\mathfrak{G}|}} \{\sum_{G\in\mathfrak{G}} d_G\|\boldsymbol{v}^G\|_p : \sum_{G\in\mathfrak{G}} \boldsymbol{v}^G = |\boldsymbol{x}|, \mathrm{supp}(\boldsymbol{v}^G) \subseteq G\}. \tag{1.8}$$

Note that $\Omega_p^\cup$ and $\Omega_p^\cap$ are equal in the case of *non-overlapping* groups, but in general they are different, as it is apparent for example from Figure 1.4. $\Omega_p^\cup$ can also be used to induce hierarchical structures as we discuss next.

A detailed analysis of $\Omega_p^\cup$ and its statistical properties, in terms of *group-support* recovery, in the low-dimensional setting, is presented in [OJV11]. In Sections 1.3.4 and 2.5.1, we see why the latent group Lasso is the "best" convex surrogate, in some sense, for group-cover structures.

Hierarchical sparsity

In a hierarchical sparsity model, the variables or groups of variables are organized over a directed *tree*, (or a forest) $\mathcal{T}$, and they satisfy hierarchical relations, e.g., an element can be selected only

if all its ancestors in $\mathcal{T}$ are also selected; this is known as the rooted connected tree structure. The more general case where variables are organized over a directed acyclic graph was also studied, see, e.g., [YB+17] and [OB16].

Hierarchical structures are found in many applications, such as image processing, to exploit the multi-scale structure of wavelet coefficients, see Figure 1.2 and [DWB08, ZRY09, BCDH10, JMOB11]; bioinformatics, to leverage the hierarchical structure of gene networks for multitask regression [KX10]; deep learning, where hierarchies of latent variables are used in convolutional neural networks [Ben09].

Such structure was shown to result in better performance than standard Lasso, both in terms of noise robustness and sample complexity. In particular, [BCDH10] showed that we can recover any s-sparse vector $\boldsymbol{x}^\natural \in \mathbb{R}^d$ that satisfies a rooted connected tree structure, using only $n = O(s)$ samples, in the linear model case, and under similar assumptions on $\boldsymbol{A}$ as in Lasso.

Both overlapping group Lasso and latent group Lasso were used in the literature to induce hierarchical structures. In particular, if we define groups in $\mathfrak{G}_H$ as each node and all its *descendants* in $\mathcal{T}$, then the corresponding $\Omega_p^{\cap}$ results in the hierarchical group lasso [ZRY09]. Figure 1.6 displays an example of the descendants groups and the corresponding unit ball of $\Omega_\infty^{\cap}$. In Section 2.5.2, we see why the ℓ_∞-hierarchical group Lasso is the "best" convex surrogate, in some sense, for the rooted connected tree structure.

On the other hand, the latent group Lasso norm was used with groups defined in $\mathfrak{G}_A$ as each node and all its *ancestors* in $\mathcal{T}$, to induce hierarchical structures. A systematic comparison of $\Omega_2^{\cap}$ with $\mathfrak{G}_H$, and $\Omega_2^{\cup}$ with $\mathfrak{G}_A$, and their statistical properties, is presented in [YB+17]. In particular, it is shown that though the two penalties are not identical with these groups, they do lead to the same class of non-zero patterns, with the difference that group Lasso shrink more aggressively parameters deep in the hierarchy.

Other group structures were also used with the latent group Lasso norm. For example, a parent-child model, where groups consist of all parent-child pairs in $\mathcal{T}$ (see Figure 1.7, left), and a family model, where groups consist of each node and all its children (see Figure 1.7, right), were proposed in order to favor tree-structures, but also allowing for a certain degree of flexibility in deviations from the rooted connected tree structure [BBC+16, RNWK11].

Exclusive Lasso

In certain applications, it is desirable to induce non-zero patterns, where sparse coefficients within a group compete against each other, i.e., a non-zero entry discourages other entries, within the same group, to be non-zero. For example, this structure naturally arises in neurobiology. Inspired by the statistical analysis in [GK02], the authors in [HDC09] consider a simple one-dimensional model, where a neuronal signal behaves as a train of spike signals with some refractory period $\Delta > 0$: there is a minimum non-zero time period Δ where a neuron remains inactive between

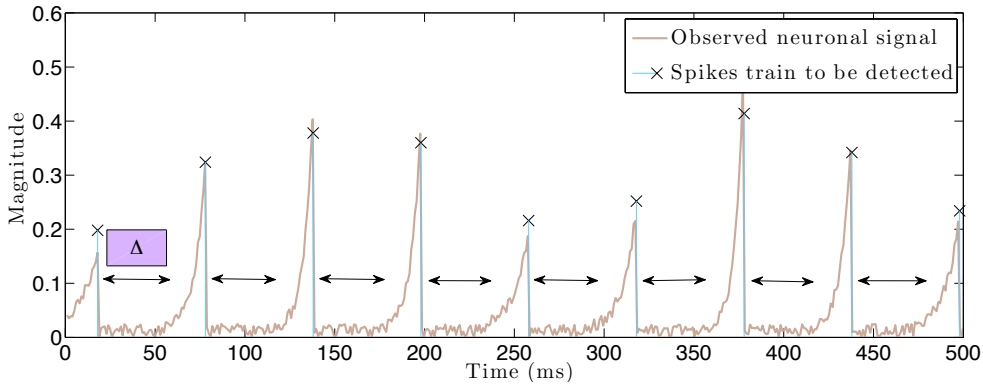


Figure 1.8: Neuronal spike train example

two consecutive electrical excitations. Figure 1.8 illustrates how a collection of noisy neuronal spike signals with $\Delta > 0$ might appear in practice.

The exclusive Lasso, also called ℓ_p/ℓ_1-norm, proposed by [ZJH10][7], promotes such *dispersive* structure. Given a collection of groups $\mathfrak{G}$, and associated positive weights $(d_G)_{G\in\mathfrak{G}}$, exclusive Lasso is defined as:

$$\Omega_p^{\text{exclusive}}(\boldsymbol{x}) = (\sum_{G\in\mathfrak{G}} \|\boldsymbol{x}_G\|_1^p)^{1/p}. \tag{1.9}$$

$\Omega_p^{\text{exclusive}}$ acts as an ℓ_1-norm on each group, and thus promotes sparsity within each group. In Sections 1.3.4 and 2.5.3, we see why the exclusive Lasso is the "best" convex surrogate, in some sense, for dispersive structures.

1.3.2 Atomic norms

A more principled general approach for choosing a convex surrogate of a structured sparsity model was proposed in [CRPW12]. This approach considers models where $\boldsymbol{x}^\natural$ is "simple" in the sense that it can be written as the sum of a few *atoms* from an atomic set $\mathcal{A}$, with possibly infinite atoms, i.e., $\boldsymbol{x}^\natural = \sum_{i=1}^s c_i \boldsymbol{a}^i$. The proposed convex penalty is then given by the gauge of the convex hull of the atomic set:

$$\|\boldsymbol{x}\|_{\mathcal{A}} = \inf\{t \geq 0 : \boldsymbol{x} \in t\,\text{conv}(\mathcal{A})\}, \tag{1.10}$$

which, assuming without loss of generality that the centroid of $\mathcal{A}$ is zero, can be rewritten as:

$$\|\boldsymbol{x}\|_{\mathcal{A}} = \inf\{\sum_{\boldsymbol{a}\in\mathcal{A}} c_{\boldsymbol{a}} : \boldsymbol{x} = \sum_{\boldsymbol{a}\in\mathcal{A}} c_{\boldsymbol{a}}\boldsymbol{a}, c_{\boldsymbol{a}} \geq 0, \forall \boldsymbol{a} \in \mathcal{A}\}, \tag{1.11}$$

$\|\boldsymbol{x}\|_{\mathcal{A}}$ is a norm, called the *atomic norm*, whenever $\mathcal{A}$ is centrally symmetric around the origin, i.e., when $\boldsymbol{a} \in \mathcal{A}$ if and only if $-\boldsymbol{a} \in \mathcal{A}$. This convex penalty is in general *intractable* to

[7] In the original definition of exclusive Lasso in [ZJH10], $\mathfrak{G}$ is a partition of V.

evaluate and to optimize (e.g., when $\mathrm{conv}(\mathcal{A})$ is the cut polytope), but for certain cases of $\mathcal{A}$ it can be evaluated exactly, or approximated via semidefinite programming, and the resulting convex optimization problems are tractable (see [CRPW12] for details). Moreover, for certain choices of $\mathcal{A}$, the atomic norm recovers popular structured sparsity-inducing norms, such as:

(i) ℓ_1**-norm:** If $\mathcal{A}$ is the set of standard basis vectors, i.e., $\mathcal{A} = \{\boldsymbol{a} \in \mathbb{R}^d : \|\boldsymbol{a}\|_0 = 1, \|\boldsymbol{a}\|_p \leq 1\}$ (for any $p \geq 1$), then $\|\boldsymbol{x}\|_{\mathcal{A}} = \|\boldsymbol{x}\|_1$.

(ii) **Latent group Lasso norm:** For a collection of groups $\mathfrak{G}$, and positive weights $(d_G)_{G\in\mathfrak{G}}$, if $\mathcal{A} = \{\boldsymbol{a} \in \mathbb{R}^d : \mathrm{supp}(\boldsymbol{a}) \subseteq G, \|\boldsymbol{a}\|_p = d_G^{-1},$ for some $G \in \mathfrak{G}\}$, then $\|\boldsymbol{x}\|_{\mathcal{A}} = \Omega_p^{\cup}(\boldsymbol{x})$ defined in Eq. (1.8), as shown in [OB12].

Furthermore, Section 1.3.4 introduces another class of structure-inducing norms, which can be viewed as atomic norms too. For other interesting examples, we refer the reader to [CRPW12]. Conditions for exact and robust recovery of $\boldsymbol{x}^\natural$ using general atomic norms, along with bounds on the corresponding sample complexity are presented in [CRPW12].

The main drawback to this approach is that the proposed convex penalties are gauge functions and thus are necessarily positively homogeneous; we discuss why this is problematic, in terms of capturing general structures, in Chapters 2 and 3.

1.3.3 Convex relaxations of submodular penalties

Another systematic approach, proposed in [Bac10a], considers choosing the convex surrogate to be the *tightest convex relaxation* of the desired structured sparsity model. Namely, given a positive-valued set function $F : 2^V \to \mathbb{R}_+$, such that $F(\emptyset) = 0$, and $F(A) > 0, \forall A \subseteq V$, encoding the structure on the support of $\boldsymbol{x} \in \mathbb{R}^d$, this approach proposes to use the *convex envelope* of $F(\mathrm{supp}(\boldsymbol{x}))$, i.e., its largest (thus tightest) convex lower bound[8], over the unit ℓ_∞-ball, as a natural convex surrogate for it.

In particular, this approach is applied in [Bac10a] for structures that can be expressed by a *monotone submodular* function (see definitions in Section 1.1). The convex envelope of a function is given by its biconjugate, i.e., the Fenchel conjugate of the Fenchel conjugate. Let $F_\infty(\boldsymbol{x}) = F(\mathrm{supp}(\boldsymbol{x})) + \iota_{\|\boldsymbol{x}\|_\infty \leq 1}(\boldsymbol{x})$, then the convex envelope of F_∞ is given by $\Theta_\infty := F_\infty^{**}$. Note that restricting the values of $\boldsymbol{x}$ to a bounded domain (e.g., unit ℓ_∞-ball) is a necessary technical requirement for deriving non-trivial relaxations of such functions. Unfortunately, the convex envelope Θ_∞ is in general *intractable* to evaluate and to optimize[9]. However, for the class of monotone submodular functions, [Bac10a] shows that Θ_∞ is given by the *Lovász extension*

[8] Throughout the thesis, we will thus use convex envelope and tightest convex relaxation interchangeably.

[9] We can see this for example from the connection between the convex envelope of F_∞ and the convex closure of F, which we establish in Chapter 2.

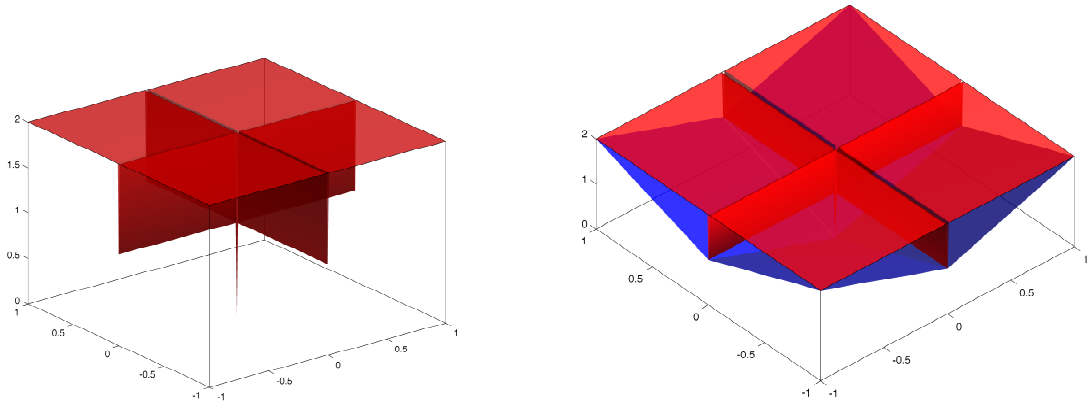


Figure 1.9: ℓ_0-pseudo-norm (red) and its convex envelope over the unit ℓ_∞-ball; the ℓ_1-norm (blue), in $\mathbb{R}^2$.

[Lov83] of F, $\Theta_\infty(\boldsymbol{x}) = f_L(|\boldsymbol{x}|)$, where f_L is defined as follows:

$$f_L(\boldsymbol{x}) = \sum_{k=1}^{d-1} x_{j_k}[F(\{j_1, \dots, j_k\}) - F(\{j_1, \dots, j_{k-1}\})], \tag{1.12}$$

where $\boldsymbol{x}$ is sorted in decreasing order $x_{j_1} \geq \cdots \geq x_{j_d}$. When F is submodular, f_L is known to be convex. For further details about the Lovász extension see Appendix A.1.

We can see from the definition of the Lovász extension in (1.12), that Θ_∞ in this case is a norm, and is efficiently computable. Tractable algorithms to solve the resulting convex optimization problems, regularized with Θ_∞, are also proposed in [Bac10a] (see also Section 1.4.1). Moreover, for certain choices of the submodular function F, Θ_∞ recovers popular structured sparsity-inducing norms, such as:

(i) ℓ_1**-norm:** If F is the cardinality function, $F(S) = |S|$, then $\Theta_\infty(\boldsymbol{x}) = \|\boldsymbol{x}\|_1$. Figure 1.9 displays the ℓ_0-pseudo-norm, restricted within the ℓ_∞-ball, and its convex envelope the ℓ_1-norm. It is easy to see from this figure, why considering the convex envelope over an unbounded domain would simply yield the zero function.

(ii) ℓ_1/ℓ_∞**-norm:** For a collection of groups $\mathfrak{G}$, and positive weights $(d_G)_{G\in\mathfrak{G}}$, if F is the overlap count function, $F(S) = \sum_{G\in\mathfrak{G}, G\cap S\neq\emptyset} d_G$, then $\Theta_\infty = \Omega_\infty^\cap$, defined in Eq. (1.7).

For other interesting examples, we refer the reader to [Bac10a]. In Chapter 2, we show that this approach is also tractable for another class of interesting set functions.

Note that the approach outlined here is similar to the one in Section 1.3.2, in the sense that both approaches attempt to compute the "best" convex surrogate of a given structured sparsity model. Indeed, the notion of convex envelope extends the notion of convex hull of sets to functions. In

particular, the epigraph of the convex envelope Θ_∞ corresponds to the closure of the convex hull of the epigraph of F_∞. Moreover, the atomic norm $\|\boldsymbol{x}\|_{\mathcal{A}}$ of a compact atomic set $\mathcal{A}$ is the convex envelope of the function $\boldsymbol{x} \to \inf\{t \geq 0 : \boldsymbol{x} \in t\mathcal{A}\}$. However, a key difference is that, unlike atomic norms, convex penalties obtained as convex envelopes of $F(\mathrm{supp}(\boldsymbol{x}))$ are not necessarily norms, when F is not monotone submodular, as we show in Chapter 2. In that chapter, we identify another class of set functions, for which the convex envelope Θ_∞ is efficiently computable.

The statistical properties of Θ_∞, with conditions for support recovery and estimation, in the high dimensional setting, are presented in [Bac10a]. In particular, it is shown that the non-zero patterns allowed with these norms correspond to the *stable* sets of F, i.e., sets $A \subseteq V$ that satisfy $\forall B \supset A, F(B) > F(A)$.

1.3.4 Homogeneous convex relaxations of ℓ_p-regularized penalties

A similar principled approach to the one discussed above, proposed in [OB12], considers ℓ_p-regularized general combinatorial functions of the form $F_p(\boldsymbol{x}) = \frac{1}{q}F(\mathrm{supp}(\boldsymbol{x})) + \frac{1}{p}\|\boldsymbol{x}\|_p$, for $p \geq 1$, where as before the set function F controls the structure of the model in terms of allowed/favored non-zero patterns, and the additional ℓ_p-norm serves to control the magnitude of the coefficients. For $p = \infty$, F_p reduces to $F_\infty(\boldsymbol{x}) = F(\mathrm{supp}(\boldsymbol{x})) + \iota_{\|\boldsymbol{x}\|_\infty \leq 1}(\boldsymbol{x})$. The choice $p \neq \infty$ might be preferable to avoid the clustering artifacts of the values of the learned vector induced by the ℓ_∞-norm.

This approach proposes to use the largest *positively homogeneous* convex lower bound of F_p as a convex surrogate for it. This is achieved by computing first the positively homogeneous envelope of F_p, i.e., its largest positively homogeneous lower bound, given by $F(\mathrm{supp}(\boldsymbol{x}))^{1/q}\|\boldsymbol{x}\|_p$, then computing the corresponding convex envelope. We call the resulting convex penalty the *homogeneous convex envelope* of F_p, and denote it by Ω_p.

[OB12] showed that the homogeneous convex envelope of F_p, for any set function F, is given by a *generalized latent group Lasso norm*:

$$\Omega_p(\boldsymbol{x}) = \min_{\boldsymbol{v}}\{\sum_{S \subseteq V} F(S)^{1/q}\|\boldsymbol{v}^S\|_p : \sum_{S \subseteq V} \boldsymbol{v}^S = |\boldsymbol{x}|, \mathrm{supp}(\boldsymbol{v}^S) \subseteq S\}. \tag{1.13}$$

Note that the norm in Eq. (1.13) indeed corresponds to a latent group Lasso norm as defined in Eq. (1.8), with $\mathfrak{G}$ containing all the power-set of V. Moreover, Ω_p can be viewed as an atomic norm, associated with the atomic set $\mathcal{A} = \{\boldsymbol{a} \in \mathbb{R}^d : \mathrm{supp}(\boldsymbol{a}) \subseteq S, \|\boldsymbol{a}\|_p = 1, \text{ for some } S \in \mathcal{D}_F\}$, where $\mathcal{D}_F$ is the *core* set of F, corresponding to the set of faces of a polytope associated with Ω_p. See [OB16, Section 2.3] for the precise definition.

Unfortunately, the homogeneous convex envelope Ω_p is also in general *intractable* to evaluate and to optimize. However, if F is a monotone submodular function, a tractable *decomposition algorithm* to compute Ω_p for any $p \geq 1$, was provided in [OB16, Section 6.3]. This algorithm

requires solving a sequence of at most d submodular minimization problems, which can be done in polynomial time for general submodular functions F (see Section A.1). Tractable algorithms to solve the resulting convex optimization problems, regularized with Ω_p, are also proposed in [OB16] (see also Section 1.4.1). Moreover, for certain choices of F (including non-submodular ones), Ω_p recovers popular structured sparsity-inducing norms, such as:

(i) ℓ_1**-norm:** If F is the cardinality function, $F(S) = |S|$, then $\Omega_p(\boldsymbol{x}) = \|\boldsymbol{x}\|_1$ for *any* $p \geq 1$.

(ii) **Submodular norms:** If F is monotone submodular, then the homogeneous convex envelope coincides with the convex envelope, when $p = \infty$, i.e., $\Omega_\infty = \Theta_\infty = f_L(|\cdot|)$, and it is then easily computable (the decomposition algorithm is not needed in this case). For general p though, the two envelopes are not necessarily equal (see next example).

(iii) ℓ_1/ℓ_p**-norm:** For a collection of groups $\mathfrak{G}$, and positive weights $(d_G)_{G\in\mathfrak{G}}$, if F is the submodular overlap count function, $F(S) = \sum_{G\in\mathfrak{G}, G\cap S\neq\emptyset} d_G$, then $\Omega_p = \Omega_p^{\cap}$, defined in Eq. (1.7), only if $\mathfrak{G}$ is a partition of V. Otherwise, for overlapping groups, this identity does not hold for $p < \infty$. In this case, the norm, called overlap count Lasso, does not have a simple closed form in general (but it can be computed using the decomposition algorithm). For more details on the difference between overlap count Lasso and ℓ_1/ℓ_p-norms, see [OB16, Section 4.1].

(iv) **Latent group Lasso norm:** If F is the (non-submodular) minimal weighted set cover function, $F(S) = \min_{\boldsymbol{\omega}\in\{0,1\}^{|\mathfrak{G}|}}\{\sum_{G\in\mathfrak{G}} d_G\omega_G : \sum_{G\in\mathfrak{G}} \omega_G \mathbb{1}_G \geq \mathbb{1}_S\}$, then $\Omega_p = \Omega_p^{\cup}$, defined in Eq. (1.8). Note that this norm is not tractable in general (e.g., when $\mathfrak{G} = 2^V$ as in Eq. (1.13), but for example if the number of groups in $\mathfrak{G}$ is polynomial, it can be computed by linear programming.

(v) ℓ_p/ℓ_1**-norm:** If F is the (non-submodular) function $F(S) = \max_{G\in\mathfrak{G}} |S \cap G|$, and $\mathfrak{G}$ is a partition of V, then $\Omega_p = \Omega_p^{\text{exclusive}}$, defined in Eq. (1.9).

For other interesting examples, we refer the reader to [OB16]. An extension of the statistical results presented in [Bac10a] for Θ_∞, in the case of monotone submodular functions, to Ω_p for the same class of functions, with any $p \geq 1$, is provided in [OB16].

This approach has a similar drawback to the approach of atomic norms, namely that the proposed convex penalties are necessarily positively homogeneous; we discuss why this is problematic, in terms of capturing general structures, in Chapter 3. In that chapter, we study the non-homogeneous counterpart of Ω_p, i.e., the convex envelope of F_p, for general set functions, and any $p \geq 1$.

1.4 Convex optimization for structured sparsity

In the previous section, we reviewed existing approaches to convert the combinatorial term F in problem (1.1) to a convex term g. The resulting convex penalties are naturally *non-smooth*. In

this section, we review two first-order methods[10] tailored to optimize the corresponding convex problem (1.2), where f is a smooth differentiable function, and g is a non-smooth convex function. In particular, f is assumed to have a Lipschitz continuous gradient with respect to $\|\cdot\|$, i.e., $\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|_* \leq L\|\boldsymbol{x} - \boldsymbol{y}\|, \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, for some $L > 0$, and where $\|\cdot\|_*$ is the dual norm of $\|\cdot\|$. In some cases, f can be further assumed to be *strongly-convex* with respect to $\|\cdot\|$, i.e., $f(\boldsymbol{x}) \geq f(\boldsymbol{y}) + \langle \boldsymbol{x} - \boldsymbol{y}, \nabla f(\boldsymbol{y})\rangle + \frac{\mu}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2, \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, for some $\mu > 0$.

In large-scale problems, first-order methods are often preferred to second-order methods, despite requiring a larger number of iterations to achieve a given accuracy, due to their lower-complexity per iteration. In the context of structured-sparsity, existing first-order optimization algorithms often fall under the categories of *proximal gradient methods* (also known as *forward–backward splitting* methods) and *Frank-Wolfe* (FW) methods (also known as *conditional gradient* methods).

In what follows, we present these two methods, and highlight their respective advantages and drawbacks. For each method, we will pay attention to its computational complexity per iteration, its convergence rate guarantees, the sparsity of its solutions, and its ability to handle *general* norms. Our interest in the last two properties is motivated by the following: First, optimization algorithms which maintain sparse iterates are desirable in structured sparsity problems, where we indeed seek sparse solutions, but also in general large-scale problems, due to the computational benefit this property entail. Second, algorithms which allow arbitrary choices of the norm $\|\cdot\|$, instead of the classical ℓ_2-norm, enable us to adapt the norm to the geometry of the given problem. This property can lead to significant improvement in the convergence rate in terms of dimension dependence, as observed for example in [KLOS14, Nes05, BWB14, dGJ13].

1.4.1 Proximal gradient methods

Proximal methods date back to [Mar70]. They have received a lot of attention both from the machine learning and signal processing communities, see, e.g., [WNF09, MRS+10, Bac10b, CP11], due to their relatively fast convergence rate and their suitability for large-scale non-smooth convex problems of the form (1.2).

Algorithm: The main idea of proximal gradient methods is to minimize at each iteration k, a quadratic approximation of f, tight at the current iterate $\boldsymbol{x}^k$, while leaving g intact. Assuming the gradient of f is L_2-Lipschitz continuous, with respect to the ℓ_2-norm, implies the following quadratic majorizer of f, at any point $\boldsymbol{y} \in \mathbb{R}^d$, and $\forall \gamma \in (0, 1/L_2]$:

$$f(\boldsymbol{x}) \leq f(\boldsymbol{y}) + \langle \boldsymbol{x} - \boldsymbol{y}, \nabla f(\boldsymbol{y})\rangle + \frac{1}{2\gamma}\|\boldsymbol{x} - \boldsymbol{y}\|_2^2. \tag{1.14}$$

[10] First-order methods are methods that only use the first derivative of the objective function.

The following problem is then solved at each iteration k:

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} \frac{1}{2\gamma^k}\|\boldsymbol{x} - (\boldsymbol{x}^k - \gamma^k \nabla f(\boldsymbol{x}^k))\|_2^2 + \lambda g(\boldsymbol{x}) \tag{1.15}$$

In the basic proximal gradient method, the next iterate is set to the unique (by strong convexity) solution of (1.15). In the special case where g is an indicator function $g = \iota_C$, this algorithm is called *projected gradient method*, since problem (1.15) reduces to a projection on C. Also if $g = 0$, we recover the standard gradient descent algorithm. Accelerated variants of the proximal gradient method, such as [Tse08, BT09a, Nes13], include an additional extrapolation step. For example, one simple version is to set the next iterate to the solution of problem (1.15) with $\boldsymbol{x}^k$ replaced with $\boldsymbol{x}^k + \alpha_k(\boldsymbol{x}^k - \boldsymbol{x}^{k-1})$, for some carefully chosen $\alpha_k \in (0, 1]$.

When the Lipschitz constant L_2 is not known, the step size γ^k can be found by line search. For example, one simple line search strategy, proposed by [BT09b, Section 1.4.3], consists of iteratively increasing γ^k until the bound in (1.14) holds. Such adaptive choice of the step size can be helpful even in cases where L_2 is known, because it allows the algorithm to adapt to the local properties of the objective.

The solution of (1.15) corresponds to an instance of the *proximal operator* [Mor62] of g:

$$\mathrm{prox}_{\lambda g}(\boldsymbol{u}) := \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{u}\|_2^2 + \lambda g(\boldsymbol{x}) \tag{1.16}$$

When g is an indicator function $g = \iota_C$, Eq. (1.16) becomes a projection on C, and is denoted by $\mathrm{proj}_C(\boldsymbol{u})$. Note that this operator is well-defined, since the solution in (1.16) is unique. The iterates of the basic proximal gradient method can then be written as

$$\boldsymbol{x}^{k+1} = \mathrm{prox}_{\lambda g}(\boldsymbol{x}^k - \gamma^k \nabla f(\boldsymbol{x}^k)),$$

and that of its simple accelerated variant as

$$\begin{aligned}\boldsymbol{y}^{k+1} &= \boldsymbol{x}^k + \alpha_k(\boldsymbol{x}^k - \boldsymbol{x}^{k-1}),\\ \boldsymbol{x}^{k+1} &= \mathrm{prox}_{\lambda g}(\boldsymbol{y}^{k+1} - \gamma^k \nabla f(\boldsymbol{y}^{k+1})).\end{aligned}$$

Complexity per iteration: The cost per iteration of proximal gradient methods is dominated by the cost of computing the proximal operator. This operator has several nice properties which are helpful for computing it, and for establishing convergence of proximal methods, see e.g., [CW05]. We review here one particular property, known as the *Moreau decomposition*, which relates the proximal operator of g to the one associated with its Fenchel conjugate g^*:

$$\boldsymbol{u} = \mathrm{prox}_g(\boldsymbol{u}) + \mathrm{prox}_{g^*}(\boldsymbol{u}) \tag{1.17}$$

This relation allows us to efficiently compute both proximal operators, whenever one of them is efficiently computable. In the special case where $g = \|\cdot\|$ is a norm, then $g^* = \iota_{\|\cdot\|_* \leq 1}$, and Eq. (1.17) reduces to

$$\boldsymbol{u} = \mathrm{prox}_{\|\cdot\|}(\boldsymbol{u}) + \mathrm{proj}_{\|\cdot\|_* \leq 1}(\boldsymbol{u}). \tag{1.18}$$

For some choices of structured sparsity-inducing norms, the proximal operator can be computed efficiently. We present now some of these examples:

(i) **ℓ_1-norm:** $\mathrm{prox}_{\lambda\|\cdot\|_1}$ has a closed form solution, given by $\mathrm{prox}_{\lambda\|\cdot\|_1}(\boldsymbol{u}) = \mathrm{sign}(\boldsymbol{u}) \circ \max\{|\boldsymbol{u}| - \lambda, 0\}$. This is the soft-thresholding operator we presented in Section (1.3.1). The proximal gradient method in this case is often called ISTA (iterative shrinkage-thresholding algorithm), and its accelerated variant FISTA (fast ISTA) [BT09a].

(ii) **Submodular norms:** If g is the norm obtained from the convex envelope Θ_∞ or the homogeneous convex envelope Ω_p of a monotone submodular function (see Sections 1.3.3 and 1.3.4)[11], then its proximal operator can be computed by a *decomposition algorithm*[OB16, Section 6.3]. As in the decomposition algorithm used to compute the norm itself, this algorithm requires solving a sequence of at most d submodular minimization problems.

(iii) **Block ℓ_1/ℓ_p-norm:** If $\mathfrak{G}$ is a partition of V, the proximal operator of $\Omega_p^\cap$, defined in Eq. (1.7), can be computed separately on each group; $[\mathrm{prox}_{\lambda\Omega_p^\cap}(\boldsymbol{u})]_G = \mathrm{prox}_{\lambda d_G\|\cdot\|_p}(\boldsymbol{u}_G), \forall G \in \mathfrak{G}$. For $p = 2$, we obtain the group soft-thresholding operator $[\mathrm{prox}_{\lambda\Omega_2^\cap}(\boldsymbol{u})]_G = \mathrm{sign}(\boldsymbol{u}_G) \circ \max\{\|\boldsymbol{u}_G\|_2 - \lambda d_G, 0\}$. For $p = \infty$, the relation in Eq. (1.18) can be exploited to compute $[\mathrm{prox}_{\lambda\Omega_\infty^\cap}(\boldsymbol{u})]_G$ as the residual of the Euclidean projection on the ℓ_1-ball, which can be done in linear time [DSSSC08].

(iv) **Hierarchical ℓ_1/ℓ_p-norm:** If the groups in $\mathfrak{G}$ are tree-structured, in the sense that every pair of groups are either disjoint or one is contained in the other, [JMOB11] provided efficient algorithms to compute $\mathrm{prox}_{\lambda\Omega_p^\cap}(\boldsymbol{u})$ for $p = 2, \infty$. The algorithm consists of sequentially projecting $\boldsymbol{u}_G$ on the dual ball, given a particular ordering of the groups. The resulting complexity is $O(d)$ for $p = 2$, and $O(dh)$ where h is the height of the tree. For example, this applies to the rooted connected tree model presented in Section 1.3.1, where the descendant groups $\mathfrak{G}_H$ are indeed tree-structured.

(v) **Overlapping ℓ_1/ℓ_p-norm:** If the groups in $\mathfrak{G}$ are overlapping and not-tree structured, $\mathrm{prox}_{\lambda\Omega_p^\cap}(\boldsymbol{u})$ becomes more difficult. It can still be solved efficiently in the special case of $p = \infty$, where $\mathrm{prox}_{\lambda\Omega_\infty^\cap}(\boldsymbol{u})$ is the dual to a quadratic min-cost flow problem, as shown in [MJBO10]. Alternatively, the decomposition algorithm can be used, since ℓ_1/ℓ_∞-norm is a submodular norm (see Section 1.3.4). However, in the general case where $p \in (1, \infty)$, no efficient algorithm to compute $\mathrm{prox}_{\lambda\Omega_p^\cap}$ is known.

(vi) **Latent group Lasso norm:** If $\mathfrak{G}$ is a partition of V, recall that $\Omega_p^\cup$, defined in Eq. 1.8, is equal to the ℓ_1/ℓ_p-norm, hence its proximal operator can be easily computed in this case.

[11] Recall that the two envelopes are equal for $p = \infty$; $\Theta_\infty = \Omega_\infty = f_L(|\cdot|)$.

However, in the overlapping-groups case, computing $\mathrm{prox}_{\lambda\Omega_p^{\cup}}$ is challenging. A simple workaround is to duplicate the variables that belong to overlaps between the groups, which reduces this case back to the partition case. However, such approach is costly when the groups have substantial overlap. Another approach to compute $\mathrm{prox}_{\lambda\Omega_p^{\cup}}$ approximately was proposed in [VRMV14]. It exploits the relation in Eq. (1.18) to reduce $\mathrm{prox}_{\lambda\Omega_p^{\cup}}$ to a projection over the intersection of ℓ_p-balls associated with each group. A preprocessing step is proposed to restrict the projection to only "active" groups. The resulting projection can be computed in general via the cyclic projection algorithm of [BD86], which is guaranteed to converge, but with no convergence rate guarantees. In the case $p = 2$, the projection can be computed by solving a dual problem based on Bertsekas' projected Newton method [Ber82], which is only guaranteed to converge under a strong regularity condition on the Hessian of the objective near the optimal solution. Hence, an efficient method to solve $\mathrm{prox}_{\lambda\Omega_p^{\cup}}$, exactly or approximately, remains an open problem.

For further examples, we refer the reader to [Bac10b] and [OB16]. Moreover, efficient implementations of the proximal solvers of the ℓ_1/ℓ_p-norm, in [MJBO10] and [JMOB11], are available in the open source toolbox SPAMS (SPArse Modeling Software) [12].

Convergence rate: Proximal gradient method converges with a rate of $O(L_2 R_2^2/k)$, when $\gamma^k = 1/L_2$, where $R_2 = \|\boldsymbol{x}^0 - \boldsymbol{x}^\star\|_2$, and $\boldsymbol{x}^\star$ is any minimizer of problem (1.2). While, accelerated proximal gradient method converges, in objective value, with a rate of $O(L_2 R_2^2/k^2)$. This rate was shown to be optimal, in terms of dependence on k, for first-order methods in [NYD83]. It is worth noting that unlike basic gradient methods, accelerated methods are not descent algorithms, i.e., the objective function does not necessarily decrease at each iteration.

Furthermore, both basic and accelerated proximal gradient methods adapt to the strong convexity of the objective, achieving a linear convergence of $O(R^2 \exp(-\frac{\mu}{L}k))$ for the basic variant, and $O(R^2 \exp(-\sqrt{\frac{\mu}{L}}k))$ for the accelerated variant, when $\mu > 0$. These rates were also shown to hold if the proximal operator is only solved approximately, as long as the approximation error decreases, at each iteration k, with a fast enough rate [SRB11, VSBV13, LMH15, LZ16].

Complexity in non-Euclidean spaces: Our discussion so far has focused on the classical *Euclidean* proximal gradient methods. However, the choice of the ℓ_2-norm in these methods, may lead to suboptimal convergence, in terms of dimension dependence, for problems which are not "well-behaved" in the ℓ_2-norm. For instance, consider the case where $g = \iota_{\|\cdot\|_\infty \leq 1}$ and f is such that both $L_2 \leq 1$ and $L_\infty \leq 1$, where L_∞ is the Lipschitz constant of f with respect to the ℓ_∞-norm. A similar setting occurs for example in maxflow problems [KLOS14]. Euclidean proximal gradient method converges with a rate of $O(d/k)$, in this case. While, if we were to measure L and R with respect to the ℓ_∞-norm, the corresponding rate would be $O(1/k)$.

Such observation motivated extensions of gradient methods to non-Euclidean methods able to

[12] http://spams-devel.gforge.inria.fr/

adapt to the geometry of the problem. In particular, extensions of proximal gradient method and its accelerated variant, where the ℓ_2-norm in Eq. (1.14) is replaced by a *Bregman divergence*, were considered for example in [Tse08, Lan12]. However, as Bregman divergences are required to be strongly convex in the underlying norm, they also can introduce unnecessary dimension dependence terms in the convergence rate.

Another extension of proximal gradient method, where the ℓ_2-norm in Eq. (1.14) is replaced by an arbitrary norm $\|\cdot\|$, was also considered in [RT14], and for the constrained case in [Nes05, AZO14]. We refer to this extension as *generalized proximal gradient method* (GPM). GPM was proved to converge in $O(LR/k)$, where both L and R are measured with respect to $\|\cdot\|$ [RT14]. However, the resulting proximal operator to be solved at each iteration in GPM, is not well-studied outside the Euclidean setting. Indeed, for non-Euclidean norms, it was only shown to be tractable in the special case where g is the indicator function of the simplex, and $\|\cdot\|$ is the ℓ_1-norm [Nes05]. We fill this gap in Chapter 4, where we provide an efficient scheme to compute the non-Euclidean proximal operator, for a broad class of regularizers and norms, including examples where the Euclidean proximal operator is not known to be efficiently computable. We also introduce an accelerated variant of GPM, with a small extra computational cost.

Solution's sparsity: The iterates of Euclidean proximal gradient method and its accelerated variant do not admit sparse representations. In Chapter 4, we show that for some choices of regularizers and norms, the iterates in GPM correspond to a *sparse* convex combination of only few atoms. Unfortunately, this property is lost with acceleration.

1.4.2 Conditional gradient methods

The Frank-Wolfe algorithm was first introduced in [FW56]. It has recently experienced a remarkable revival, particularly in the context of sparse optimization and machine learning, due to its relatively cheap and sparse updates.

FW is a *projection-free* method for optimizing convex objectives over a compact convex domain, i.e., for solving the special case of problem (1.2), where g is an indicator function; $g(\boldsymbol{x}) = \iota_C(\boldsymbol{x})$, for some compact convex set $C \subseteq \mathbb{R}^d$. It was also extended to handle any closed convex function g, with a *bounded* domain, see, e.g., [BLM09, Nes15, HJN15, YZS17]. We refer to this generalized version as *generalized condition gradient* (GCG) method. Furthermore, variants of GCG able to handle special cases of functions g with unbounded domain, were presented in [HJN15] for norm regularizers, and in [YZS17] for gauge regularizers.

Algorithm: The main idea of conditional gradient methods is to minimize at each iteration k, a linear approximation of f, given by its supporting hyperplane at the current iterate $\boldsymbol{x}^k$, while leaving g intact. By convexity, the supporting hyperplane of f at any point $\boldsymbol{y} \in \mathbb{R}^d$, is given by:

$$f(\boldsymbol{x}) \geq f(\boldsymbol{y}) + \langle \nabla f(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y} \rangle. \tag{1.19}$$

The following problem is then solved at each iteration k:

$$\hat{\boldsymbol{s}} \in \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} \langle \nabla f(\boldsymbol{x}^k), \boldsymbol{x}\rangle + \lambda g(\boldsymbol{x}) \tag{1.20}$$

In FW and GCG, the next iterate is updated by taking a convex combination of the previous iterate and the new point; $\boldsymbol{x}^{k+1} = (1-\gamma)\boldsymbol{x}^k + \gamma\hat{\boldsymbol{s}}$, where γ is either chosen by line search $\gamma \in \arg\min_{\gamma\in[0,1]} f(\boldsymbol{x}^k + \gamma(\hat{\boldsymbol{s}} - \boldsymbol{x}^k))$, or set to $\gamma = \frac{2}{k+2}$.

Note that problem (1.20) corresponds to computing a subgradient of the Fenchel conjugate of g, i.e., $\hat{\boldsymbol{s}} \in \partial g^*(-\nabla f(\boldsymbol{x}^k))$. We will thus refer to this operation as the Fenchel conjugate operator of g. In the constrained variant, where g is an indicator function, this operator reduces to the *linear minimization oracle* (LMO) associated with C; $\hat{\boldsymbol{s}} \in \mathrm{LMO}_C(\boldsymbol{z}) := \arg\min_{\boldsymbol{x}\in C}\langle \boldsymbol{z}, \boldsymbol{x}\rangle$.

Complexity per iteration: The cost per iteration of conditional gradient methods is dominated by the cost of computing the Fenchel conjugate operator or the LMO. These operators are usually much cheaper than proximal operators. Indeed, in several problems, both within structured sparsity (e.g., latent group Lasso, exclusive Lasso, and general overlapping group Lasso (for $p \neq \infty$)), and beyond it in applications such as structured SVM, sparse PCA, variational inference, and submodular minimization, computing the proximal/projection operator is expensive or even intractable, while the LMO can be efficiently implemented. We provide below two examples of structured sparsity-inducing norms whose LMO admits a closed form solution:

(i) ℓ_1**-ball:** $\mathrm{LMO}_{\|\boldsymbol{x}\|_1\leq 1}(\boldsymbol{z}) = -\operatorname{sign}(\boldsymbol{z}_{i_{\max}})\mathbb{1}_{i_{\max}}$, where $i_{\max} \in \arg\max_i |z_i|$.

(ii) **Latent group Lasso ball:** $\mathrm{LMO}_{\Omega_p^{\cup}(\boldsymbol{x})\leq 1}(\boldsymbol{z}) = -d_{G_{\max}}^{-1}\operatorname{sign}(\boldsymbol{z}_{G_{\max}}) \circ \left(\frac{\boldsymbol{z}_{G_{\max}}}{\|\boldsymbol{z}_{G_{\max}}\|_q}\right)^{q-1}$, where $G_{\max} \in \arg\max_{G\in\mathfrak{G}} \|\boldsymbol{z}_G\|_q$.

Furthermore, in Chapter 2, we show that the LMO, and Fenchel conjugate operator, of several other examples of structured sparsity-inducing penalties (including latent group Lasso, exclusive Lasso, and general overlapping group Lasso) can be efficiently computed via linear programming. For other interesting examples, we refer the reader to [Jag13].

Convergence rate: Convergence results for conditional gradient methods in the literature are expressed using a variety of "constants"[13], characterizing the properties f and the domain of g. To simplify our presentation, we omit these constants here.

CGD converges in objective value with a rate of $O(1/k)$ [Cla10, Jag13, Nes15, YZS17]. This convergence rate still holds if the LMO is only solved approximately [Jag13]. Moreover, this rate is tight in general.

Indeed, [CC68] and [Wol70] showed that, even when f is strongly convex, classical FW converges at the slow rate of $\Omega(1/k^{1+\delta})$, for any $\delta > 0$, if the optimal solution $\boldsymbol{x}^\star$ lies at the the boundary of

[13] These constants can be dimension dependent.

the feasible set $\mathcal{C}$. Over the past years, significant effort was made towards investigating whether projection-free methods with convergence rates matching that of accelerated projected gradient descent exists. Table 1.1 provides a summary of such results.

In the case where f is strongly convex, [GM86] showed that FW converges linearly, if the feasible set is a polytope, and the optimal solution is in its interior. When the location of $\boldsymbol{x}^\star$ is unrestricted, [GH15] obtains a $O(1/k^2)$ rate, for strongly convex sets, while [GH13] obtains a linear rate, for polyhedral feasible sets, using a variant of LMO, called local LMO, where $\boldsymbol{x}$ is restricted in an ℓ_2-ball, i.e., $\hat{\boldsymbol{s}} \in \arg\min_{\boldsymbol{x}\in C, \|\boldsymbol{x}-\boldsymbol{x}^k\|_2 \leq r_t} \langle \nabla f(\boldsymbol{x}^k), \boldsymbol{x}\rangle$ for some $r_t > 0$.

In the more general case, where $f(\boldsymbol{x}) = g(\boldsymbol{A}\boldsymbol{x}) + \langle \boldsymbol{b}, \boldsymbol{x}\rangle$, for some $\boldsymbol{A} \in \mathbb{R}^{n\times d}, \boldsymbol{b} \in \mathbb{R}^d$, and where g is strongly convex, [LJJ15, BS17] showed that a variant of FW, called *away-step* FW, enjoys *linear convergence*, if the feasible set is a polytope. Away-step FW chooses at each iteration, to either take the regular FW direction, or an away direction, which remove weights from currently active 'bad' atoms. The direction which leads to better progress is chosen. The classical variant of FW also converges linearly, for a special case of this class of functions, namely if f is the least squares loss, even when the feasible set if not a polytope [BT04].

In the general non-strongly convex case, [LZ16] presented a variant of FW able to achieve an accelerated rate in terms of the number of gradient evaluations needed, but not in terms of LMO calls. Accelerated variants of FW achieving a faster rate $\tilde{O}(1/k^2)$,[14] in both gradient and LMO calls, for polyhedral feasible sets, can be easily obtained by combining generic acceleration schemes such as [LMH15, LZ16] with the *linearly convergent* variants of FW such as [LJJ15, GH13]. However, the constants involved in the resulting convergence rate are dimension-dependent, and in practice we observe that this acceleration does not seem useful. A more careful study of the performance of such variants is thus needed.

For the penalized variant of problem (1.2), the convergence of GCG is less well studied. To the best of our knowledge, the only convergence result which improves on the $O(1/k)$, is provided by [Nes15], which shows the GCG converges with a $O(1/k^2)$ rate, if g is strongly convex and has a bounded domain.

Complexity in non-Euclidean spaces: Note that the iterates of FW and GCD, and accordingly their performance, do not depend on the choice of the norm used to measure smoothness and strong convexity. Hence, convergence guarantees of these methods can often be obtained with respect to arbitrary norms. For example, for the constrained variant of problem (1.2), the iterates of FW were shown in [Jag13] to satisfy $f(\boldsymbol{x}^k) - f(\boldsymbol{x}^\star) \leq O(\frac{C_f}{k+2})$, where C_f is a *curvature* constant, characterizing the "non-linearity" of f, defined as

$$C_f = \sup_{\gamma\in[0,1]} \{\frac{2}{\gamma^2}\left(f(\boldsymbol{x}) - f(\boldsymbol{y}) + \langle \nabla f(\boldsymbol{y}), \boldsymbol{x}-\boldsymbol{y}\rangle\right) : \boldsymbol{y}, \boldsymbol{s} \in C, \boldsymbol{x} = \boldsymbol{y} + \gamma(\boldsymbol{s}-\boldsymbol{y})\}.$$

It is easy to see that $C_f \leq LD^2$, where $D = \max_{\boldsymbol{x},\boldsymbol{y}\in C} \|\boldsymbol{x}-\boldsymbol{y}\|$, with both L and D measured

[14] The $\tilde{O}$ notation hides logarithmic terms.

Ref.	Oracle	Feasible Set	Objective Fct.	Location of $x^\star$	Rate
[Jag13]	LMO	convex	convex	unrestricted	$O(1/k)$
[GM86]	LMO	polytope	strongly convex	interior	$O(\exp(-k))$
[BT04]	LMO	convex	$f(\boldsymbol{x}) = \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2$	interior	$O(\exp(-k))$
[GH13]	Local LMO	polytope	strongly convex	unrestricted	$O(\exp(-k))$
[GH15]	LMO	strongly convex	strongly convex	unrestricted	$O(1/k^2)$
[LZ16]	LMO	convex	convex	unrestricted	$O(1/k)$ LMO $O(1/k^2)$ gradients
[LJJ15, BS17]	LMO + Away Step	polytope	$f(\boldsymbol{x}) = g(\boldsymbol{A}\boldsymbol{x}) + \langle \boldsymbol{b}, \boldsymbol{x} \rangle$ g strongly convex	unrestricted	$O(\exp(-k))$
[LMH15, LZ16] + [LJJ15, GH13]	LMO + Away Step	polytope	convex	unrestricted	$\tilde{O}(1/k^2)$

Table 1.1: Summary of FW convergence rate results.

with respect to any norm. Note though that not all the variants of FW discussed above share this property. For example, the variant in [GH13] uses the ℓ_2-norm in its local LMO. Similarly, the accelerated variants based on [LMH15, LZ16] also implicitly use the ℓ_2-norm.

Solution's sparsity: One of the main attractive properties of FW is the sparsity of its iterates. At each iteration k, $\boldsymbol{x}^k$ corresponds to a *sparse* convex combination of k "atoms" in C. To see this, let $C = \mathrm{conv}(\mathcal{A})$ where $\mathcal{A}$ is an atomic set, then the LMO returns an atom, since $\mathrm{LMO}_C(\boldsymbol{z}) = \arg\min_{\boldsymbol{x} \in A} \langle \boldsymbol{z}, \boldsymbol{x} \rangle$.

Conclusion: The discussion in Sections 1.4.1 and 1.4.2 can be summarized as follows: Conditional gradient methods enjoy cheap and sparse iterates, their performance is independent of the choice of the norm used to measure properties of f, but they suffer a relatively slow convergence rate in general, and are not able to handle general regularizers with unbounded domain. On the other hand, (accelerated) proximal gradient methods enjoy *optimal* convergence rates, in terms of iteration-dependence, but require more expensive and dense iterates. Their performance does depend on the choice of the norm used to measure properties of f, where a careful choice can avoid unnecessary dimension-dependence. Developing an algorithm which can achieve the best of both worlds remains an *open question* of great theoretical and practical interest.

1.5 Non-convex approaches

In this section, we briefly review non-convex approaches to structured sparsity, which directly address the structured sparsity learning problem (1.1), without relying on convex relaxations.

Existing non-convex approaches have mostly focused on constrained formulations of Problem (1.1), where F is an indicator function over a set of allowed supports $\mathcal{M}$; $F(S) = \iota_{S \in \mathcal{M}}(S)$. The resulting problem is then of the form

$$\min_{\boldsymbol{x} \in \mathbb{R}^d} \{ f(\boldsymbol{x}) : \mathrm{supp}(\boldsymbol{x}) \in \mathcal{M} \} \tag{1.21}$$

Several methods have been developed to provide approximate solutions to this NP-Hard problem. These methods typically belong to two categories; greedy algorithms and discrete projected gradient descent methods. Theoretical recovery guarantees for these methods were also provided, under some conditions on f, such as restricted isometry property (RIP), both for the simple sparsity model [Tro04], and for general structured sparsity models [HZM11, BCDH10].

1.5.1 Greedy algorithms

Most existing greedy sparse-recovery algorithms are variants of orthogonal matching pursuit (OMP) [MZ93, TG07] and forward selection methods [Mil02, Wei05].

The main idea of these iterative greedy algorithms is to add variables, that maximize progress locally, to the support of the current estimate $\boldsymbol{x}^k$, starting from $\boldsymbol{x}^0 = 0$, until the target sparsity or model complexity s is reached. In particular, when $\mathcal{M}$ is the set of s-sparse supports, i.e., $\mathcal{M} = \{S : |S| \leq s\}$, and f is any loss function, forward selection[15] algorithm adds, at each iteration k, the variable which reduces the objective the most:

$$\begin{aligned} i_k &\in \operatorname*{arg\,min}_{i \in S_k^c} \min_{\operatorname{supp}(\boldsymbol{x}) \subseteq S_k} f(\boldsymbol{x}) \\ S_{k+1} &= S_k \cup \{i_k\} \end{aligned} \tag{1.22}$$

Let $\hat{S}$ be the returned support, then the estimate solution is given by $\hat{x} \in \arg\min_{\operatorname{supp}(\boldsymbol{x}) \subseteq \hat{S}} f(\boldsymbol{x})$.

The greedy selection criterion (1.22) can be implemented efficiently when f is the least squares loss, but in general it can be computationally expensive. Another possible choice is to select the variable which is most correlated with the residual. OMP uses such criterion, and executes the following steps at each iteration k:

$$\begin{aligned} i_k &\in \operatorname*{arg\,max}_{i \in V} |\langle \mathbb{1}_i, \nabla f(\boldsymbol{x}^k) \rangle| \\ S_{k+1} &= S_k \cup \{i_k\} \\ \boldsymbol{x}^{k+1} &\in \operatorname*{arg\,min}_{\operatorname{supp}(\boldsymbol{x}) \subseteq S_{k+1}} f(\boldsymbol{x}) \end{aligned} \tag{1.23}$$

For example, if $f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2$, we have $i_k \in \arg\max_{i \in V} |\langle \boldsymbol{a}_i, \boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}^k \rangle|$, where $\boldsymbol{a}_i$ is the ith column of $\boldsymbol{A}$, and $\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}^k$ is a residual term. OMP is then computationally more efficient than forward selection, but satisfies a weaker optimality guarantee (see, e.g., [EKDN16]).

An interesting connection between these greedy algorithms and submodular maximization was established in [DK11, EKDN16], where it is shown that the function $R(S) = -\min_{\operatorname{supp}(\boldsymbol{x}) \subseteq S} f(\boldsymbol{x})$ satisfies a weak notion of submodularity.

A popular variant of OMP worth mentioning is the compressive sampling matching pursuit

[15] This is sometimes also called OMP in the literature.

(CoSaMP) algorithm[NT09]. In this variant, the s largest elements of the gradient are added to the support, instead of only the largest one, and the estimation step (1.23) is followed by a projection step, where we retain only the s largest entries of $\boldsymbol{x}^{k+1}$.

Extensions of these matching pursuit algorithms to general structured sparsity models were also developed. For instance, [HZM11] presented a generalization of OMP to group sparse structures, where $\mathcal{M}$ is the set of supports which can be covered by the union of s-groups, i.e., $\mathcal{M} = \{S : F(S) \leq s\}$, where $F(S) = \min_{\boldsymbol{\omega} \in \{0,1\}^{|\mathfrak{G}|}} \{\sum_{G \in \mathfrak{G}} d_G \omega_G : \sum_{G \in \mathfrak{G}} \omega_G \mathbb{1}_G \geq \mathbb{1}_S\}$ is the minimal weighted set cover function, and $\mathfrak{G}$ is a given collection of groups. In this variant, each iteration selects a group G to add to the current support instead of a single index.

Similarly, [BCDH10] presented a generalization of CoSaMP to general structured sparsity models, described as union of s-dimensional spaces, i.e., $\mathcal{M} = \{S : |\operatorname{supp}(S)| \leq s, \operatorname{supp}(S) \subseteq G \text{ for some } G \in \mathfrak{G}\}$. In this variant, the projection of the gradient and the estimate on the set of s-sparse supports, is replaced by a general projection step over $\mathcal{M}$:

$$\operatorname{proj}_{\mathcal{M}}(\boldsymbol{u}) \in \underset{\operatorname{supp}(\boldsymbol{x}) \in \mathcal{M}}{\arg\min} \|\boldsymbol{x} - \boldsymbol{u}\|_2. \tag{1.24}$$

When $\mathcal{M} = \{S : |S| \leq s\}$, $\operatorname{proj}_{\mathcal{M}}(\boldsymbol{u})$ returns the largest s-entries of $\boldsymbol{u}$, and is known as the *hard-thresholding* operator. In general though, this projection is NP-Hard, which motivated the development of extensions of CoSaMP, allowing for approximate projections [DNW13, HIS15a].

1.5.2 Discrete projected gradient descent method

Another popular non-convex method used to solve problem (1.21) is the discrete projected gradient descent method. As in the convex variant of this algorithm (see Section 1.4.1), the iterates are given by

$$\boldsymbol{x}^{k+1} \in \operatorname{proj}_{\mathcal{M}}(\boldsymbol{x}^k - \gamma \nabla f(\boldsymbol{x}^k)), \tag{1.25}$$

where the gradient of f is assumed to be L-Lipschitz continuous, and the step size $\gamma \in (0, 1/L]$.

When $\mathcal{M}$ is the set of s-sparse supports, this method is known as the iterative hard thresholding (IHT) method [HGT06]. The generalization of IHT, given by (1.25), was studied in [BCDH10], for general structured sparsity models described as union of s-dimensional spaces. Furthermore, variants allowing for inexact projections were proposed in [GE13, HIS15a, JRD16].

We conclude the discussion in this section with some examples of structured sparsity models $\mathcal{M} = \{S : F(S) \leq s\}$, which admit exact or approximate discrete projections:

(i) **Sparse model:** When $F(S) = |S|$, as discussed above, $\operatorname{proj}_{\mathcal{M}}(\boldsymbol{u})$ is the hard-thresholding operator, which returns the largest (in absolute value) s-entries of $\boldsymbol{u}$.

(ii) **Tree model:** If $\mathcal{M}$ is the set of s-sparse rooted connect tree supports, then $\operatorname{proj}_{\mathcal{M}}$ can be

computed exactly via dynamic programming [CT13, BBC$^+$16].

(iii) **Dispersive model:** If $\mathcal{M}$ is the set of s-sparse supports, with a minimum distance $\Delta > 0$ between any two non-zero, i.e., $\mathcal{M} = \{S : |S| \leq s, |i-j| > \Delta, \forall i \neq j \in S\}$, then $\mathrm{proj}_{\mathcal{M}}$ can be computed exactly via linear programming [HDC09].

(iv) **Minimal set cover model:** When $F(S) = \min_{\boldsymbol{\omega} \in \{0,1\}^{|\mathfrak{G}|}} \{\sum_{G \in \mathfrak{G}} d_G \omega_G : \sum_{G \in \mathfrak{G}} \omega_G \mathbb{1}_G \geq \mathbb{1}_S\}$, the projection results in a weighted maximum coverage problem, which is NP-Hard in general, but can be approximated by a greedy algorithm achieving a $(1 - 1/e)$-approximation ratio [NWF78]. This was used in [JRD16], which showed that near optimal solutions to problem (1.21) can still obtained with this approximate greedy projection. Moreover, [BBC$^+$16] showed that for certain group structures $\mathfrak{G}$, where the groups have acyclic interactions, the projection can be computed exactly via dynamic programming.

(v) **Graph model:** [HIS15b] introduced a weighted graph model, where given a graph $\mathcal{G} = (V, E)$, $\mathcal{M} = \{S : |S| \leq s, \gamma(F_S) \leq g, w(F_S) \leq b\}$, where $\gamma(F_S)$ is the number of connected components formed by the forest F_S corresponding to S and $w(F_S)$ is the total weight of edges in the forest F_S. The projection in this case is also NP-Hard, but it can be approximated by a nearly-linear time algorithm provided in [HIS15b]. This approximate projection can be used in the framework of [HIS15a] to provide near optimal solutions to problem (1.21).

1.6 Overview of contributions

This thesis presents novel methods for incorporating general structured sparsity models in learning problems, which balance computational and statistical efficiency trade-offs.

The first part of the thesis (Chapters 2 and 3) focuses on how to turn discrete descriptions of structures to convex ones, amenable to efficient optimization, without loosing too much in the process. The second part (Chapter 4) is concerned with how to efficiently optimize the resulting convex problems. In the third part of the thesis (Chapters 5 and 6), we return to the discrete descriptions and try to address them directly.

In particular, the contributions made by each chapter are as follows:

- The goal of Chapter 2 is to compute tractable tight convex relaxations for a large range of sparsity structures. Prior to our work, this was only feasible for structures described by monotone submodular functions. This chapter introduces a new class of structured sparsity penalties, called “ILP penalties", which are set functions that can be expressed by linear programs with integral solutions. By borrowing tools from integer programming, we demonstrate that the convex envelope of functions in this class can be computed and optimized efficiently. We also show that many important heuristically designed penalties in the literature fall within our framework, and propose other new interesting penalties. We

also illustrate on concrete examples how the common practice of imposing homogeneity on convex penalties leads to an unnecessary loss of structure.

- The goal of Chapter 3 is to study which structure is necessarily lost by relaxing general ℓ_p-regularized combinatorial functions to continuous convex penalties, and which is preserved. This was previously studied only for the *homogeneous* convex envelope. Motivated by the observation made in Chapter 2, we consider instead the *non-homogeneous* convex envelope, and set out to present a rigorous comparison of the two relaxations. We thus study their geometric properties and present statistical conditions under which consistent estimation and model selection is possible under each relaxation. This chapter demonstrates that the *non-homogeneous* convex envelope is *tighter*, and is able to preserve, both in a geometric and a statistical sense, a larger class of combinatorial structures than its homogeneous counterpart; specifically it can geometrically preserve any *monotone* structure. This translates to better support recovery performance, as we characterize statistically and observe empirically.

- The goal of Chapter 4 is to develop faster convex optimization methods, to solve problems regularized with convex structured sparsity-inducing penalties. In particular, we consider a generalized extension of the proximal gradient method (GPM). Unlike classical proximal gradient methods, this method does not operate in the Euclidean space, but employs instead other general norms. This flexibility allows this method to adapt to the geometry of the given problem leading to significant improvements in computational complexity. Prior to our work, this method was only known to be feasible when the convex regularizer is the simplex constraint, and the chosen norm is the ℓ_1-norm. This chapter proposes a novel tractable scheme to compute the iterates required by GPM for any *polyhedral* regularizer and norm, thus establishing the tractability of GPM for this broad class of functions. We also introduce an accelerated variant of GPM, with a small extra computational cost. Furthermore, for a special class of regularizers, namely for atomic norms with linearly independent atoms, and a matching norm, we devise an efficient greedy algorithm that computes the iterates of GPM, in almost the same cost as Frank-Wolfe iterates. The resulting iterates also correspond to a sparse convex combination of few atoms. We illustrate on important examples of structured sparsity-inducing norms, Lasso and Latent Group Lasso, that our results offer significant speed-up over state-of-the-art methods.

- The goal of Chapter 5 is to handle structures for which the convex approach fails. An important implication of the results in Chapter 3 is that non-monotone structures do not admit tight convex relaxations. Furthermore, such relaxations are in general intractable, for (even monotone) structures, outside the class of submodular and ILP penalties. To address such cases, this chapter proposes to use a discrete proximal gradient descent method, efficient for several classes of structures, including submodular, supermodular and ILP penalties. We demonstrate numerically superior performance over alternative heuristic convex methods.

- The goal of Chapter 6 is to handle structured "sparse" signals where the small coeffi-

cients are not exactly zero. To this end, we propose a probabilistic mixture model with combinatorial priors. This chapter adapts the non-convex method introduced in Chapter 5 to approximate the corresponding non-convex maximum a posteriori estimate. The resulting algorithm is again efficient for several classes of structures, including submodular, supermodular and ILP penalties. We demonstrate numerically that our proposed approach performs better than alternative convex methods for non-truly sparse signals.

2 Convex Relaxations via Linear Programming

2.1 Introduction

In this chapter, we adopt the convex approach to structured sparse learning. As discussed in the introduction chapter, the challenge in this approach resides in finding a computationally tractable convex surrogate that tightly captures the desired structured sparsity model. We follow the systematic approach adopted in [Bac10a], where the convex surrogate is chosen to be the *tightest* convex relaxation, i.e., the *convex envelope*, of a combinatorial penalty expressing the desired structure. However, we recall that evaluating and optimizing such convex surrogates is in general *intractable*. In this chapter, we are thus interested in addressing the following question:

> Which set functions, able to naturally express structured sparsity models, admit *tractable convex envelopes*, which can be efficiently optimized?

2.1.1 Related work

The above question was partially answered in [Bac10a], which showed that the convex envelope of *monotone submodular functions* can be efficiently computed and optimized (see Section 1.3.3). Though a natural model for a number of useful structures, submodular functions are unable to capture all structures encountered in practice. On the other hand, [OB12] considered general ℓ_p-regularized set functions and provided formulations for their *homogeneous* convex envelopes. These formulations can be computed and optimized efficiently for submodular functions and a limited number of other special cases, but are still intractable in general (see Section 1.3.4). Moreover, in this chapter we observe that imposing homogeneity may cost an unnecessary loss of structure in some cases (see Section 2.5). This effect is studied in details in Chapter 3.

2.1.2 Contributions

In this chapter, we propose a framework for structured sparsity modeling based on *linear programming*, that addresses both tractability and tightness concerns of convex approaches. We

introduce a new class of set functions, namely functions that can be expressed via *a linear program (LP) with integral solutions*, which admit tractable convex envelopes. We call this class of functions ILP penalties. Our framework complements the submodular modeling approach; it allows us to derive tight convex relaxations for several submodular and non-submodular functions.

Our specific contributions are summarized as follows:

- We identify a sufficient condition for the convex envelope of a general set function to be tractable (Section 2.2).
- We show that ILP penalties in particular satisfy this sufficient condition. The resulting convex envelopes (which are not necessarily norms) take the form of linear programs, and can be evaluated and optimized efficiently (Section 2.4).
- We provide examples of functions that belong to a subclass of ILP penalties, consisting of functions that can be expressed via an LP with *totally unimodular* constraints. As a result, we recover and give new theoretical interpretations to several popular structured sparsity norms, such as the latent group Lasso, hierarchical group Lasso, and exclusive Lasso. We also define new convex penalties, that can be used to express structures encountered in practice with no known tight convex relaxation (Section 2.5). Their performance is illustrated on numerical examples in Section 2.6.

This chapter is based on the joint work with Volkan Cevher [EHC15].

2.2 Tractable convex envelopes

We consider set functions $F : 2^V \to \overline{\mathbb{R}}$, where $V = \{1, \cdots, d\}$, to encode structured sparsity models over the support of an unknown parameter vector $\boldsymbol{x} \in \mathbb{R}^d$. As in [Bac10a], we propose to use the convex envelope of $F(\mathrm{supp}(\boldsymbol{x}))$ over the unit ℓ_∞-ball as a convex surrogate for it. Recall that restricting the convex envelope of $F(\mathrm{supp}(\boldsymbol{x}))$ over a bounded domain is necessary, otherwise it would evaluate to a constant. In this chapter, we thus assume, without loss of generality, that $\|\boldsymbol{x}\|_\infty \leq 1$. It is also interesting to consider the case where the magnitude of $\boldsymbol{x}$ is penalized with a general ℓ_p-norm, as done in [OB12]. We defer this case to Chapter 3.

Recall that the convex envelope of a function is given by its biconjugate. We start by identifying a sufficient condition for the tractable computation of the biconjugate of $F(\mathrm{supp}(\boldsymbol{x}))$ over the unit ℓ_∞-ball.

Lemma 1. *Given a set function $F : 2^V \to \overline{\mathbb{R}}$, let $F_\infty(\boldsymbol{x}) = F(\mathrm{supp}(\boldsymbol{x})) + \iota_{\|\boldsymbol{x}\|_\infty \leq 1}(\boldsymbol{x})$ and denote its biconjugate by $\Theta_\infty := F_\infty^{**}$. If F admits an extension $\hat{f} : [0,1]^d \to \overline{\mathbb{R}}$, i.e., $\hat{f}(\mathbb{1}_S) = F(S), \forall S \subseteq V$, which satisfy the following assumptions:*

A1. $\hat{f}$ is a proper and lower semi-continuous (l.s.c.) convex function,

A2. $\max_{\boldsymbol{s}\in\{0,1\}^d} |\boldsymbol{z}|^\top \boldsymbol{s} - \hat{f}(\boldsymbol{s}) = \max_{\boldsymbol{s}\in[0,1]^d} |\boldsymbol{z}|^\top \boldsymbol{s} - \hat{f}(\boldsymbol{s})$, $\forall \boldsymbol{z} \in \mathbb{R}^d$,

then $\Theta_\infty(\boldsymbol{x}) = \inf_{\boldsymbol{s}\in[0,1]^d}\{\hat{f}(\boldsymbol{s}) : \boldsymbol{s} \geq |\boldsymbol{x}|\}$, *and* Θ_∞ *can be efficiently computed, if the resulting optimization problem can be efficiently solved.*

Proof. It holds that

$$\begin{aligned}
F_\infty^*(\boldsymbol{z}) &= \sup_{\|\boldsymbol{x}\|_\infty \leq 1} \boldsymbol{x}^\top \boldsymbol{z} - F(\mathrm{supp}(\boldsymbol{x})) \\
&= \sup_{\boldsymbol{s}\in\{0,1\}^d} \sup_{\substack{\|\boldsymbol{x}\|_\infty \leq 1 \\ \mathbb{1}_{\mathrm{supp}(\boldsymbol{x})} = \boldsymbol{s}}} \boldsymbol{x}^\top \boldsymbol{z} - \hat{f}(\boldsymbol{s}) && \text{(by A1)} \\
&= \max_{\boldsymbol{s}\in\{0,1\}^d} |\boldsymbol{z}|^\top \boldsymbol{s} - \hat{f}(\boldsymbol{s}) && \text{(by Hölder's inequality)} \\
&= \max_{\boldsymbol{s}\in[0,1]^d} |\boldsymbol{z}|^\top \boldsymbol{s} - \hat{f}(\boldsymbol{s}) && \text{(by A2)}
\end{aligned}$$

Assumption $A2$ guarantees that the above convex relaxation of F_∞^* is exact. Otherwise, the last equality will only hold as an upper bound.

$$\begin{aligned}
F_\infty^{**}(\boldsymbol{x}) &= \sup_{\boldsymbol{z}\in\mathbb{R}^d} \boldsymbol{x}^\top \boldsymbol{z} - F_\infty^*(\boldsymbol{z}) \\
&= \sup_{\boldsymbol{z}\in\mathbb{R}^d} \min_{\boldsymbol{s}\in[0,1]^d} \boldsymbol{z}^\top \boldsymbol{x} - |\boldsymbol{z}|^\top \boldsymbol{s} + \hat{f}(\boldsymbol{s}) \\
&\overset{\star}{=} \min_{\boldsymbol{s}\in[0,1]^d} \sup_{\substack{\boldsymbol{z}\in\mathbb{R}^d \\ \mathrm{sign}(\boldsymbol{z}) = \mathrm{sign}(\boldsymbol{x})}} |\boldsymbol{z}|^\top (|\boldsymbol{x}| - \boldsymbol{s}) + \hat{f}(\boldsymbol{s}) \\
&= \inf_{\boldsymbol{s}\in[0,1]^d} \{\hat{f}(\boldsymbol{s}) : \boldsymbol{s} \geq |\boldsymbol{x}|\}
\end{aligned}$$

where $F_\infty^{**}(\boldsymbol{x}) = +\infty$ if $\boldsymbol{x} \notin [-1,1]^d \cap \mathrm{dom}(\hat{f})$. Given assumption $A1$, $(\star)$ holds by Sion's minimax theorem [S+58, Corollary 3.3]. □

Note that the main difficulty in computing the convex envelope of F_∞ is that the Fenchel conjugate F_∞^* requires solving a combinatorial problem which in general is NP-Hard. In fact, if F_∞ has a tractable Fenchel conjugate, its envelope can be numerically approximated by a subgradient method [JSK11].

Remark 1. *It is worth noting that, without assumption $A2$, the resulting convex function in Lemma 1 will still be a convex lower bound of F_∞, albeit not necessarily the* tightest *one.*

A natural choice for the convex extension $\hat{f}$ required in Lemma 1 is the *convex closure* f^- of F, i.e., the *largest* convex lower bound of F on $[0,1]^d$ (see Appendix A.2). Indeed, the convex closure of *any* proper set function F does satisfy the two assumptions in Lemma 1 (see Proposition 26 and Corollary 12). The following proposition shows that this is actually the only possible choice of $\hat{f}$ satisfying the assumptions in Lemma 1.

Proposition 1. *Given a set function $F : 2^V \to \overline{\mathbb{R}}$, and an extension $\hat{f} : [0,1]^d \to \overline{\mathbb{R}}$ of F satisfying assumptions A1 and A2 in Lemma 1, then $\hat{f} = f^-$ is the convex closure of F, and $\Theta_\infty(\boldsymbol{x}) = \inf_{\boldsymbol{s}\in[0,1]^d}\{f^-(\boldsymbol{s}) : \boldsymbol{s} \geq |\boldsymbol{x}|\}$.*

Proof. Assume that $\hat{f} \neq f^-$, then $\exists \bar{\boldsymbol{s}} \in [0,1]^d$ s.t. $\hat{f}(\bar{\boldsymbol{s}}) < f^-(\boldsymbol{s})$, by A1 and the definition of f^- as the largest convex lower bound on F. f^- can also be equivalently defined as $f^-(\boldsymbol{s}) = \max_{\boldsymbol{\kappa}\in\mathbb{R}^d,\rho\in\mathbb{R}}\{\boldsymbol{\kappa}^\top\boldsymbol{s} + \rho : \boldsymbol{\kappa}(S) + \rho \leq F(S), \forall S \subseteq V\}$ (see Def. 20). Let $\bar{\boldsymbol{\kappa}}, \bar{\rho}$ be the corresponding maximizers for $\bar{\boldsymbol{s}}$, then $\bar{\boldsymbol{\kappa}}(S) + \bar{\rho} \leq F(S), \forall S \subseteq V$. Hence, we have

$$\begin{aligned}
\min_{\boldsymbol{s}\in[0,1]^d} \hat{f}(\boldsymbol{s}) - \bar{\boldsymbol{\kappa}}^\top\boldsymbol{s} &\leq \hat{f}(\bar{\boldsymbol{s}}) - \bar{\boldsymbol{\kappa}}^\top\bar{\boldsymbol{s}} \\
&< f^-(\bar{\boldsymbol{s}}) - \bar{\boldsymbol{\kappa}}^\top\bar{\boldsymbol{s}} \\
&= \bar{\rho} \leq \min_{\boldsymbol{s}\in\{0,1\}^d} f^-(\boldsymbol{s}) - \bar{\boldsymbol{\kappa}}^\top\boldsymbol{s} \\
&= \min_{\boldsymbol{s}\in[0,1]^d} \hat{f}(\boldsymbol{s}) - \bar{\boldsymbol{\kappa}}^\top\boldsymbol{s}, \qquad \text{(by A2)}
\end{aligned}$$

which leads to a contradiction. □

Hence, computing the convex envelope Θ_∞ reduces to computing the convex closure f^- of F. Unfortunately, computing and optimizing the convex closure itself is NP-hard in general [Von07].

However, if F is a monotone submodular function, its convex closure is given by its Lovász extension, i.e., $f^- = f_L$ (see Proposition 28), which can be efficiently computed (see Section A.1). Lemma 1 then recovers the result by [Bac10a] showing that the corresponding convex envelope is $\Theta_\infty(\boldsymbol{x}) = \inf_{\boldsymbol{s}\in[0,1]^d}\{f_L(\boldsymbol{s}) : \boldsymbol{s} \geq |\boldsymbol{x}|\} = f_L(|\boldsymbol{x}|)$. In this case, Θ_∞ is a norm. However, unlike [OB12], we do not impose homogeneity on our convex relaxation of F_∞. As a result, contrary to classical convex penalties, Θ_∞ is not necessarily a norm.

Set functions whose convex closure is efficiently computable are not limited to submodular functions. Indeed, we can naturally express various structured sparsity models via integer programs (IP), where the linear objective encourages classical sparsity penalties (e.g., simple sparsity and group sparsity), and the linear constraints enforce further structure on the support. When the constraints result in an *integral* solvable polyhedron, the corresponding model admits a tractable convex envelope. This observation is a direct corollary of properties of integer programs, which we review next.

2.3 Review of integral linear programming

Given a non-empty polyhedron $P := \{\boldsymbol{\beta} \in \mathbb{R}^m_+ : \boldsymbol{M}\boldsymbol{\beta} \leq \boldsymbol{c}\}$, where $\boldsymbol{M} \in \mathbb{R}^{l\times m}$ and $\boldsymbol{c} \in \mathbb{R}^l$, solving the IP over P given by $\max_{\boldsymbol{\beta}\in\mathbb{Z}^m}\{\boldsymbol{\theta}^\top\boldsymbol{\beta} : \boldsymbol{\beta} \in P\}$ is NP-Hard in general. It is natural to consider instead the corresponding LP-relaxation, i.e., $\max_{\boldsymbol{\beta}\in\mathbb{R}^m}\{\boldsymbol{\theta}^\top\boldsymbol{\beta} : \boldsymbol{\beta} \in P\}$, where the

integrality constraint is relaxed. We say that a polyhedron P is *solvable*, if we can solve linear programs over P efficiently[1].

In general, LP-relaxation only obtains an upper bound on the optimal value of the IP. However, in some cases the LP-relaxation is *exact*, i.e., $\max_{\boldsymbol{\beta}\in\mathbb{Z}_+^m}\{\boldsymbol{\theta}^\top\boldsymbol{\beta} : \boldsymbol{\beta}\in P\} = \max_{\boldsymbol{\beta}\in\mathbb{R}_+^m}\{\boldsymbol{\theta}^\top\boldsymbol{\beta} : \boldsymbol{\beta}\in P\}$. We call the corresponding LP *integral*.

Definition 1 (Integral linear programs[2])**.** *A linear program* $\max_{\boldsymbol{\beta}\in\mathbb{R}_+^m}\{\boldsymbol{\theta}^\top\boldsymbol{\beta} : \boldsymbol{M}\boldsymbol{\beta}\leq \boldsymbol{c}\}$ *is said to be integral if it has at least one integral optimal solution.*

Proposition 2 (Integral polyhedra [NW99, Prop. 1.3])**.** *If P is* integral*, i.e., all of its extreme points are integral, then the corresponding LP is integral for all $\boldsymbol{\theta}\in\mathbb{R}^l$ for which it has an optimal solution and vice versa.*

We present below a sufficient condition for P to be integral.

Definition 2 (Total unimodularity)**.** *A matrix $\boldsymbol{M}\in\mathbb{R}^{l\times m}$ is totally unimodular (TU) iff the determinant of every square submatrix of $\boldsymbol{M}$ is 0 or ± 1.*

Proposition 3 ([NW99, Prop. 2.2])**.** *Given a TU matrix $\boldsymbol{M}\in\mathbb{R}^{l\times m}$, the polyhedron $P := \{\boldsymbol{\beta}\in\mathbb{R}_+^m : \boldsymbol{M}\boldsymbol{\beta}\leq\boldsymbol{c}\}$ is integral for all vectors $\boldsymbol{c}\in\mathbb{Z}^l$ for which it is non-empty.*

Checking if a matrix is TU can be done via a $O((l+m)^3)$-time algorithm [Tru90]. A practical implementation of a simplified version of this algorithm, with a slower $O((l+m)^5)$-time complexity, is provided by [WT13][3]. Moreover, one can often identify if a matrix is TU by inspection. Indeed, in Section 2.5, we are able to identify if matrices in the examples we consider are TU, by exploiting certain properties of total unimodularity. We list these properties in the appendix of this chapter.

Remark 2. *A weaker sufficient condition for the integrality of P is for the system $\boldsymbol{M}\boldsymbol{\beta}\leq\boldsymbol{c}$ to be total dual integral (TDI) [GP79] and $\boldsymbol{c}$ to be integral[4]. This condition can also be verified in polynomial time [CLS84]. For more details about TDI, we refer the reader to [NW99].*

2.4 Integral linear programming penalties

We are now ready to present a simple template for our proposed structured sparsity model.

Definition 3 (ILP penalties)**.** *We define an* integral linear programming *(ILP) penalty as $F_{\mathrm{ILP}}(\mathrm{supp}(\boldsymbol{x}))$ where F_{ILP} is a set function that can be written as*

$$F_{\mathrm{ILP}}(S) := \inf_{\boldsymbol{\omega}\in\{0,1\}^M}\{\boldsymbol{d}^\top\boldsymbol{\omega} + \boldsymbol{e}^\top\boldsymbol{s} : \boldsymbol{M}\boldsymbol{\beta}\leq\boldsymbol{c}, \mathbb{1}_S = \boldsymbol{s}\}, \tag{2.1}$$

[1] P can be solvable even if m, l are exponentially large, e.g., submodular polyhedra are solvable (see Sect. A.1).

[2] *Integral* linear programs (which is an LP with at least one integral optimal solution) should be distinguished from *integer* linear programs (where variables are constrained to be integral).

[3] Software available at: https://www.utdallas.edu/~klaus/TUtest/index.html

[4] For example, the linear systems describing submodular polyhedra are TDI [Fuj05, Corollary 3.21].

where $\boldsymbol{\beta} = \begin{bmatrix} \boldsymbol{\omega} \\ \boldsymbol{s} \end{bmatrix}$ *and the polytope* $P = \{\boldsymbol{\beta} \in [0,1]^{d+M} : \boldsymbol{M}\boldsymbol{\beta} \leq \boldsymbol{c}\}$ *is* integral *and* solvable. *In the special case where* $\boldsymbol{c} \in \mathbb{Z}^l$ *and the linear system* $\boldsymbol{M}\boldsymbol{\beta} \leq \boldsymbol{c}$ *is TDI (see Remark 2), we call* F_{ILP} *a* TDI penalty*, and if in addition* $\boldsymbol{M}$ *is TU (see Definition 2), we call* F_{ILP} *a* TU penalty.

The above template offers the following parameters: $\boldsymbol{e} \in \mathbb{R}^d$ is an arbitrary weight vector encouraging (weighted) sparsity on the support, $\boldsymbol{\omega} \in \{0,1\}^M$ is useful for modeling latent variables (e.g. groups), and accordingly the weight vector $\boldsymbol{d} \in \mathbb{R}^M$ encourages (weighted) sparsity on the latent variables (e.g., group sparsity), and finally the linear constraints $\boldsymbol{M}\boldsymbol{\beta} \leq \boldsymbol{c}$, where $\boldsymbol{M} \in \mathbb{R}^{l \times (M+d)}$ and $\boldsymbol{c} \in \mathbb{R}^l$, enforce further structure on the sparsity pattern. For example, to express simple sparsity, we set $\boldsymbol{e} = \mathbb{1}$ and everything else to zero. We provide more interesting examples in Section 2.5.

From Prop. 2, it follows that proper ILP penalties satisfy the assumptions described in Lemma 1, where the convex extension is defined as $\hat{f}_{\text{ILP}}(\boldsymbol{s}) := \inf_{\boldsymbol{\omega} \in \{0,1\}^M} \{\boldsymbol{d}^\top \boldsymbol{\omega} + \boldsymbol{e}^\top \boldsymbol{s} : \boldsymbol{M}\boldsymbol{\beta} \leq \boldsymbol{c}\}, \forall \boldsymbol{s} \in [0,1]^d$, and by Prop. 1, $f^-_{\text{ILP}} = \hat{f}_{\text{ILP}}$. The resulting convex envelope is given below.

Proposition 4 (Convex envelope of ILP penalties)**.** *The convex envelope of an ILP penalty over the* ℓ_∞*-ball is given by the following LP:*

$$\Theta^{\text{ILP}}_\infty(\boldsymbol{x}) = \inf_{\boldsymbol{s} \in [0,1]^d, \boldsymbol{\omega} \in [0,1]^M} \{\boldsymbol{d}^\top \boldsymbol{\omega} + \boldsymbol{e}^\top \boldsymbol{s} : \boldsymbol{M}\boldsymbol{\beta} \leq \boldsymbol{c}, |\boldsymbol{x}| \leq \boldsymbol{s}\} \tag{2.2}$$

$\Theta^{\text{ILP}}_\infty$ often admits a closed form solution (see Section 2.5). Otherwise, it can be evaluated by solving the LP (2.2), which can be done efficiently, since P is assumed solvable in Def. 3. Furthermore, since the Fenchel conjugate-type operator of $\Theta^{\text{ILP}}_\infty$ can be computed efficiently, again via LP, then we can use the *conditional gradient method* to solve the learning problems regularized with $\Theta^{\text{ILP}}_\infty$ (see Section 1.4.2).

2.5 Examples of totally unimodular penalties

Besides allowing tractable tight convex relaxations, the choice of ILP penalties, and in particular TU penalties[5], is motivated by their ability to capture several important structures encountered in practice. In what follows, we study several TU penalties and their convex relaxations. We present a reinterpretation of several well-known norms in the literature, as well as introduce new ones.

2.5.1 Group sparsity

Group sparsity is an important class of structured sparsity models that arise naturally in machine learning and signal processing applications (see Section 1.3.1), where prior information on $\boldsymbol{x}$ dictates certain groups of variables to be selected or discarded together.

[5]We focus on TU penalties because of the relative ease of checking if this property holds. It is an interesting research question to identify what structures can be captured by more general penalties (e.g., TDI penalties).

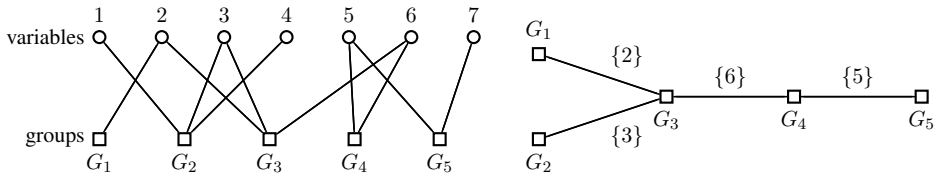


Figure 2.1: (Left) Bipartite graph representation, (Right) Intersection graph representation of the group structure $\mathfrak{G} = \{G_1 = \{2\}, G_2 = \{1,3,4\}, G_3 = \{2,3,6\}, G_4 = \{5,6\}, G_5 = \{5,7\}\}$

A group sparsity model thus features a group structure $\mathfrak{G}$, which is a collection of potentially overlapping groups $\mathfrak{G} = \{G_1, \cdots, G_M\}$ that cover the ground set V, where each group $G_i \subseteq V$ is a subset of variables. A group structure construction immediately supports two compact graph representations (see Figure 2.1).

First, we can represent $\mathfrak{G}$ as a bipartite graph $(V \cup \mathfrak{G}, E)$ [BBC$^+$16], where the groups form one set of vertices, and the variables form the other. A variable $i \in V$ is connected by an edge to a group $G_j \in \mathfrak{G}$ iff $i \in G_j$. We denote by $\boldsymbol{B} \in \{0,1\}^{d \times M}$ the biadjacency matrix of this bipartite graph; $B_{ij} = 1$ iff $i \in G_j$, and by $\boldsymbol{E} \in \{0,1\}^{|E| \times (M+d)}$ its edge-node incidence matrix; $E_{ij} = 1$ iff the vertex j is incident to the edge $e_i \in E$. Second, we can represent $\mathfrak{G}$ as an intersection graph $(\mathfrak{G}, E)$ [BBC$^+$16], where the vertices are the groups $G_i \in \mathfrak{G}$. Two groups G_i and G_j are connected by an edge iff $G_i \cap G_j \neq \emptyset$. This structure makes it explicit whether groups themselves have cyclic interactions via variables, and identifies computational difficulties.

Overlap count function

In group sparse models, we typically seek to express the support of $\boldsymbol{x}$ using only few groups. One natural penalty to consider then is the monotone submodular function that counts the weighted number of groups that are intersected by the support $F_\cap(S) = \sum_{G_i \in \mathfrak{G}, S \cap G_i \neq \emptyset} d_i$. The convex envelope of this function is the ℓ_∞-group Lasso norm (see Section 1.3.1), as shown in [Bac10a]. We now show how to express $F_\cap$ as a TU penalty.

Definition 4 (Overlap count function as TU penalty)**.** *We can rewrite $F_\cap$ as*

$$F_\cap(S) = \min_{\boldsymbol{\omega} \in \{0,1\}^M} \{\boldsymbol{d}^T \boldsymbol{\omega} : \boldsymbol{H}\boldsymbol{\beta} \leq 0, \mathbb{1}_S = \boldsymbol{s}\},$$

where $\boldsymbol{H}$ is the following matrix:

$$\boldsymbol{H} := \begin{bmatrix} -\boldsymbol{I}_M, \boldsymbol{H}_1 \\ -\boldsymbol{I}_M, \boldsymbol{H}_2 \\ \cdots \\ -\boldsymbol{I}_M, \boldsymbol{H}_d \end{bmatrix}, \ \boldsymbol{H}_k(i,j) = \begin{cases} 1 & \text{if } j = k, j \in G_i \\ 0 & \text{otherwise} \end{cases},$$

$\boldsymbol{d} \in \mathbb{R}_+^M$ are positive group weights, and $\boldsymbol{H}\boldsymbol{\beta} \leq \boldsymbol{0}$ corresponds to $s_j \leq w_i, \forall j \in G_i$.

For any coefficient in the support $S = \mathrm{supp}(\boldsymbol{x})$, the constraint $\boldsymbol{H\beta} \leq 0$ forces all the groups that contain this coefficient to be selected. $\boldsymbol{H}$ is TU, since each row of $\boldsymbol{H}$ contains at most two non-zero entries, and the entries in each row with two non-zeros sum up to zero, which is a sufficient condition for total unimodularity [NW99, Proposition 2.6]. The convex envelope of $F_{\cap}(\mathrm{supp}(\boldsymbol{x}))$ is then given by Proposition 4, which yields the group Lasso norm, as expected.

Corollary 1. *The convex envelope of $F_{\cap}(\mathrm{supp}(\boldsymbol{x}))$ over the unit ℓ_∞-ball is*

$$\Theta_\infty^{\cap}(x) = \sum_{G_i \in \mathfrak{G}} d_i \|\boldsymbol{x}_{G_i}\|_\infty,$$

for $\boldsymbol{x} \in [-1, 1]^d$, and $\Theta_\infty^{\cap}(x) = \infty$ otherwise.

Minimal set cover

The Overlap count function induces supports corresponding to the *intersection* of the complements of groups, while in several applications, it is desirable to explain the support of $\boldsymbol{x}$ as the *union* of groups in $\mathfrak{G}$. In particular, we can seek the minimal set cover of $\boldsymbol{x}$.

Definition 5 (Minimal weighted set cover)**.**

$$F_{\cup}(S) := \min_{\boldsymbol{\omega} \in \{0,1\}^M} \{\boldsymbol{d}^T \boldsymbol{\omega} : \boldsymbol{B\omega} \geq \mathbb{1}_S\},$$

where $\boldsymbol{d} \in \mathbb{R}_+^M$ are positive group weights, and $\boldsymbol{B}$ is the biadjacency matrix of the bipartite graph representation of $\mathfrak{G}$.

$F_{\cup}$ is a *non-submodular* function that was previously considered in [BBCK13, OJV11, HZM11]. Evaluating $F_{\cup}$ is NP-Hard in general [6], but in some cases, namely when $F_{\cup}$ is an ILP penalty, it can be computed exactly. The latent group Lasso norm (see Section 1.3.1) was proposed as a potential convex surrogate for it, and was later shown in [OB12] to be its *homogeneous* convex envelope. But is this the tightest convex relaxation, even without imposing homogeneity?

In general no; the latent group Lasso norm can be tractably evaluated (if the number of groups is polynomial), while the convex envelope $\Theta_\infty^{\cup}$ of $F_{\cup}$ is in general NP-Hard to evaluate[7]. Hence, the two convex penalties do not coincide in general. However, we show that when $F_{\cup}$ is an ILP penalty they do. In this case, the convex envelope of $F_{\cup}$ is given by Proposition 4, which indeed yields the ℓ_∞-latent group Lasso norm.

[6] $F_{\cup}$ can be approximated by a greedy algorithm achieving an optimal approximation ratio of $O(\log d)$ [Fei98].
[7] In chapter 3, we show that $\Theta_\infty^{\cup}(\mathbb{1}_S) = F_{\cup}(S), \forall S \subseteq V$, hence the intractability of $\Theta_\infty^{\cup}$ follows from that of $F_{\cup}$.

Corollary 2. *If $F_\cup(\mathrm{supp}(\boldsymbol{x}))$ is an ILP penalty, its convex envelope over the unit ℓ_∞-ball is*

$$\begin{aligned}\Theta_\infty^\cup(\boldsymbol{x}) &= \min_{\boldsymbol{\omega}\in[0,1]^M}\{\boldsymbol{d}^T\boldsymbol{\omega} : \boldsymbol{B}\boldsymbol{\omega} \geq |\boldsymbol{x}|\} \\ &= \min_{\boldsymbol{v}\in[-1,1]^{d\times M}}\{\sum_{i=1}^M d_i\|\boldsymbol{v}^{G_i}\|_\infty : \boldsymbol{x} = \sum_{i=1}^M \boldsymbol{v}^{G_i}, \mathrm{supp}(\boldsymbol{v}^{G_i}) \subseteq G_i\}, \end{aligned} \tag{2.3}$$

for $\boldsymbol{x} \in [-1,1]^d$, and $\Theta_\infty^\cup(\boldsymbol{x}) = \infty$ otherwise.

Note that (2.3) differs slightly from the original definition of ℓ_∞-latent group Lasso norm, with the additional constraint $\boldsymbol{v}^{G_i} \in [-1,1]^d$. If we consider instead a more relaxed version of $F_\cup$, where $\boldsymbol{\omega} \in \mathbb{Z}^M$, we recover the usual formulation of latent group Lasso norm, with $\boldsymbol{v}^{G_i} \in \mathbb{R}^d$.

A special case where $F_\cup$ is an ILP penalty occurs when the biadjacency matrix $\boldsymbol{B}$ is TU— we call the corresponding $\mathfrak{G}$ a *TU group structure*. $F_\cup$ is a TU penalty in this case. To see this, note that $F_\cup$ can be written in the form given in Definition 3 with $\boldsymbol{M} = [-\boldsymbol{B}, \boldsymbol{I}_d]$ and $\boldsymbol{c} = \boldsymbol{0}$. Thus, when $\boldsymbol{B}$ is TU, so is $\boldsymbol{M}$ [NW99, Proposition 2.1]. Several group structures considered in the literature are indeed TU group structures.

Example 1 (Bipartite groups)**.** *One important class of TU group structures are groups whose intersection graph is bipartite (which includes acyclic graphs) [BBC+16, Proposition 2]. One such example is illustrated in Figure 2.1. This class clearly includes* non-overlapping groups*, for which $\Theta_\infty^\cup = \Theta_\infty^\cap$. It also includes other interesting group structures, such as the* parent-child *model, defined for variables organized over a tree, where groups consists of all parent-child pairs (see Figure 1.7, left), and the* family *model, where groups consists of each node and all its children (see Figure 1.7, right) [BBC+16]. Such groups are used with the latent group Lasso norm to encourage hierarchical structures, while allowing some flexibility to deviate from the exact tree-model (defined in Section 2.5.2) [BBC+16, RNWK11].*

Example 2 (Interval groups)**.** *Another important class of TU group structures are groups that lead to an interval matrix $\boldsymbol{B}$, i.e., a binary matrix such that in each row the 1s appear consecutively [NW99, Corollary 2.10]. For example, when variables are organized over a tree, groups consisting of each node and its ancestors lead to an interval matrix (after permutation of the columns) [BBC+16]. Such groups are also used with the latent group Lasso norm to encourage hierarchical structures [YB+17]. Another interesting example occurs in the context of dispersive models (c.f., Section 2.5.3).*

Sparse set cover

Both penalties we considered so far only induce sparsity on the group level; if a group is selected, all variables within the group are encouraged to be non-zero. In some applications, it is desirable to enforce sparsity both on the individual and the group level. This motivates the following natural penalty.

Definition 6 (Sparse weighted g-group cover)**.**

$$F_{\cup,g}(S) := \inf_{\boldsymbol{\omega}\in\{0,1\}^M} \{\mathbb{1}^T \boldsymbol{s} : \boldsymbol{B}\boldsymbol{\omega} \geq \boldsymbol{s}, \boldsymbol{d}^T\boldsymbol{\omega} \leq g, \mathbb{1}_S = \boldsymbol{s}\},$$

where $g \in \mathbb{R}_+$ is a group budget, $\boldsymbol{d} \in \mathbb{R}_+^M$ are positive group weights, $\boldsymbol{B}$ is the biadjacency matrix of the bipartite graph representation of $\mathfrak{G}$.

$F_{\cup,g}$ is an extension of the minimal set cover penalty, where instead of looking for the signal with the smallest cover, we seek the sparsest signal that admit a cover with fewer than g groups (if $\boldsymbol{d} = \mathbb{1}$). $F_{\cup,g}$ is a TU penalty whenever $\widetilde{\boldsymbol{B}} = \begin{bmatrix} \boldsymbol{B} \\ \mathbb{1} \end{bmatrix}$ is TU [NW99, Proposition 2.1], which is the case, for example, when $\boldsymbol{B}$ is an interval matrix (c.f., Example 2).

Corollary 3. *If $F_{\cup,g}(\mathrm{supp}(\boldsymbol{x}))$ is an ILP penalty, its convex envelope over the unit ℓ_∞-ball is*

$$\Theta_\infty^{\cup,g}(\boldsymbol{x}) = \inf_{\boldsymbol{\omega}\in[0,1]^M} \{\|\boldsymbol{x}\|_1 : \boldsymbol{B}\boldsymbol{\omega} \geq |\boldsymbol{x}|, \boldsymbol{d}^T\boldsymbol{\omega} \leq g\},$$

for $\boldsymbol{x} \in [-1,1]^d$, and $\Theta_\infty^{\cup,g}(\boldsymbol{x}) = \infty$ otherwise.

The resulting convex program thus combines the latent group Lasso norm with the ℓ_1-norm and provides an alternative to the sparse group Lasso in [SFHT13], for the overlapping groups case. We observe that in this case the convex envelope $\Theta_\infty^{\cup,g}$ is not a norm, and if we were to impose homogeneity on it, we would have completely lost the group sparsity structure. Indeed, we know from [OB12] that for any set function where $F(\{e\}) = 1$ for all singletons $e \in V$ and $F(S) > |A|, \forall A \subseteq V$— a property which applies to $F_{\cup,g}$— the corresponding homogeneous convex envelope is the ℓ_1-norm. We illustrate the performance of sparse g-group cover penalty and the effect of this loss of structure via a numerical example in Section 2.6.1.

2.5.2 Hierarchical sparsity

We study the hierarchical sparsity model (see Section 1.3.1), where the coefficients of $\boldsymbol{x}$ are organized over a tree (or a forest) $\mathcal{T}$, and its support is sparse and form a rooted connected subtree of $\mathcal{T}$, i.e., a node is in the support iff all its ancestors are in the support too (see Figure 2.2). We can naturally describe such a hierarchical model as a TU model.

Definition 7 (Tree ℓ_0-penalty)**.**

$$F_{T,0}(S) := \boldsymbol{d}^T\mathbb{1}_S + \iota_{\boldsymbol{T}\mathbb{1}_S\geq 0}(S)$$

where $\boldsymbol{d} \in \mathbb{R}_+^M$ are positive group weights, and $\boldsymbol{T}$ is the edge-node incidence matrix of the directed tree $\mathcal{T}$, i.e., $T_{li} = 1$ and $T_{lj} = -1$ iff $e_l = (i,j)$ is an edge in $\mathcal{T}$ between parent node i and its child node j, i.e., $\boldsymbol{T}\boldsymbol{s} \geq 0$ encodes the constraint $s_{parent} \geq s_{child}$ for $\boldsymbol{s} = \mathbb{1}_S$.

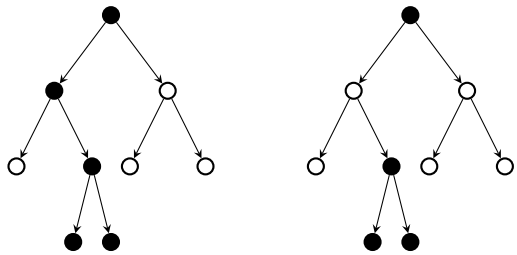


Figure 2.2: Valid selection (left), *Invalid* selection (right)

This is indeed a TU penalty since each row of $\mathcal{T}$ contains two non-zero entries that sum up to zero [NW99, Proposition 2.6].

Corollary 4. *The convex envelope of $F_{T,0}(\mathrm{supp}(\boldsymbol{x}))$ over the unit ℓ_∞-ball is given by*

$$\Theta_\infty^{T,0}(\boldsymbol{x}) = \sum_{G \in \mathfrak{G}_H} \|\boldsymbol{x}_G\|_\infty,$$

for $\boldsymbol{x} \in [-1,1]^d$ and $\Theta_\infty^{T,0}(\boldsymbol{x}) = \infty$ otherwise, where the groups $G \in \mathfrak{G}_H$ are defined as each node and all its descendants in $\mathcal{T}$.

Proof. Since $F_{T,0}(\mathrm{supp}(\boldsymbol{x}))$ is a TU-penalty, its convex envelope is given by Proposition 4.

$$\begin{aligned}\Theta_\infty^{T,0}(\boldsymbol{x}) &= \min_{\boldsymbol{s} \in [0,1]^d} \{\boldsymbol{d}^\top \boldsymbol{s} : \boldsymbol{T}\boldsymbol{s} \geq 0, |\boldsymbol{x}| \leq \boldsymbol{s}\} \\ &= \sum_{G_i \in \mathfrak{G}_H} d_i \|\boldsymbol{x}_{G_i}\|_\infty\end{aligned}$$

The second equality holds since any feasible $\boldsymbol{s}$ satisfies $\boldsymbol{s} \geq |\boldsymbol{x}|$ and $s_{\text{parent}} \geq s_{\text{child}}$. Hence, starting from the leaves, each leaf satisfies $s_i \geq |x_i|$, and to minimize $\boldsymbol{d}^T \boldsymbol{s}$, we simply set $s_i = x_i$. For a node i with two children j, k as leaves, it should satisfy $s_i \geq \max\{|x_i|, |s_j|, |s_k|\}$, thus $s_i = \max\{|x_i|, |x_j|, |x_k|\}$, and so on. Thus, $s_i = \max_{\{j \in G_j\}} |x_j|$ where the group G_j consists of j and all its descendants in $\mathcal{T}$. □

Note that the resulting convex norm $\Theta_\infty^{T,0}$ is the ℓ_∞-hierarchical group Lasso norm [JMOB11], which is commonly used as a convex surrogate for the hierarchical sparsity model. As a special case of the ℓ_∞-group Lasso norm, with $\mathfrak{G} = \mathfrak{G}_H$, $\Theta_\infty^{T,0}$ is known to be the convex envelope of the corresponding $F_\cap$, which we denote by $F_{\cap,\mathfrak{G}_H}$ (see Section 2.5.1). Note though the subtle difference between $F_{\cap,\mathfrak{G}_H}$ and $F_{T,0}$; $F_{\cap,\mathfrak{G}_H}$ only *encourages* the tree structure, while $F_{T,0}$ *enforces* it. Indeed, the two penalties are only equal for sets that satisfy the tree structure, i.e., $F_{\cap,\mathfrak{G}_H}(S) = F_{T,0}(S), \forall S \subseteq V, \boldsymbol{T}\mathbb{1}_S \geq 0$. Note also that unlike $F_{\cap,\mathfrak{G}_H}$, $F_{T,0}$ is not submodular. This difference is lost though in the convex domain, since $F_{T,0}$ and $F_{\cap,\mathfrak{G}_H}$ have the same convex envelope. We explain this artifact in Chapter 3.

2.5.3 Dispersive sparsity

The sparsity models we considered thus far encourage clustering. The implicit structure in these models is that coefficients within a group exhibit a positive, reinforcing correlation. Loosely speaking, if a coefficient within a group is important, so are the others. However, in many applications, the opposite behavior may be true. That is, sparse coefficients within a group compete against each other [ZJH10, HDC09, GK02].

Hence, we describe models that encourage the dispersion of sparse coefficients. Here, dispersive models still inherit a known group structure $\mathfrak{G}$, which underlie their interactions in the opposite manner to the group models in Section 2.5.1.

Group dispersive model

One natural model for dispersiveness is to allow only a certain budget of coefficients, e.g., only one, to be selected in each group. This model can be expressed by the following two set functions:

$$F_{\boldsymbol{D}}(S) := \begin{cases} 0 & \text{if } S = \emptyset \\ 1 & \text{if } \max_{G \in \mathfrak{G}} |S \cap G| \leq 1 \\ \infty & \text{otherwise} \end{cases}, \qquad \tilde{F}_{\boldsymbol{D}}(S) := \max_{G \in \mathfrak{G}} |S \cap G|,$$

where $F_{\boldsymbol{D}}$ enforces the dispersive structure, while $\tilde{F}_{\boldsymbol{D}}$ only encourages it. Both functions are *not* submodular. Whenever the groups in $\mathfrak{G}$ form a *partition* of V, [OB12] showed that the *homogeneous* convex envelope of both $F_{\boldsymbol{D}}$ and $\tilde{F}_{\boldsymbol{D}}$ is the exclusive Lasso norm in [ZJH10] (see Section 1.3.1). Is this the tightest convex relaxation for both penalties, for other group structures, and without imposing homogeneity?

In what follows, we show that the ℓ_∞-exclusive Lasso norm is the convex envelope of $\tilde{F}_{\boldsymbol{D}}$, whenever it is an ILP penalty, which holds in particular for TU group structures, including partition groups. But it is *not* the convex envelope of $F_{\boldsymbol{D}}$. First, we express $F_{\boldsymbol{D}}$ and $\tilde{F}_{\boldsymbol{D}}$ as IPs.

Definition 8 (Dispersive penalties)**.** *We can rewrite $F_{\boldsymbol{D}}$ and $\tilde{F}_{\boldsymbol{D}}$ as*

$$F_{\boldsymbol{D}}(S) = \inf_{\omega \in \{0,1\}} \{\omega : \boldsymbol{B}^T \mathbb{1}_S \leq \omega \mathbb{1}\}, \text{ and } \qquad \tilde{F}_{\boldsymbol{D}}(S) = \inf_{\omega \in \mathbb{Z}_+} \{\omega : \boldsymbol{B}^T \mathbb{1}_S \leq \omega \mathbb{1}\},$$

where $\boldsymbol{B}$ is the biadjacency matrix of the bipartite graph representation of $\mathfrak{G}$.

Both $F_{\boldsymbol{D}}$ and $\tilde{F}_{\boldsymbol{D}}$ are TU penalties whenever $\boldsymbol{B}$ is TU, since $\boldsymbol{B}^T$ is then also TU [NW99, Proposition 2.1]. Recall from Example 1 that partition groups lead to a TU matrix $\boldsymbol{B}$. Another important example of a TU group structure arises from the simple one-dimensional model of neuronal signals suggested by [HDC09]. In this model, neuronal signals are modeled as a train of spike signals with some refractory period $\Delta \geq 0$, where the minimum distance between two non-zeros is Δ (see Figure 1.8). This structure can be enforced via an interval matrix $\boldsymbol{B}^T = \boldsymbol{D}$,

where each row (corresponding to a group) consists of Δ consecutive ones, which is TU (see Example 2).

$$\boldsymbol{D} = \begin{bmatrix} 1 & 1 & \cdots & 1 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 1 & \cdots & 1 & 1 & 0 & \cdots & 0 \\ & & & & \ddots & & & & \\ 0 & \cdots & 0 & 0 & 1 & 1 & \cdots & 1 & 1 \end{bmatrix}_{(d-\Delta+1)\times d}$$

Corollary 5. *If $F_{\boldsymbol{D}}(\mathrm{supp}(\boldsymbol{x}))$ is an ILP penalty, its convex envelope over the unit ℓ_∞-ball is*

$$\Theta_\infty^{\boldsymbol{D}}(\boldsymbol{x}) = \max_{G\in\mathfrak{G}} \|\boldsymbol{x}_G\|_1 + \iota_{\boldsymbol{B}^T|\boldsymbol{x}|\leq\mathbb{1}}(\boldsymbol{x}),$$

for $\boldsymbol{x} \in [-1,1]^d$ and $\Theta_\infty^{\boldsymbol{D}} = \infty$ otherwise. Similarly, if $\tilde{F}_{\boldsymbol{D}}(\mathrm{supp}(\boldsymbol{x}))$ is an ILP penalty, its convex envelope over the unit ℓ_∞-ball is $\tilde{\Theta}_\infty^{\boldsymbol{D}}(\boldsymbol{x}) = \max_{G\in\mathfrak{G}} \|\boldsymbol{x}_G\|_1$ for $\boldsymbol{x} \in [-1,1]^d$ and $\tilde{\Theta}_\infty^{\boldsymbol{D}} = \infty$ otherwise.

Notice that unlike the homogeneous convex envelope of $F_{\boldsymbol{D}}$, $\Theta_\infty^{\boldsymbol{D}}$ is not exactly the exclusive Lasso norm; it has an additional budget constraint $\boldsymbol{B}^T|\boldsymbol{x}| \leq \mathbb{1}$. Thus in this case, imposing homogeneity leads to the loss of part of the structure.

Sparse group dispersive model

In some applications, it may be desirable to seek the sparsest signal satisfying the dispersive structure. This can be achieved by incorporating sparsity into the dispersive penalty F_D.

Definition 9 (Dispersive ℓ_0-penalty)**.**

$$F_{\boldsymbol{D},0}(S) := |S| + \iota_{\boldsymbol{B}^T\mathbb{1}_S\leq\mathbb{1}}(S),$$

where $\boldsymbol{B}$ is the biadjacency matrix of the bipartite graph representation of $\mathfrak{G}$.

$F_{\boldsymbol{D},0}$ is again a TU penalty when $\boldsymbol{B}$ is TU, and its convex envelope follows from proposition 4.

Corollary 6. *If $F_{\boldsymbol{D},0}(\mathrm{supp}(\boldsymbol{x}))$ is an ILP penalty, its convex envelope over the unit ℓ_∞-ball is*

$$\Theta_\infty^{\boldsymbol{D},0}(\boldsymbol{x}) = \|\boldsymbol{x}\|_1 + \iota_{\boldsymbol{B}^T|\boldsymbol{x}|\leq\mathbb{1}}(\boldsymbol{x}),$$

for $\boldsymbol{x} \in [-1,1]^d$ and $\Theta_\infty^{\boldsymbol{D},0} = \infty$ otherwise.

We observe in this case too that the convex envelope $\Theta_\infty^{\boldsymbol{D},0}$ is not a norm (see Figure 2.3), and if we were to impose homogeneity on it, we would have completely lost the dispersive structure. Indeed, similar to the sparse g-group cover penalty $F_{\cup,g}$, the homogeneous convex envelope of $F_{\boldsymbol{D},0}$ is the ℓ_1-norm. We illustrate the performance of the dispersive ℓ_0-penalty and effect of this loss of structure via a numerical example in Section 2.6.2.

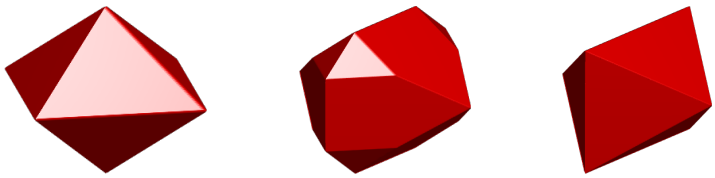


Figure 2.3: $\Theta_{\infty}^{\boldsymbol{D},0}(\boldsymbol{x}) \leq 1$ (left) $\Theta_{\infty}^{\boldsymbol{D},0}(\boldsymbol{x}) \leq 1.5$ (middle) $\Theta_{\infty}^{\boldsymbol{D},0}(\boldsymbol{x}) \leq 2$ (right) for $\mathfrak{G} = \{\{1,2\},\{2,3\}\}$

2.6 Experiments

In this section, we assess the performance of two novel penalties we have proposed in section 2.5.1, the sparse g-group cover and the dispersive ℓ_0-penalty, in estimating a sparse signal $\boldsymbol{x}^\natural \in \mathbb{R}^d$, whose support satisfy the structure assumed under these two models. In particular, we consider the problem of recovering $\boldsymbol{x}^\natural$ from compressive measurements $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}^\natural + \boldsymbol{\varepsilon}$, where $\boldsymbol{A} \in \mathbb{R}^{n\times d}$ is a measurement matrix, and $\boldsymbol{\varepsilon} \in \mathbb{R}^n$ is a noise vector. We compare the performance of the estimated solutions obtained by solving

$$\boldsymbol{x}^\star \in \underset{\boldsymbol{x}\in[-1,1]^d}{\arg\min}\{\Theta(\boldsymbol{x}) : \|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\| \leq \|\boldsymbol{\varepsilon}\|\}, \tag{2.4}$$

using either the classical ℓ_1-norm, $\Theta(\boldsymbol{x}) = \|\boldsymbol{x}\|_1$, or the convex envelope Θ_{∞}^{ILP}, with the ILP penalty tailored to the structure. This comparison also serves to illustrate the impact of imposing homogeneity, since as discussed in section 2.5.1, the homogeneous convex envelope of both penalties we are considering here is the ℓ_1-norm.

2.6.1 Sparse g-group cover model

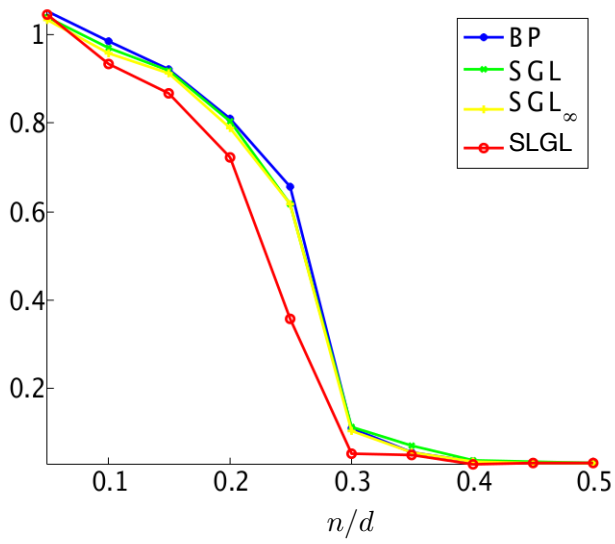
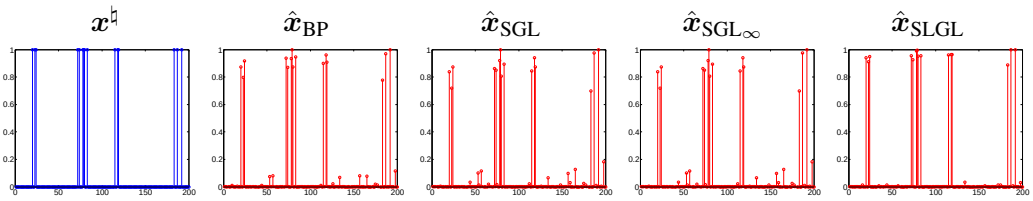


Figure 2.4: Recovery error of SLGL, SGL, SGL$_\infty$, and BP

We consider the problem of estimating $\boldsymbol{x}^\natural$, when its support is sparse and can be covered with *few*



Error 0.128 Error 0.181 Error 0 .085 Error 0.058

Figure 2.5: Example of a generated sparse signal with $g = 5$ group cover (blue), and the corresponding estimated signals (red) with BP, SGL, SGL$_\infty$, and SLGL, for $n = 0.25d$. Each plot represents the value of x_i as a function of i.

groups. In this setting, sparse group Lasso defined as $(1-\alpha)\sum_{G\in\mathfrak{G}}\sqrt{|G|}\|\boldsymbol{x}_G\|_p + \alpha\|\boldsymbol{x}_G\|_1$, where $p = 2$ and $\alpha \in [0, 1]$, was proposed in [SFHT13] to induce the desired effect of sparsity on both the individual and the group level. In our framework, the proposed sparse g-group cover $F_{\cup,g}$ is a natural penalty to consider in this case.

We generate an s-sparse signal $\boldsymbol{x}^\natural$, with $s = 15$, in dimensions $d = 200$, covered by $g = 5$ groups, randomly chosen from the $M = 29$ groups. The groups generated are interval groups, of equal size of 10 coefficients, and with an overlap of 3 coefficients between each two consecutive groups. The true signal $\boldsymbol{x}^\natural$ has 3 non-zero coefficients, with equal value set to one, in each of its 5 active groups (see Figure 2.5). As discussed in example 2, interval groups are a TU group structure, hence $\Theta_\infty^{\cup,g}$ is the tightest convex relaxation of $F_{\cup,g}$ in this case. We draw a noise vector $\boldsymbol{\varepsilon}$ with i.i.d Gaussian entries of variance $\sigma = 0.01$ and a measurement matrix $\boldsymbol{A}$ with normalized columns of standard i.i.d. Gaussian entries. We solve problem (2.4) with the data-fidelity constraint $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2 \leq \|\boldsymbol{\varepsilon}\|_2$, using the true ℓ_2-norm of the noise, and with the regularizer Θ chosen to be either the ℓ_1-norm (BP), or the convex envelope $\Theta_\infty^{\cup,g}$, which we call sparse latent group Lasso (SLGL), or the sparse group Lasso norm (SGL). We also compare against SGL$_\infty$ where we set $p = \infty$ in SGL, which is better suited for signals with equal valued non-zero coefficients.

We use the convex solver CVX [GB14] to obtain high accuracy solutions $\hat{x}$ to each formulation. We generate the data randomly 10 times and report the averaged results. Figure 2.4 plots the relative recovery error $\frac{\|\boldsymbol{x}^\natural - \hat{\boldsymbol{x}}\|_2}{\|\boldsymbol{x}^\natural\|_2}$ achieved with the different regularizers, as we vary the number of measurements n. Figure 2.5 shows the estimated solutions obtained with each regularizer, and their corresponding relative recovery error, for $n = 0.25d$ measurements. Since the true signal exhibit strong overall sparsity we use $\alpha = 0.95$ in SGL as suggested in [SFHT13] (we tried several values of α, and this seemed to give the best results for SGL). SLGL clearly outperforms the other regularizers. Indeed, SLGL is able to capture the correct underlying structure leading to better reconstruction. We can observe from Figure 2.5 that the solution returned by SLGL have almost no non-zeros outside the correct support, while solutions returned by ℓ_1-norm and SGL violate the assumed model.

2.6.2 Sparse dispersive model

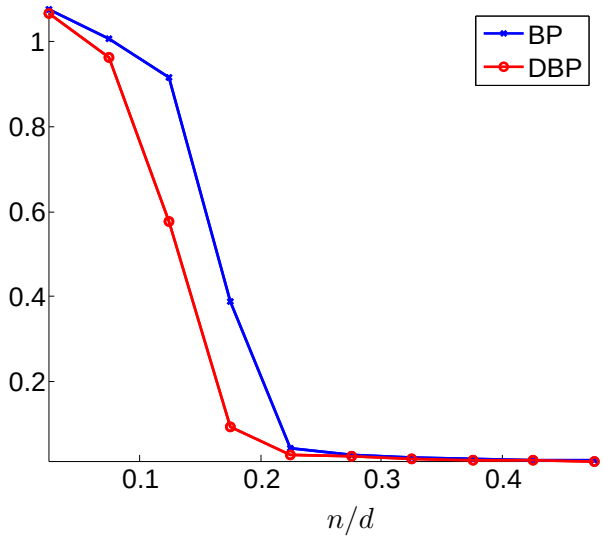


Figure 2.6: Recovery error of BP and DBP

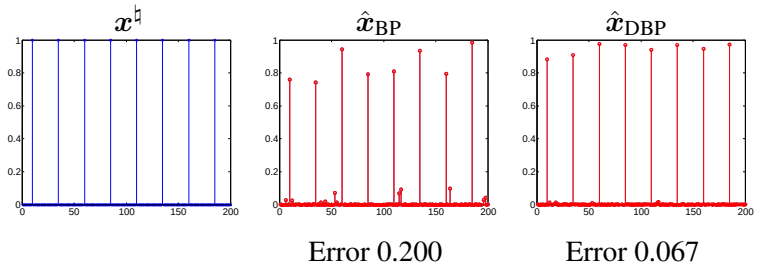


Figure 2.7: Example of a generated spike train (blue), and the corresponding estimated signals (red) with BP and DBP, for $n = 0.18d$. Each plot represents the value of x_i as a function of i.

We consider the problem of estimating a spike train signal $\boldsymbol{x}^\natural$, i.e., a signal with a minimum distance $\Delta > 0$ between any two non-zero coefficients. As discussed in section 2.5.3, a natural penalty to consider in this setting is the dispersive ℓ_0-penalty $F_{\boldsymbol{D},0}$ with $\boldsymbol{B}^T = \boldsymbol{D}$, which is a TU penalty in this case, and hence $\Theta_\infty^{\boldsymbol{D},0}$ is its tightest convex relaxation.

We generate a spike train $\boldsymbol{x}^\natural$ in dimensions $d = 200$ with a refractory period of $\Delta = 25$ and with all non-zero coefficients set to one (see Figure 2.7). We draw a *sparse* noise vector $\boldsymbol{\varepsilon}$ with 15 i.i.d non-zero Gaussian entries of variance $\sigma = 0.01$ and a measurement matrix $\boldsymbol{A}$ with normalized columns of standard i.i.d. Gaussian entries. We solve problem (2.4) with the data-fidelity constraint $\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_1 \leq \|\boldsymbol{\varepsilon}\|_1$ (since the noise is sparse), using the true ℓ_1-norm of the noise, and with the regularizer Θ chosen to be either the ℓ_1-norm (BP) or the convex envelope $\Theta_\infty^{\boldsymbol{D},0}$ (DBP).

We use the convex solver CVX [GB14] to obtain high accuracy solutions $\hat{x}$ to each formulation. We generate the data randomly 20 times and report the averaged results. Figure 2.6 plots the relative recovery error $\frac{\|\boldsymbol{x}^\natural - \hat{\boldsymbol{x}}\|_2}{\|\boldsymbol{x}^\natural\|_2}$, achieved with the two regularizers, as we vary the number

of measurements n. Figure 2.7 shows the estimated solutions obtained with each regularizer, and their corresponding relative recovery error, for $n = 0.18d$ measurements. DBP clearly outperforms BP. Indeed, we can see from Figure 2.6 that DBP requires less measurements than BP to achieve a given error, which confirms the theoretical characterization in [HDC09]. We can also observe from Figure 2.7 that the solution returned by DBP have almost no non-zeros within the Δ intervals, while the solution returned by ℓ_1-norm violate this model.

Finally, both numerical illustrations in Section 2.6.1 and 2.6.2 reinforces the message that certain structures cannot be captured by homogeneous convex penalties.

2.7 Discussion

We have presented a principled recipe for designing structure-inducing convex penalties, by exploiting classical tools from linear programming. Given some desired structural constraints on the support, we express them via an integer program, then check if the linear constraints result in an integral polytope. Testing this property can be done with the help of two sufficient conditions of integrality, total unimodularity and total dual integrality, which can be verified by polynomial time algorithms or by inspection. The resulting convex envelope is easy to evaluate via LP and the corresponding learning problems can be efficiently solved via standard optimization techniques.

This simple recipe allowed us to rederive several prevalent structure-inducing norms, as well as define new interesting convex penalties. Several of our proposed penalties are not norms. Enforcing homogeneity in such cases leads to an unnecessary loss of structure. We provided simulations on synthetic examples that illustrate the effect of this loss, in comparison with our proposed non-homogeneous convex penalties, which achieve better recovery performance.

The discussion in this chapter leads to several questions worth investigating, such as: For which class of set functions the homogeneous convex envelope leads to a loss of structure, not incurred by its non-homogeneous counterpart? When do the estimators, using either the homogeneous or non-homogeneous relaxation of non-submodular functions, recover the true parameter vector? What is the (non-homogeneous) convex envelope of $F(\mathrm{supp}(\boldsymbol{x}))$ when regularized with general ℓ_p-norms. These questions will be addressed in Chapter 3, and the following additional questions are good directions for future research.

Open question 1. The examples of structures captured by ILP penalties, we provided in Section 2.5, all belong to the subclass of TU penalties. *What are examples of structures that can be captured by ILP penalties that are not TU penalties?* Identifying such examples would allow us to provide tight convex relaxations for more complicated structures (e.g., minimal set covers with non-TU groups). The TDI sufficient condition can be useful in this respect, but is harder to check just by inspection.

Open question 2. In Section 2.2, we saw that the convex closure of any set function is the only convex extension satisfying the assumptions in Lemma 1. An important question to answer then

is: *What are set functions, beyond submodular and ILP penalties, admitting a tractable convex closure?* By Lemma 1, such set functions will then also admit tractable convex envelopes.

Finally, it is our hope that by expanding the class of functions that can be addressed by disciplined convex approaches, beyond submodular functions, our framework will inspire new applications of structured sparsity in various fields.

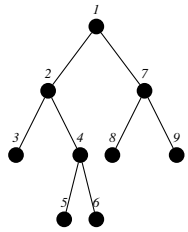
2.8 Appendix: Review of total unimodularity

We list several properties which are helpful to identify totally unimodular (TU) matrices. Recall that a matrix $\boldsymbol{M} \in \mathbb{R}^{l\times m}$ is TU iff the determinant of every square submatrix of $\boldsymbol{M}$ is 0, or ± 1.

Theorem 1 (Equivalent characterization, [NW99, Theorem 2.7]). *A matrix $\boldsymbol{M} \in \mathbb{R}^{l\times m}$ is TU iff for every $J \subseteq \{1, \cdots, m\}$, there exits a partition J_1, J_2 of J such that $|\sum_{j\in J_1} M_{ij} - \sum_{j\in J_2} M_{ij}| \leq 1$ for $i = 1, \cdots, l$.*

Example 3 (Tree matrix)**.** *A simple example of a TU matrix, where it is easy to see that the condition in Theorem 1 holds, is given by the constraint matrix of the tree ℓ_0-penalty (see Def. 7):*

$$\boldsymbol{T} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \end{bmatrix}$$



For every $J \subseteq \{1, \cdots, m\}$, we can choose $J_1 = J$ and $J_2 = \emptyset$, $|\sum_{j\in J_1} T_{ij}| \leq 1$.

Proposition 5 (TU preserving operations, [NW99, Proposition 2.1])**.** *M is TU iff the matrices obtained by the following operations are also TU.*

1. *Taking the transpose of $\boldsymbol{M}$.*
2. *Appending identity: $(\boldsymbol{M}, \boldsymbol{I})$.*
3. *Deleting a row (column) with at most one non-zero entry from M.*
4. *Interchanging two rows (columns) in $\boldsymbol{M}$.*
5. *Multiplying a row (column) from $\boldsymbol{M}$ by (-1).*
6. *Duplicating rows (column) of $\boldsymbol{M}$.*
7. *Applying pivot operation (operations in reduced Gaussian elimination) on $\boldsymbol{M}$.*

Proposition 6 (Some sufficient conditions for TU, [NW99])**.** *M is TU if any of the following conditions hold.*

1. *M has no more than* two non-zero elements *in each column, then $\boldsymbol{M}$ is TU iff its rows can be partitioned into two subsets such that elements of same sign are in different sets, and elements of opposite sign are in the same set.*
2. *M is the edge-node incidence matrix of a bipartite graph (this is a special case of the previous condition).*
3. *M is an interval matrix (i.e., whose columns has consecutive ones).*

3 Homogeneous and Non-Homogeneous Convex Relaxations

3.1 Introduction

In this chapter, we continue with the convex approach to structured sparse learning. In our discussion so far, we have seen two systematic approaches to relax a combinatorial penalty expressing the desired structured sparsity model to a convex penalty. The approach adopted in [Bac10a] and Chapter 2 considers the tightest convex relaxation of the combinatorial penalty over the unit ℓ_∞-ball. While the approach proposed by [OB12] considers the tightest positively *homogeneous* convex relaxation of the combinatorial penalty regularized by an ℓ_p-norm. Homogeneity is a natural requirement imposed to ensure the invariance of the regularizer to rescaling of the data. By going from the discrete to the convex domain, some of the structure is necessarily lost under both relaxations. However, in Chapter 2 we observed that the homogeneity requirement may cost further loss of structure in several cases. This observation motivates the following question:

> When do the *homogeneous* and *non-homogeneous* convex relaxations differ and which structures can be encoded by each?

In order to answer this question rigorously, we are interested in studying the algebraic and geometric properties of both relaxations, as well as their statistical properties in the context of regularized learning problems.

3.1.1 Related work

The algebraic and geometric properties of the homogeneous convex relaxation are well-studied in [OB12]. The authors show, for instance, that the resulting norm takes the form of a generalized latent group Lasso norm [OJV11]. They also show that any *monotone submodular* set function is preserved, in some sense, by such relaxation. The statistical properties though of these norms were only investigated so far in special cases, e.g., for norms associated with monotone submodular functions [OB12], and for the latent group Lasso norm [OJV11].

3.1.2 Contributions

In this chapter, we study the algebraic and geometric properties of the non-homogeneous relaxation and identify which combinatorial structures are preserved by it in a manner similar to [OB12] for the homogeneous one. We further study the statistical properties of both relaxations. In particular, this chapter makes the following contributions:

- We derive formulations of the non-homogeneous tightest convex relaxation of general ℓ_p-regularized combinatorial penalties (Section 3.2.1). We show that any *monotone* set function is preserved by such relaxation (Section 3.2.2).
- We identify necessary conditions for support recovery in learning problems regularized by general convex penalties (Section 3.3.1).
- We propose an adaptive weight estimation scheme and provide sufficient conditions for support recovery under the asymptotic regime (Section 3.3.2). This scheme does not require any irrepresentability condition and is applicable to general monotone convex regularizers.
- We identify sufficient conditions with respect to combinatorial penalties which ensure that the sufficient support recovery conditions hold with respect to the associated convex relaxations (Section 3.4).
- We illustrate numerically the effect on support recovery of the choice of the relaxation as well as the adaptive weights scheme (Section 3.5).

This chapter is based on the joint work with Francis Bach and Volkan Cevher [EHBC18].

In the sequel, we defer all proofs to the appendix of this chapter.

3.2 Combinatorial penalties and convex relaxations

We consider positive-valued set functions of the form $F : 2^V \to \overline{\mathbb{R}}_+$, where $V = \{1, \cdots, d\}$, such that $F(\varnothing) = 0, F(A) > 0, \forall A \subseteq V$, to encode structured sparsity models. For generality, we do not assume a priori that F is *monotone*. However, as we argue in the sequel, convex relaxations of non-monotone set functions is hopeless. Also, unless explicitly stated, we do not assume that F is *submodular*. The domain of F is defined as $\mathcal{D} := \{A : F(A) < +\infty\}$. We assume that it covers V, i.e., $\cup_{A \in \mathcal{D}} A = V$, which is equivalent to assuming F is finite at singletons if F is monotone.

We consider the same model in [OB12], parametrized by $\boldsymbol{x} \in \mathbb{R}^d$, with general ℓ_p-regularized combinatorial penalties:

$$F_p(\boldsymbol{x}) = \frac{1}{q} F(\mathrm{supp}(\boldsymbol{x})) + \frac{1}{p} \|\boldsymbol{x}\|_p^p$$

for $p \geq 1$, where the set function F controls the structure of the model in terms of allowed/favored non-zero patterns and the ℓ_p-norm serves to control the magnitude of the coefficients. Allowing

F to take infinite values let us enforce hard constraints. For $p = \infty$, F_p reduces to $F_\infty(\boldsymbol{x}) = F(\mathrm{supp}(\boldsymbol{x})) + \iota_{\|\boldsymbol{x}\|_\infty \leq 1}(\boldsymbol{x})$. Considering the case $p \neq \infty$ is appealing to avoid the clustering artifacts of the values of the learned vector induced by the ℓ_∞-norm.

We study two natural candidates for a convex surrogate of F_p; the homogeneous convex envelope Ω_p of F_p, i.e., the convex envelope of its positively homogeneous envelope given by $F(\mathrm{supp}(\boldsymbol{x}))^{1/q}\|\boldsymbol{x}\|_p$ (see Section 1.3.4), and the direct convex envelope Θ_p of F_p. Note that from the definition of convex envelope, it holds that $\Theta_p \geq \Omega_p$.

3.2.1 Homogeneous and non-homogeneous convex envelopes

In [OB12], the homogeneous convex envelope Ω_p of F_p was shown to correspond to the latent group Lasso norm with groups set to all elements of the power set 2^V. We recall this form of Ω_∞ in Lemma 2 as well as a variational form of Ω_p which highlights the relation between the two. Other variational forms can be found in the Appendix.

Lemma 2 ([OB12])**.** *The homogeneous convex envelope Ω_p of F_p is given by*

$$\Omega_p(\boldsymbol{x}) = \inf_{\boldsymbol{\eta} \in \mathbb{R}_+^d} \frac{1}{p} \sum_{j=1}^d \frac{|x_j|^p}{\eta_j^{p-1}} + \frac{1}{q}\Omega_\infty(\boldsymbol{\eta}), \tag{3.1}$$

$$\Omega_\infty(\boldsymbol{x}) = \min_{\boldsymbol{\alpha} \geq 0} \Big\{ \sum_{S \subseteq V} \alpha_S F(S) : \sum_{S \subseteq V} \alpha_S \mathbb{1}_S \geq |\boldsymbol{x}| \Big\}. \tag{3.2}$$

The non-homogeneous convex envelope Θ_p of F_p is only considered thus far in the case where $p = \infty$. In chapter 2, we showed that $\Theta_\infty(\boldsymbol{x}) = \inf_{\boldsymbol{\eta} \in [0,1]^d}\{f^-(\boldsymbol{\eta}) : \boldsymbol{\eta} \geq |\boldsymbol{x}|\}$, where f^- is the convex closure of F, i.e., the largest convex lower bound of F on $[0,1]^d$ (see Appendix A.2).

Lemma 3 presents a variational form of Θ_∞ that parallels (3.2). We also derive the non-homogeneous convex envelope Θ_p of F_p for any $p \geq 1$ and present the variational form relating it to Θ_∞ in Lemma 3. For simplicity, the variational form (3.3) presented below holds only for monotone functions F; the general form and other variational forms that parallel the ones known for the homogeneous envelope are presented in the appendix of this chapter.

Lemma 3. *The non-homogeneous convex envelope Θ_p of F_p, for monotone functions F, is given by*

$$\Theta_p(\boldsymbol{x}) = \inf_{\boldsymbol{\eta} \in [0,1]^d} \frac{1}{p} \sum_{j=1}^d \frac{|x_j|^p}{\eta_j^{p-1}} + \frac{1}{q}\Theta_\infty(\boldsymbol{\eta}), \tag{3.3}$$

$$\Theta_\infty(\boldsymbol{x}) = \min_{\alpha \geq 0} \Big\{ \sum_{S \subseteq V} \alpha_S F(S) : \sum_{S \subseteq V} \alpha_S \mathbb{1}_S \geq |\boldsymbol{x}|, \sum_{S \subseteq V} \alpha_S = 1 \Big\}. \tag{3.4}$$

The infima in (3.1) and (3.3), for $\boldsymbol{x} \in \mathrm{dom}(\Theta_p)$, can be replaced by a minimization, if we extend $b \to \frac{a}{b}$ by continuity in zero with $\frac{a}{0} = \infty$ if $a \neq 0$ and 0 otherwise, as suggested in [JOB10]

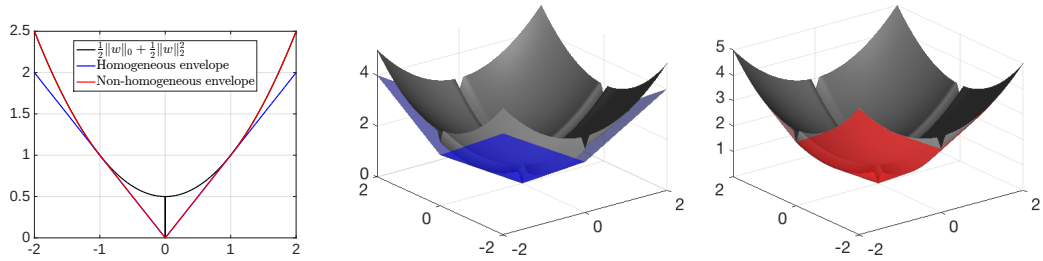


Figure 3.1: ℓ_2-regularized cardinality example in one dimension (left) and two dimensions (middle: homogeneous, right: non-homogeneous).

and [BJM+12]. Note that, for $p = 1$, both relaxations reduce to $\Omega_1 = \Theta_1 = \|\cdot\|_1$. Hence, the ℓ_1-relaxations essentially lose the combinatorial structure encoded in F. We thus follow up on the case $p > 1$.

In order to decide when to employ Ω_p or Θ_p, it is of interest to study the respective properties of these two relaxations and to identify when they coincide. We start by recalling in Remark 3 that the homogeneous and non-homogeneous envelopes are identical, for $p = \infty$, for monotone submodular functions.

Remark 3. *If F is a monotone submodular function, then $\Theta_\infty(\boldsymbol{x}) = \Omega_\infty(\boldsymbol{x}) = f_L(|\boldsymbol{x}|), \forall \boldsymbol{x} \in [-1, 1]^d$, where f_L denotes the Lovász extension of F (see [OB12] and [Bac10a]).*

The two relaxations do not coincide in general: Note the added constraints $\eta \in [0, 1]^d$ in (3.3) and the sum constraint on α in (3.4). Another clear difference to note is that Ω_p are norms that belong to the broad family of H-norms [MMP13, BJM+12], as shown in [OB12]. On the other hand, by virtue of being non-homogeneous, Θ_p are not norms in general. We illustrate below two interesting examples where Ω_p and Θ_p differ.

Example 4 (Berhu penalty)**.** *Since the cardinality function $F(S) = |S|$ is a monotone submodular function, $\Theta_\infty(\boldsymbol{x}) = \Omega_\infty(\boldsymbol{x}) = \|\boldsymbol{x}\|_1$. However, this is not the case for $p \neq \infty$. In particular, we consider the ℓ_2-regularized cardinality function $F_2^{card}(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x}\|_0 + \frac{1}{2}\|\boldsymbol{x}\|_2^2$. Figure 3.1 shows that the non-homogeneous envelope is* tighter *than the homogeneous one in this case. Indeed, Ω_2^{card} is simply the ℓ_1-norm, while Θ_2^{card} is given by $[\Theta_2^{card}(\boldsymbol{x})]_i = |x_i|$ if $|x_i| \leq 1$ and $[\Theta_2^{card}(\boldsymbol{x})]_i = \frac{1}{2}|x_i|^2 + \frac{1}{2}$ otherwise. This penalty, called "Berhu," is introduced in [Owe07] to produce a robust ridge regression estimator and is shown to be the convex envelope of F_2^{card} in [JSK11].*

This kind of behavior, where the non-homogeneous relaxation Θ_p acts as an ℓ_1-norm on the small coefficients and as $\frac{1}{p}\|\boldsymbol{x}\|_p^p$ for large ones, is not limited to the Berhu penalty, but holds for general set functions. However the point where the penalty moves from one mode to the other depends on the structure of F and is different along each coordinate. This is easier to see via the second variational form of Θ_p presented in the Appendix. We further illustrate in the following example.

Example 5 (Range penalty)**.** *Consider the range function defined as range$(A) = \max(A) - \min(A) + 1$ where $\max(A)$ ($\min(A)$) denotes maximal (minimal) element in A. This penalty*

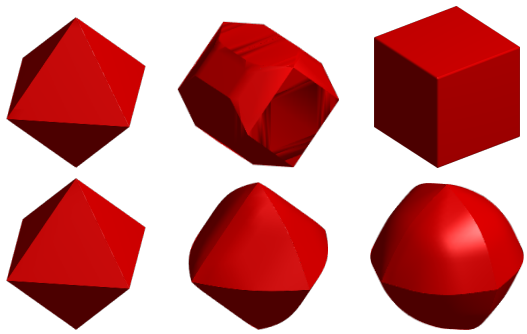


Figure 3.2: Balls of different radii of the non-homogeneous ℓ_∞-convex envelope of the range function (top): $\Theta_\infty(\boldsymbol{x}) \leq 1$ (left), $\Theta_\infty(\boldsymbol{x}) \leq 2$ (middle), $\Theta_\infty(\boldsymbol{x}) \leq 3$ (right) and of its ℓ_2-convex envelope (bottom): $\Theta_2(\boldsymbol{x}) \leq 1$ (left), $\Theta_2(\boldsymbol{x}) \leq 2$ (middle), $\Theta_2(\boldsymbol{x}) \leq 4$ (right).

allow us to favor the selection of interval non-zero patterns on a chain or rectangular patterns on grids. It was shown in [OB12] that $\Omega_p(\boldsymbol{x}) = \|\boldsymbol{x}\|_1$ for any $p \geq 1$. On the other hand, Θ_p has no closed form solution, but is different from the ℓ_1-norm. Figure 3.2 illustrates the balls of different radii of Θ_∞ and Θ_2. We can see how the penalty morphs from ℓ_1-norm to ℓ_∞ and squared ℓ_2-norm respectively, with different "speed" along each coordinate. Looking carefully for example on the ball $\Theta_2(\boldsymbol{x}) \leq 2$, we can see that the penalty acts as an ℓ_1-norm along the (x, z)-plane and as a squared ℓ_2-norm along the (y, z)-plane.

We highlight other ways in which the two relaxations differ and their implications in the sequel.

Computational complexity: Note that Ω_p and Θ_p are still intractable to compute and optimize, in general. However, for certain classes of functions, they are tractable. For example, since for monotone submodular functions, $\Omega_\infty = \Theta_\infty$ is the Lovász extension of F, as stated in Remark 3, then they can be efficiently computed (see Section 1.3.3). Moreover, efficient algorithms to compute Ω_p, and the associated proximal operator, and to solve learning problems regularized with Ω_p are proposed in [OB12] (see also Sections 1.3.4 and 1.4.1). Similarly, if F can be expressed by an *integral linear program* as in Chapter 2, then Ω_∞ , Θ_∞ and their Fenchel conjugate operators can be computed efficiently by linear programs (see Section 2.4). Hence, we can use *conditional gradient* algorithms for numerical solutions. Note also that the formulations (3.1) and (3.3) are jointly convex in $(\boldsymbol{x}, \boldsymbol{\eta})$, since $(z, t) \to t|\frac{z}{t}|^p$ is the perspective function of $z \to |z|^p$ (see [BV04, p.89]). It is then possible to compute and optimize Ω_p and Θ_p for general p, whenever the case $p = \infty$ admit efficient algorithms.

3.2.2 Lower combinatorial envelopes

In this section, we are interested in analyzing which combinatorial structures are preserved by each relaxation. To that end, we generalize the notion of *lower combinatorial envelope* (LCE) [OB12]. The homogeneous LCE F_- of F is defined as the set function which agrees with the ℓ_∞-homogeneous convex relaxation of F at the vertices of the unit hypercube, i.e., $F_-(A) = \Omega_\infty(\mathbb{1}_A), \forall A \subseteq V$.

For the non-homogeneous relaxation, we define the non-homogeneous LCE similarly as $\tilde{F}_-(A) = \Theta_\infty(\mathbb{1}_A)$. The ℓ_∞-relaxation reflects most directly the combinatorial structure of the function F. Indeed, ℓ_p-relaxations only depend on F through the ℓ_∞-relaxation as expressed in the variational forms (3.1) and (3.3).

We say Ω_∞ is a tight relaxation of F if $F = F_-$. Similarly, Θ_∞ is a tight relaxation of F if $\tilde{F}_- = F$. Ω_∞ and Θ_∞ are then *extensions* of F from $\{0,1\}^d$ to $\mathbb{R}^d$; in this sense, the relaxation is tight for all $\boldsymbol{x}$ of the form $\boldsymbol{x} = \mathbb{1}_A$. Moreover, following the definition of convex envelope, the relaxation Ω_∞ (resp. Θ_∞) is the same for F and F_- (resp. F and $\tilde{F}_-$), and hence, the LCE can be interpreted as the combinatorial structure preserved by each convex relaxation.

The homogeneous relaxation can capture any monotone submodular function [OB12]. Since Ω_∞ is the Lovász extension in this case, and hence, $F_-(A) = \Omega_\infty(\mathbb{1}_A) = f_L(\mathbb{1}_A) = F(A)$. Also, since the two ℓ_∞-relaxations are identical for this class of functions (see Remark 3), their LCEs are also equal, i.e., $\tilde{F}_-(A) = \Theta_\infty(\mathbb{1}_A) = \Omega_\infty(\mathbb{1}_A) = F(A)$.

The LCEs, however, are not equal in general. In fact, the non-homogeneous relaxation is tight for a larger class of functions. In particular, the following proposition shows that $\tilde{F}_-$ is equal to the *monotonization* of F, that is $\tilde{F}_-(A) = \inf_{S\subseteq V}\{F(S) : A \subseteq S\}$, for all set functions F, and is thus equal to the function itself if F is monotone.

Proposition 7. *The non-homogenous lower combinatorial envelope can be written as*

$$\begin{aligned}\tilde{F}_-(A) &= \Theta_\infty(\mathbb{1}_A)\\ &= \inf_{\alpha_S\in\{0,1\}}\{\sum_{S\subseteq V}\alpha_S F(S) : \sum_{S\subseteq V}\alpha_S \mathbb{1}_S \geq \mathbb{1}_A, \sum_{S\subseteq V}\alpha_S = 1\}\\ &= \inf_{S\subseteq V}\{F(S) : A \subseteq S\}.\end{aligned}$$

Proof. To see why we can restrict α_S to be integral, let $\mathcal{E} = \{S : \alpha_S > 0\}$, then $\forall T \subseteq V$ such that $\exists e \in A, e \notin T$, then $\sum_{\alpha_S>0, S\neq T}\alpha_S = 1$ and hence $\alpha_T = 0$. Hence $\forall S \in \mathcal{E}$ we have $A \subseteq S$ and $\sum_{\alpha_S>0}\alpha_S F(S) = \min_{\alpha_S>0} F(S)$. □

Proposition 7 argues that the non-homogeneous convex envelope is tight if and only if F is monotone. Two important practical implications follow from this result.

Given a target model that cannot be expressed by a monotone function, it is impossible to obtain a tight convex relaxation.

Example 6 (Tree ℓ_0-penalty)**.** *Given a directed tree $\mathcal{T}$, consider the tree ℓ_0-penalty presented in Chapter 2: $F(S) := |S| + \iota_{\boldsymbol{T}\mathbb{1}_S\geq 0}(S)$, where $\boldsymbol{T}$ is the edge-node incidence matrix of $\mathcal{T}$ (see Section 2.5.2). This penalty enforces the selection of sparse rooted connected subtrees as supports. $F_{T,0}$ is* not monotone*, hence no convex relaxation can preserve it entirely. Indeed, recall from Section 2.5.2 that $\Theta_\infty(\boldsymbol{x}) = \sum_{G_i\in\mathfrak{G}_H}\|\boldsymbol{x}_{G_i}\|_\infty$ (hierarchical group Lasso), where $\mathfrak{G}_H$ is the collection of groups consisting of each node and all its descendants in $\mathcal{T}$. Hence,*

$\tilde{F}_-(A) = \Theta_\infty(\mathbb{1}_A) = |\{i : G_i \cap A\}|$, *which is a submodular function, implying also that* $F_-(A) = \tilde{F}_-(A)$. *Thus, in this case, convex relaxations can capture a more relaxed penalty encouraging the tree structure, but not the non-monotone hard constraints.*

More problematic examples are given by models expressed by *symmetric* functions, i.e., $F(S) = F(S^c), \forall S \subseteq V$. In this case, $\tilde{F}_-(A) = F(V) = F(\emptyset) = 0$; thus the structure is completely lost by convex relaxations. For such models, non-convex methods can be potentially better, as demonstrated in Chapter 5.

On the other hand, if the model can be expressed by a monotone non-submodular set function, the homogeneous relaxation may not be tight, and hence, the non-homogeneous relaxation can be more useful. For instance, [OB12] shows that for any set function where $F(\{e\}) = 1$ for all singletons $e \in V$ and $F(A) \geq |A|, \forall A \subseteq V$, the homogeneous LCE $F_-(A) = |A|$ and accordingly Ω_p is the ℓ_1-norm, thus losing completely the specific structure encoded in F.

We discuss some examples that fall in this class of functions, where the non-homogeneous relaxation is tight while the homogeneous one is not.

Example 7 (Range penalty)**.** *Consider range*$(A) = \max(A) - \min(A) + 1$. *For* $F(A) =$ *range*(A), *we have* $F_-(A) = |A|$, *while* $\tilde{F}_- = F$ *by Prop. 7.*

Example 8 (Down-monotone sparse structures)**.** *A natural class of structured sparsity penalties are penalties of the form* $F(A) = |A| + \iota_{A \in \mathcal{M}}(A)$, *which favor sparse non-zero patterns among a set* $\mathcal{M}$ *of allowed patterns. If* $\mathcal{M}$ *is down-monotone, i.e.,* $\forall A \in \mathcal{M}, \forall B \subseteq A, B \in \mathcal{M}$, *then the non-homogeneous relaxation preserves its structure, i.e.,* $\tilde{F}_- = F$ *by Prop. 7, while its homogeneous relaxation is oblivious to the hard constraints, with* $F_-(A) = |A|$. *This class includes for example the following models:*

- ***Sparse set cover:*** *Given a collection of predefined groups* $\mathfrak{G} = \{G_1, \cdots, G_M\}$, *consider the sparse* g*-group cover penalty, introduced in Chapter 2:* $F(A) = |A| + \iota_{\boldsymbol{B}^T\boldsymbol{\omega} \geq \mathbb{1}_A, \mathbb{1}^T\boldsymbol{\omega} \leq g}(A)$, *where the columns of* $\boldsymbol{B}$ *correspond to the indicator vectors of the groups, i.e.,* $\boldsymbol{B}_{V,i} = \mathbb{1}_{G_i}$ *(see Section 2.5.1). This penalty enforces the selection of sparse supports that can be covered with* g*-groups.*

- ***Dispersive*** ℓ_0***-penalty:*** *Similarly given* $\mathfrak{G} = \{G_1, \cdots, G_M\}$, *consider the dispersive* ℓ_0*-penalty, introduced in Chapter 2:* $F(A) = |A| + \iota_{\boldsymbol{B}^T\mathbb{1}_A \leq \mathbb{1}}(A)$, *where* $\boldsymbol{B}$ *is defined as in the previous example (see Section 2.5.3). The dispersive penalty enforces the selection of sparse supports where no two non-zeros are selected from the same group. Neural sparsity models induce such structures [HDC09].*

- ***Weighted graph model:*** *Given a graph* $\mathcal{G} = (V, E)$, *consider a relaxed version of the weighted graph model of [HIS15b]:* $F(A) = |A| + \iota_{\gamma(F_A) \leq g, w(F_A) \leq b}(A)$, *where* $\gamma(F_A)$ *is the number of connected components formed by the forest* F_A *corresponding to* A *and*

$w(F_A)$ is the total weight of edges in the forest F_A. This model describes a wide range of structures, including 1D-clustering, tree hierarchies, and the Earth Mover Distance model.

3.3 Sparsity-inducing properties of convex relaxations

The notion of LCE captures the combinatorial structure preserved by convex relaxations in a geometric sense. In this section, we characterize the preserved structure from a statistical perspective.

To this end, we consider the linear regression model $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}^\natural + \boldsymbol{\varepsilon}$, where $\boldsymbol{A} \in \mathbb{R}^{n\times d}$ is a fixed design matrix, $\boldsymbol{y} \in \mathbb{R}^n$ is the response vector, and $\boldsymbol{\varepsilon}$ is a vector of iid random variables with mean 0 and variance σ^2. Given $\lambda_n > 0$, we define $\boldsymbol{x}^\star$ as a minimizer of the regularized least-squares:

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda_n \Phi(\boldsymbol{x}), \tag{3.5}$$

We are interested in the sparsity-inducing properties of Ω_p and Θ_p on the solutions of (3.5). In this section, we consider though the more general setting where Φ is any proper normalized ($\Phi(0) = 0$) convex function which is absolute, i.e., $\Phi(\boldsymbol{x}) = \Phi(|\boldsymbol{x}|)$ and monotonic in the absolute values of $\boldsymbol{x}$, that is $|\boldsymbol{x}| \geq |\boldsymbol{x}'| \Rightarrow \Phi(\boldsymbol{x}) \geq \Phi(\boldsymbol{x}')$. In what follows, monotone functions refer to this notion of monotonicity.

We determine in Section 3.3.1 necessary conditions for support recovery in (3.5) and in Section 3.3.2 we provide sufficient conditions for support recovery and consistency of a variant of (3.5). As both Ω_p and Θ_p are normalized absolute monotone convex functions, the results presented in this section apply directly to them as a corollary.

For simplicity, we assume $\boldsymbol{Q} = \boldsymbol{A}^\top\boldsymbol{A}/n \succ 0$, thus $\boldsymbol{x}^\star$ is unique. This forbids the high-dimensional setting. We expect though the insights developed towards the presented results to contribute to understanding the high-dimensional learning setting, which we defer to a later work.

3.3.1 Continuous stable supports

Existing results on the consistency of special cases of the estimator (3.5) typically rely heavily on decomposition properties of Φ [NRWY11, Bac10a, OJV11, OB12]. The notions of decomposability assumed in these prior works are either too strong or too specific to be applicable to the general convex penalties Ω_p and Θ_p we are considering. Instead, we introduce a general weak notion of decomposability applicable to any absolute monotone convex regularizer.

Definition 10 (Decomposability)**.** *Given $J \subseteq V$ and $\boldsymbol{x} \in \mathbb{R}^d$, $\operatorname{supp}(\boldsymbol{x}) \subseteq J$, we say that Φ is* decomposable *at $\boldsymbol{x}$ w.r.t J if $\exists M_J > 0$ such that $\forall \Delta \in \mathbb{R}^d, \operatorname{supp}(\Delta) \subseteq J^c$,*

$$\Phi(\boldsymbol{x} + \Delta) \geq \Phi(\boldsymbol{x}) + M_J\|\Delta\|_\infty.$$

Figure 3.3: Unit balls of ℓ_0-pseudo-norm, restricted to the unit ℓ_∞-ball (left), $\ell_{0.5}$-quasi-norm (middle), and ℓ_1-norm (right).

For example, for the ℓ_1-norm, this decomposability property holds for any $J \subseteq V$ and $\boldsymbol{x} \in \mathbb{R}^d$, with $M_J = 1$.

It is reasonable to expect this property to hold at the solution $\boldsymbol{x}^\star$ of (3.5) and its support $J^\star = \operatorname{supp}(\boldsymbol{x}^\star)$. Theorem 2 shows that this is indeed the case. In Section 3.3.2, we devise an estimation scheme able to recover supports J that satisfy this property at *any* $\boldsymbol{x} \in \mathbb{R}^d$. This leads then to the following notion of *continuous* stable supports, which characterizes supports with respect to the continuous penalty Φ. In Section 3.4, we relate this to the notion of *discrete* stable supports, which characterizes supports with respect to the combinatorial penalty F.

Definition 11 (Continuous stability)**.** *We say that $J \subseteq V$ is* weakly stable *w.r.t Φ if* there exists $\boldsymbol{x} \in \mathbb{R}^d$, $\operatorname{supp}(\boldsymbol{x}) = J$ *such that Φ is* decomposable *at $\boldsymbol{x}$ w.r.t J. Furthermore, we say that $J \subseteq V$ is* strongly stable *w.r.t Φ if* for all $\boldsymbol{x} \in \mathbb{R}^d$ *s.t.* $\operatorname{supp}(\boldsymbol{x}) \subseteq J$, *$\Phi$ is* decomposable *at $\boldsymbol{x}$ w.r.t J.*

Theorem 2 considers slightly more general estimators than (3.5) and shows that weak stability is a necessary condition for a non-zero pattern to be allowed as a solution.

Theorem 2 (Necessary conditions[1])**.** *The minimizer $\boldsymbol{x}^\star$ of $\min_{\boldsymbol{x}\in\mathbb{R}^d} L(\boldsymbol{x}) - \boldsymbol{z}^\top \boldsymbol{x} + \lambda\Phi(\boldsymbol{x})$, where L is a strongly-convex and smooth loss function and $\boldsymbol{z} \in \mathbb{R}^d$ has a continuous density w.r.t to the Lebesgue measure, has a* weakly stable *support w.r.t. Φ, with probability one.*

This new result extends and simplifies the result in [Bac10a] which considers the special case of quadratic loss functions and Φ being the ℓ_∞-convex relaxation of a submodular function. The proof we present, in the Appendix, is also shorter and simpler.

Corollary 7. *Assume $\boldsymbol{y} \in \mathbb{R}^d$ has a continuous density w.r.t to the Lebesgue measure, then the support of the minimizer $\boldsymbol{x}^\star$ of Eq.* (3.5) *is weakly stable w.r.t Φ with probability one.*

3.3.2 Adaptive estimation

Restricting the choice of regularizers in (3.5) to convex relaxations as surrogates to combinatorial penalties is motivated by computational tractability concerns. However, other non-convex regularizers such as ℓ_α-quasi-norms [KF00, FF93] or more generally penalties of the form $\Phi(\boldsymbol{x}) = \sum_{i=1}^d \phi(|\boldsymbol{x}_i|)$, where ϕ is a monotone concave penalty [FL01, DDFG10, GRC09] can be more advantageous than the convex ℓ_1-norm. Such penalties are closer to the ℓ_0-pseudo-norm and penalize more aggressively small coefficients, thus they have a stronger sparsity-inducing effect than ℓ_1-norm (see Figure 3.3).

The authors in [JOB10] extended such concave penalties to the ℓ_α/ℓ_2- quasi-norm $\Phi(\boldsymbol{x}) = \sum_{i=1}^M \|\boldsymbol{x}_{G_i}\|_\alpha$, where $\alpha \in (0,1)$, which enforces sparsity at the group level more aggressively. We generalize this to $\Phi(|\boldsymbol{x}|^\alpha)$ where Φ is any structured sparsity-inducing monotone convex regularizer. These non-convex penalties lead to intractable estimation problems, but approximate solutions can be obtained by majorization-minimization algorithms, as suggested for e.g., in [FBDN07, ZL08, CWB08].

Lemma 4. *Let Φ be a monotone convex function, $\Phi(|\boldsymbol{x}|^\alpha)$ admits the following majorizer, $\forall \boldsymbol{x}^0 \in \mathbb{R}^d$, $\Phi(|\boldsymbol{x}|^\alpha) \leq (1-\alpha)\Phi(|\boldsymbol{x}^0|^\alpha) + \alpha\Phi(|\boldsymbol{x}^0|^{\alpha-1} \circ |\boldsymbol{x}|)$, which is tight at $\boldsymbol{x}^0$.*

We consider the adaptive weight estimator (3.6) resulting from applying a 1-step majorization-minimization to (3.5),

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda_n \Phi(|\boldsymbol{x}^0|^{\alpha-1} \circ |\boldsymbol{x}|), \tag{3.6}$$

where $\boldsymbol{x}^0$ is a $\sqrt{n}$-consistent estimator to $\boldsymbol{x}^\natural$, that is converging to $\boldsymbol{x}^\natural$ at rate $1/\sqrt{n}$ (typically obtained from $\boldsymbol{x}^0 = \mathbb{1}$ or ordinary least-squares).

We study sufficient support recovery and estimation consistency conditions for (3.6) for general convex monotone regularizers Φ. Such consistency results have been established for (3.6), in the classical asymptotic setting, only in the special case of ℓ_1-norm in [Zou06] and for the (non-adaptive) estimator (3.5) for homogeneous convex envelopes of monotone submodular functions, for $p = \infty$ in [Bac10a] and for general p in [OB12], in the high dimensional setting, and for latent group Lasso norm in [OJV11], in the asymptotic setting.

Compared to prior works, the discussion of support recovery is complicated here by the fact that Φ is not necessarily a norm (e.g., if $\Phi = \Theta_p$) and only satisfies a weak notion of decomposability.

As in [Zou06], we consider the classical asymptotic regime in which the model generating the data is of fixed finite dimension d while $n \to \infty$. As before, we assume $\boldsymbol{Q} \succ 0$ and thus the minimizer of (3.6) is unique, we denote it by $\boldsymbol{x}^\star$.

The following Theorem extends the results of [Zou06] for the ℓ_1-norm to any normalized absolute

[1] This theorem and its proof are due primarily to F. Bach.

monotone convex regularizer if the true support satisfy the sufficient condition of strong stability in Definition 11. As we previously remarked this condition is trivially satisfied for the ℓ_1-norm.

Theorem 3 (Consistency and Support Recovery)**.** *Let $\Phi : \mathbb{R}^d \to \overline{\mathbb{R}}_+$ be a proper normalized absolute monotone convex function and denote by J the true support $J = \operatorname{supp}(\boldsymbol{x}^\natural)$. If $|\boldsymbol{x}^\natural|^\alpha \in \operatorname{int} \operatorname{dom} \Phi$, J is* strongly stable *with respect to Φ and λ_n satisfies $\frac{\lambda_n}{\sqrt{n}} \to 0$, $\frac{\lambda_n}{n^{\alpha/2}} \to \infty$, then the estimator* (3.6) *is consistent and asymptotically normal, i.e., it satisfies*

$$\sqrt{n}(\boldsymbol{x}^\star_J - \boldsymbol{x}^\natural_J) \xrightarrow{d} \mathcal{N}(0, \sigma^2 \boldsymbol{Q}_{JJ}^{-1}), \tag{3.7}$$

and

$$P(\operatorname{supp}(\boldsymbol{x}^\star) = J) \to 1. \tag{3.8}$$

Consistency results in most existing works are established under various necessary conditions on $\boldsymbol{A}$, some of which are difficult to verify in practice, such as the *irrepresentability condition* (c.f., [Zou06, Bac10a, OJV11, OB12]). Adding data-dependent weights does not require such conditions and allows recovery even in the correlated measurement matrix setup as illustrated in our numerical results (c.f., Sect. 3.5).

3.4 Sparsity-inducing properties of combinatorial penalties

In Section 3.3, we derived necessary and sufficient conditions for support recovery defined with respect to the continuous convex penalties Ω_p and Θ_p. In this Section, we translate these to conditions with respect to the combinatorial penalties F_p themselves. Hence, the results of this section allows one to check which supports to expect to recover, without the need to compute the corresponding convex relaxation. To that end, we introduce in Section 3.4.1 discrete counterparts of weak and strong stability, and show in Section 3.4.2 that discrete strong stability is a sufficient, and in some cases necessary, condition for support recovery.

3.4.1 Discrete stable supports

We recall the concept of discrete stable sets [Bac10a], also referred to as *flat* or *closed* sets [KG12]. We refer to such sets as discrete weakly stable sets and introduce a stronger notion of discrete stability.

Definition 12 (Discrete stability)**.** *Given a monotone set function $F : 2^V \to \overline{\mathbb{R}}_+$, a set $J \subseteq V$ is said to be* weakly stable *w.r.t F if $\forall i \in J^c, F(J \cup \{i\}) > F(J)$.*
A set $J \subseteq V$ is said to be strongly stable *w.r.t F if $\forall A \subseteq J, \forall i \in J^c, F(A \cup \{i\}) > F(A)$.*

Note that discrete stability imply in particular feasibility, i.e., $F(J) < +\infty$. Also, if F is a strictly monotone function, such as the cardinality function, then all supports are stable w.r.t F. It is interesting to note that for monotone submodular functions, weak and strong stability

are equivalent. In fact, this equivalence holds for a more general class of functions, called ρ-submodular.

Definition 13. *A function $F : 2^V \to \mathbb{R}$ is ρ-submodular iff $\exists \rho \in (0, 1]$ s.t., $\forall B \subseteq V, A \subseteq B, i \in B^c$*

$$\rho[F(B \cup \{i\}) - F(B)] \leq F(A \cup \{i\}) - F(A)$$

The notion of ρ-submodularity was introduced in [LLN06, Definition 15]. It is a special case of the weakly DR-submodular property defined for continuous functions [HSK17]. It is also related to the notion of weak submodularity (c.f., [DK11, EKDN16]). We show in the appendix that ρ-submodularity is a stronger condition than weak submodularity.

Proposition 8. *If F is a finite-valued monotone function, F is ρ-submodular iff discrete weak stability is equivalent to strong stability.*

Example 9. *The range function range$(A) = \max(A) - \min(A) + 1$ is ρ-submodular with $\rho = \frac{1}{d-1}$.*

3.4.2 Relation between discrete and continuous stability

This section provides several technical results relating the discrete and continuous notions of stability. It thus provides us with the necessary tools to characterize which supports can be correctly estimated w.r.t the combinatorial penalty itself, without going through its relaxations.

Proposition 9. *Given any monotone set function F, all sets $J \subseteq V$ strongly stable w.r.t to F are also strongly stable w.r.t Ω_p and Θ_p.*

It follows then by Theorem 3 that discrete strong stability is a sufficient condition for correct estimation.

Corollary 8 (Sufficient condition)**.** *If Φ is equal to Ω_p or Θ_p for $p \in (1, \infty)$ and $\mathrm{supp}(\boldsymbol{x}^\natural) = J$ is* strongly stable *w.r.t F, then Theorem 3 holds, i.e., the adaptive estimator* (3.6) *is consistent and correctly recovers the support. This also holds for $p = \infty$ if we further assume that $\|\boldsymbol{x}^\natural\|_\infty < 1$.*

Furthermore, if F is ρ-submodular, then by Proposition 8, it is enough for $\mathrm{supp}(\boldsymbol{x}^\natural) = J$ to be weakly stable w.r.t F for Corollary 8 to hold. Conversely, Proposition 10 below shows that discrete strong stability is also a necessary condition for continuous strong stability, in the case where $p = \infty$ and F is equal to its LCE.

Proposition 10. *If $F = F_-$ and J is strongly stable w.r.t Ω_∞, then J is strongly stable w.r.t F. Similarly, for any monotone F, if J is strongly stable w.r.t Θ_∞, then J is strongly stable w.r.t F.*

Finally, in the special case of monotone submodular function, the following Corollary 9, along with Proposition 9 demonstrates that all definitions of stability become equivalent. We thus recover the result in [Bac10a] showing that discrete weakly stable supports correspond to the set of allowed non-zero patterns for monotone submodular functions.

Corollary 9. *If F is monotone submodular and J is weakly stable w.r.t $\Omega_\infty = \Theta_\infty$ then J is weakly stable w.r.t F.*

3.4.3 Examples

We highlight in this section what are the supports recovered by the adaptive estimator (AE) (3.6) with the homogeneous convex relaxation Ω_p and non-homogeneous convex relaxation Θ_p of some examples of structure priors. For simplicity, we will focus on the case $p = \infty$. Also in all the examples we consider below, weak and strong discrete stability are equivalent, so we omit the weak/strong specification. Note that it is desirable that the regularizer used enforces the recovery of *only* the non-zero patterns satisfying the desired structure.

Monotone submodular functions: As discussed above, for this class of functions, all stability definitions are equivalent and $\Omega_\infty = \Theta_\infty = f_L$. As a result, AE recovers any discrete stable non-zero pattern. This includes the following examples (c.f., [OB12] for further examples).

- **Cardinality:** $F(A) = |A|$. As a strictly monotone function, all supports are stable w.r.t to it. Thus AE recovers *all* non-zero patterns with Ω_∞ and Θ_∞, given by the ℓ_1-norm.

- **Overlap count function:** $F_\cap(A) = \sum_{G \in \mathfrak{G}, G \cap A \neq \emptyset} d_G$ where $\mathfrak{G}$ is a collection of predefined groups G and d_G their associated weights. Ω_∞ and Θ_∞ are given by the ℓ_1/ℓ_∞-norm (see Section 2.5.1), and stable patterns are complements of union of groups. For example, for hierarchical groups (i.e., groups consisting of each node and its descendants on a tree), AE recovers rooted connected tree supports.

- **Modified range function:** The range function can be transformed into a submodular function, if scaled by a constant as suggested in [Bac10a], yielding the monotone submodular function $F^{\text{mr}}(A) = d - 1 + \text{range}(A), \forall A \neq \emptyset$ and $F^{\text{mr}}(\varnothing) = 0$. This can actually be written as an instance of $F_\cap$ with groups defined as $\mathfrak{G} = \{[1, k] : 1 \leq k \leq d\} \cup \{[k, d] : 1 \leq k \leq d\}$ (see Figure 1.5, left). This norm was proposed to induce interval patterns by [JAB11], and indeed its stable patterns are interval supports. We will compare this function in the experiments with the direct convex relaxations of the range function.

Range function: The range function is $\frac{1}{d-1}$-submodular, thus its discrete strongly and weakly stable supports are identical and they correspond to interval supports. As a result, AE recovers interval supports with Θ_∞. On the other hand, since the homogeneous LCE of the range function is the cardinality, AE recovers all supports with Ω_∞.

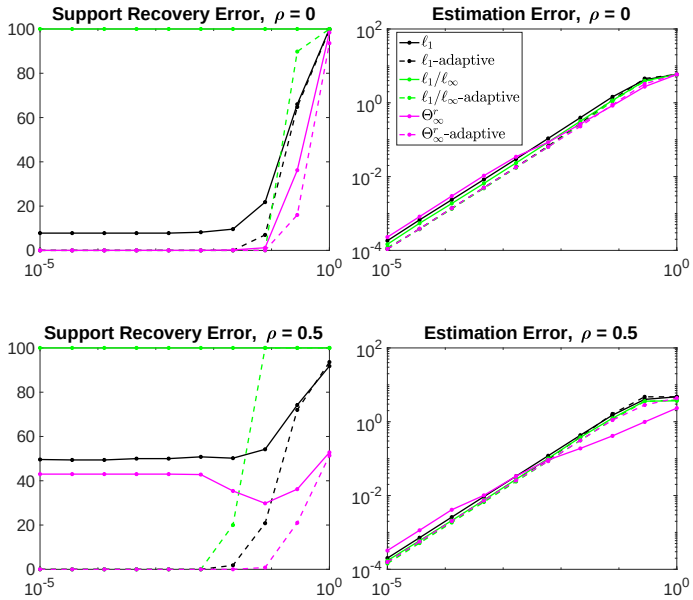


Figure 3.4: (Left column) Best Hamming distance and (Right column) best least square error to the true vector $\boldsymbol{x}^\natural$, along the regularization path, averaged over 5 runs.

Down monotone sparse structures: Functions of the form $F(A) = |A| + \iota_{A\in\mathcal{M}}(A)$, where $\mathcal{M}$ is down-monotone, also have their discrete strongly and weakly stable supports identical and given by the feasible set $\mathcal{M}$ (see Example 8). Since their homogeneous LCE is also the cardinality, then AE recovers all supports with Ω_∞, and only feasible supports with Θ_∞.

3.5 Experiments

To illustrate the results presented in this chapter, we consider the problem of estimating the support of a parameter vector $\boldsymbol{x}^\natural \in \mathbb{R}^d$ whose support is an interval. It is natural then to choose as combinatorial penalty the range function whose stable supports are intervals. We aim to study the effect of adaptive weights, as well as the effect of the choice of homogeneous vs. non-homogeneous convex relaxation for regularization, on the quality of support recovery.

As discussed in Section 3.4.3, the ℓ_∞-homogeneous convex envelope of the range is simply the ℓ_1-norm. Its ℓ_∞-non-homogeneous convex envelope Θ^r_∞ can be computed using the variational form (3.3), where only interval sets need to be considered in the constraints, leading to a quadratic number of constraints. We also consider the ℓ_1/ℓ_∞-norm that corresponds to the convex relaxation of the modified range function F^{mr}.

We consider a simple regression setting in which $\boldsymbol{x}^\natural \in \mathbb{R}^d$ is a constant s-sparse signal whose support is an interval. The choice of $p = \infty$ is well suited for constant valued signals. The design matrix $\boldsymbol{A} \in \mathbb{R}^{d\times n}$ is either drawn as (1) an i.i.d Gaussian matrix with normalized columns, or (2) a correlated Gaussian matrix with normalized columns, with the off-diagonal values of the

covariance matrix set to a value $\rho = 0.5$. We observe noisy linear measurements $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}^\natural + \boldsymbol{\varepsilon}$, where the noise vector is i.i.d. with variance σ^2, where σ is varied between 10^{-5} and 1. We solve problem (3.6) with and without adaptive weights $|\boldsymbol{x}^0|^{\alpha-1}$, where $\boldsymbol{x}^0$ is taken to be the least squares solution and $\alpha = 0.3$.

We assess the estimators obtained through the different regularizers both in terms of support recovery and in terms of estimation error. Figure 3.4 plots (in log scale) these two criteria against the noise level σ. We plot the best achieved error on the regularization path, where the regularization parameter λ was varied between 10^{-6} and 10^3. We set the parameters to $d = 250, s = 100, n = 500$.

We observe that the adaptive weight scheme helps in support recovery, especially in the correlated design setting. Indeed, Lasso is only guaranteed to recover the support under an "irrepresentability condition" [Zou06]. This is satisfied with high probability only in the non-correlated design case. On the other hand, adaptive weights allow us to recover any strongly stable support, without any additional condition, as shown in Theorem 3. The ℓ_1/ℓ_∞-norm performs poorly in this setup. In fact, the modified range function F^{mr}, introduced a gap of d between non-empty sets and the empty set. This leads to the undesirable behavior, already documented in [Bac10a, JAB11] of adding all the variables in one step, as opposed to gradually. Adaptive weights seem to correct for this effect, as seen by the significant improvement in performance. Finally, note that choosing the "tighter" non-homogeneous convex relaxation leads to better support recovery. Indeed, Θ^r_∞ performs better than ℓ_1-norm in all setups.

3.6 Discussion

We presented an analysis of homogeneous and non-homogeneous convex relaxations of ℓ_p-regularized combinatorial penalties. Our results show that structure encoded by monotone submodular priors can be equally well expressed by both relaxations, while the non-homogeneous relaxation is able to express better the structure of more general monotone set functions. We also identified necessary and sufficient stability conditions on the supports to be correctly recovered. We proposed an adaptive weight scheme that is guaranteed to recover supports that satisfy the sufficient stability conditions, in the asymptotic setting, even under correlated design matrix.

We expect the theoretical insights developed in this chapter to help guide the design of convex structure-inducing penalties in the future. In particular, our hope is that our results will motivate further applications of the non-homogeneous convex envelope, which was so far relatively less well-studied, as a tool for structured sparse learning.

The discussion in this chapter raises the following open questions:

Open question 3. In structured sparse learning, we are often interested in the high-dimensional setting, where $n < d$. A natural direction for future work is then to *extend our statistical analysis of the adaptive weight estimator* (3.6) *to the high-dimensional setting,* both in terms of necessary

conditions (Theorem 2) and sufficient conditions (Theorem 3) for support recovery.

Open question 4. In this chapter, we showed that the non-homogeneous convex envelope is tight for monotone functions, as characterized by the notion of lower combinatorial envelope, i.e., $\tilde{F}_{-} = F$ iff F is monotone (see Prop. 7). A natural question then arises: *What are the set functions for which the homogeneous relaxation is tight?*

We already know that a sufficient condition is for F to be monotone submodular, i.e., $F_{-} = F$ if F is monotone submodular. However, submodularity is not *necessary*. We can easily see this from the exclusive Lasso example, where $F(A) = \max_{G \in \mathfrak{G}} |A \cap G|$ is not submodular, but its homogeneous relaxation (which result in the exclusive Lasso norm) is tight (see [OB12, p. 14]). On the other hand, a clear necessary condition is for F to be subadditive, as F_{-} itself is always subadditive. It remains to close this gap by identifying a necessary and sufficient condition on F that ensure its homogeneous relaxation is tight. Such characterization would allow us to identify for which functions, beyond submodular ones, homogeneity can be imposed without sacrificing the combinatorial structure.

Open question 5. In Theorem 2, we showed that *weak continuous stability* is a necessary condition for support recovery using the general estimator (3.5). In the case of monotone submodular functions, this translates to *weak discrete stability* being a necessary condition w.r.t F. *Is weak discrete stability also a necessary condition for support recovery in general?*

On the other hand, in Theorem 3 and Corollary 8, we showed that *strong discrete stability* is a sufficient condition for support recovery using the adaptive estimator (3.6). However, in some cases this condition is too restrictive. For example, for the penalty $F(A) = \max_{G \in \mathfrak{G}} |A \cap G|$, which is used to encourage dispersive supports (see Section 2.5.3), discrete weakly stable supports are supports that have equal number of non-zeros in each group, while discrete strongly stable supports are the trivial empty and ground set. *Is it possible to guarantee consistency and support recovery, with the adaptive estimator or another estimator, under a weaker sufficient condition?*

3.7 Appendix: Proofs

3.7.1 Variational forms of convex envelopes (Proof of Lemma 3)

In this section, we recall the different variational forms of the homogeneous convex envelope derived in [OB12] and derive similar variational forms for the non-homogeneous convex envelope, which includes the ones stated in Lemma 3. These variational forms will be needed in some of our proofs below.

Lemma 5. *The homogeneous convex envelope Ω_p of F_p admits the following variational forms.*

$$\Omega_\infty(\boldsymbol{x}) = \min_{\boldsymbol{\alpha}}\{\sum_{S\subseteq V}\alpha_S F(S) : \sum_{S\subseteq V}\alpha_S \mathbb{1}_S \geq |\boldsymbol{x}|, \alpha_S \geq 0\}. \tag{3.9}$$

$$\Omega_p(\boldsymbol{x}) = \min_{\boldsymbol{v}}\{\sum_{S\subseteq V} F(S)^{1/q}\|\boldsymbol{v}^S\|_p : \sum_{S\subseteq V}\boldsymbol{v}^S = |\boldsymbol{x}|, \mathrm{supp}(\boldsymbol{v}^S)\subseteq S\}. \tag{3.10}$$

$$= \max_{\boldsymbol{\kappa}\in\mathbb{R}^d_+}\sum_{i=1}^d \kappa_i^{1/q}|x_i| \text{ s.t. } \boldsymbol{\kappa}(A)\leq F(A), \forall A\subseteq V. \tag{3.11}$$

$$= \inf_{\boldsymbol{\eta}\in\mathbb{R}^d_+}\frac{1}{p}\sum_{j=1}^d\frac{|x_j|^p}{\eta_j^{p-1}} + \frac{1}{q}\Omega_\infty(\boldsymbol{\eta}). \tag{3.12}$$

Lemma 6. *The non-homogeneous convex envelope Θ_p of F_p admits the following variational forms.*

$$\Theta_\infty(\boldsymbol{x}) = \inf_{\boldsymbol{\alpha}}\{\sum_{S\subseteq V}\alpha_S F(S) : \sum_{S\subseteq V}\alpha_S\mathbb{1}_S \geq |\boldsymbol{x}|, \sum_{S\subseteq V}\alpha_S = 1, \alpha_S\geq 0\}. \tag{3.13}$$

$$= \inf_{\boldsymbol{v}}\{\sum_{S\subseteq V} F(S)\|\boldsymbol{v}^S\|_\infty : \sum_{S\subseteq V} v^S = |x|, \sum_{S\subseteq V}\|\boldsymbol{v}^S\|_\infty = 1, \mathrm{supp}(\boldsymbol{v}^S)\subseteq S\}. \tag{3.14}$$

$$\Theta_p(\boldsymbol{x}) = \max_{\boldsymbol{\kappa}\in\mathbb{R}^d}\sum_{j=1}^d\psi_j(\kappa_j, x_j) + \min_{S\subseteq V} F(S) - \boldsymbol{\kappa}(S),\ \forall \boldsymbol{x}\in\mathrm{dom}(\Theta_p(\boldsymbol{x})). \tag{3.15}$$

$$= \inf_{\boldsymbol{\eta}\in[0,1]^d}\frac{1}{p}\sum_{j=1}^d\frac{|x_j|^p}{\eta_j^{p-1}} + \frac{1}{q}f^-(\boldsymbol{\eta}), \tag{3.16}$$

where f^- is the convex closure of F, and $\mathrm{dom}(\Theta_p) = \{\boldsymbol{x}|\exists\boldsymbol{\eta}\in[0,1]^d \text{ s.t } \mathrm{supp}(\boldsymbol{x})\subseteq\mathrm{supp}(\boldsymbol{\eta}), \boldsymbol{\eta}\in\mathrm{dom}(f^-)\}$*, and where we define*

$$\psi_j(\kappa_j, x_j) := \begin{cases}\kappa_j^{1/q}|x_j| & \text{if } |x_j|\leq\kappa_j^{1/p}, \kappa_j\geq 0\\ \frac{1}{p}|x_j|^p + \frac{1}{q}\kappa_j & \text{otherwise.}\end{cases}$$

If F is monotone, $\Theta_\infty(\boldsymbol{x}) = f^-(|\boldsymbol{x}|)$, then we can replace f^- by Θ_∞ in (3.16) *and we can restrict $\boldsymbol{\kappa}\in\mathbb{R}^d_+$ in* (3.15).

In what follows, we present the proof of each variational form in Lemma 6.

Recall first that in Proposition 1 of Chapter 2, we showed that $\Theta_\infty(\boldsymbol{x}) = \inf_{\boldsymbol{\eta}\in[0,1]^d}\{f^-(\boldsymbol{\eta}) : \boldsymbol{\eta} \geq |\boldsymbol{x}|\}$. Note then that Θ_∞ is monotone even if F is not. On the other hand, if F is monotone, then f^- is monotone on $[0,1]^d$ and $\Theta_\infty(\boldsymbol{x}) = f^-(|\boldsymbol{x}|)$.

The variational form (3.13) follows directly from Proposition 1 and the properties of the convex closure f^-, as shown in the following corollary.

Corollary 10. *Given any set function $F : 2^V \to \overline{\mathbb{R}}_+$ and its corresponding convex closure f^-, the convex envelope of $F(\mathrm{supp}(\boldsymbol{x}))$ over the unit ℓ_∞-ball is given by*

$$\Theta_\infty(\boldsymbol{x}) = \inf_{\alpha}\{\sum_{S\subseteq V}\alpha_S F(S) : \sum_{S\subseteq V}\alpha_S \mathbb{1}_S \geq |\boldsymbol{x}|, \sum_{S\subseteq V}\alpha_S = 1, \alpha_S \geq 0\}.$$

Note that $\mathrm{dom}(\Theta_\infty) = \{\boldsymbol{x} : \exists \boldsymbol{\eta} \in [0,1]^d \cap \mathrm{dom}(f^-), \boldsymbol{\eta} \geq |\boldsymbol{x}|\}$.

Proof. This follows directly by plugging in $\Theta_\infty(\boldsymbol{x}) = \inf_{\boldsymbol{\eta}\in[0,1]^d}\{f^-(\boldsymbol{\eta}) : \boldsymbol{\eta} \geq |\boldsymbol{x}|\}$ the variational form of the convex closure from Definition 19;

$$f^-(\boldsymbol{\eta}) = \inf_{\boldsymbol{\alpha}}\{\sum_{S\subseteq V}\alpha_S F(S) : \boldsymbol{\eta} = \sum_{S\subseteq V}\alpha_S \mathbb{1}_S, \sum_{S\subseteq V}\alpha_S = 1, \alpha_S \geq 0\}.$$

□

Next, we derive the convex envelope of F_p for a general $p \geq 1$.

Proposition 11. *Given any set function $F : 2^V \to \overline{\mathbb{R}}_+$ and its corresponding convex closure f^-, the convex envelope of $F_{\mu\lambda}(\boldsymbol{x}) = \mu F(\mathrm{supp}(\boldsymbol{x})) + \lambda\|\boldsymbol{x}\|_p^p$ is given by*

$$\Theta_p(\boldsymbol{x}) = \inf_{\boldsymbol{\eta}\in[0,1]^d} \lambda \sum_{j=1}^{d} \frac{|x_j|^p}{\eta_j^{p-1}} + \mu f^-(\boldsymbol{\eta}).$$

Note that $\mathrm{dom}(\Theta_p) = \{\boldsymbol{x} | \exists \boldsymbol{\eta} \in [0,1]^d \text{ s.t } \mathrm{supp}(\boldsymbol{x}) \subseteq \mathrm{supp}(\boldsymbol{\eta}), \boldsymbol{\eta} \in \mathrm{dom}(f^-)\}$.

Proof. Like in Lemma 1, we assume we are given a proper l.s.c. convex extension $\hat{f}$ of F, which satisfies $\max_{\boldsymbol{\eta}\in\{0,1\}^d} |\boldsymbol{z}|^T\boldsymbol{\eta} - \hat{f}(\boldsymbol{\eta}) = \max_{\boldsymbol{\eta}\in[0,1]^d} |\boldsymbol{z}|^T\boldsymbol{\eta} - \hat{f}(\boldsymbol{\eta}), \forall \boldsymbol{z} \in \mathbb{R}^d$, which is true for $\hat{f} = f^-$. Then, we have first for the case where $p = 1$:

$$\begin{aligned}
F^*_{\mu\lambda}(\boldsymbol{s}) &= \sup_{\boldsymbol{x}\in\mathbb{R}^d} \boldsymbol{x}^T s - \mu F(\mathrm{supp}(\boldsymbol{x})) - \lambda\|\boldsymbol{x}\|_1 \\
&= \sup_{\boldsymbol{\eta}\in\{0,1\}^d} \sup_{\substack{\mathbb{1}_{\mathrm{supp}(\boldsymbol{x})}=\boldsymbol{\eta} \\ \mathrm{sign}(\boldsymbol{x})=\mathrm{sign}(\boldsymbol{s})}} |\boldsymbol{x}|^T(|\boldsymbol{s}| - \lambda\mathbb{1}) - \mu F(\boldsymbol{\eta}) \\
&= \iota_{\{|\boldsymbol{s}|\leq\lambda\mathbb{1}\}}(\boldsymbol{s}) - \inf_{\boldsymbol{\eta}\in\{0,1\}^d} \mu F(\boldsymbol{\eta}).
\end{aligned}$$

Hence, $F^{**}_{\mu\lambda}(\boldsymbol{x}) = \lambda\|\boldsymbol{x}\|_1 + \inf_{\boldsymbol{\eta}\in\{0,1\}^d} \lambda F(\boldsymbol{\eta})$. For the case $p \in (1,\infty)$, we have:

$$\begin{aligned}
F^*_{\mu\lambda}(\boldsymbol{s}) &= \sup_{\boldsymbol{x}\in\mathbb{R}^d} \boldsymbol{x}^T\boldsymbol{s} - \mu F(\mathrm{supp}(\boldsymbol{x})) - \lambda\|\boldsymbol{x}\|_p^p \\
&= \sup_{\boldsymbol{\eta}\in\{0,1\}^d} \sup_{\substack{\mathbb{1}_{\mathrm{supp}(\boldsymbol{x})}=\boldsymbol{\eta} \\ \mathrm{sign}(\boldsymbol{x})=\mathrm{sign}(\boldsymbol{s})}} |\boldsymbol{x}|^T|\boldsymbol{s}| - \lambda\|\boldsymbol{x}\|_p^p - \mu F(\boldsymbol{\eta}) \\
&= \sup_{\boldsymbol{\eta}\in\{0,1\}^d} \frac{\lambda(p-1)}{(\lambda p)^q}\boldsymbol{\eta}^T|\boldsymbol{s}|^q - \mu F(\boldsymbol{\eta}) \qquad (|s_i| = \lambda p|x_i^\star|^{p-1}, \forall \eta_i \neq 0) \\
&= \sup_{\boldsymbol{\eta}\in[0,1]^d} \frac{\lambda(p-1)}{(\lambda p)^q}\boldsymbol{\eta}^T|\boldsymbol{s}|^q - \mu\hat{f}(\boldsymbol{\eta}).
\end{aligned}$$

We denote $\hat{\lambda} = \frac{\lambda(p-1)}{(\lambda p)^q}$.

$$\begin{aligned}
F^{**}_{\mu\lambda}(\boldsymbol{x}) &= \sup_{\boldsymbol{s}\in\mathbb{R}^d} \boldsymbol{x}^T\boldsymbol{s} - F^*_{\mu\lambda}(\boldsymbol{s}) \\
&= \sup_{\boldsymbol{s}\in\mathbb{R}^d} \min_{\boldsymbol{\eta}\in[0,1]^d} \boldsymbol{s}^T\boldsymbol{x} - \hat{\lambda}\boldsymbol{\eta}^T|\boldsymbol{s}|^q + \mu\hat{f}(\boldsymbol{\eta}) \\
&\overset{\star}{=} \inf_{\boldsymbol{\eta}\in[0,1]^d} \sup_{\substack{\boldsymbol{s}\in\mathbb{R}^p \\ \mathrm{sign}(\boldsymbol{s})=\mathrm{sign}(\boldsymbol{x})}} |\boldsymbol{s}|^T|\boldsymbol{x}| - \hat{\lambda}\boldsymbol{\eta}^T|\boldsymbol{s}|^q + \mu\hat{f}(\boldsymbol{\eta}) \\
&= \inf_{\boldsymbol{\eta}\in[0,1]^d} \lambda(|\boldsymbol{x}|^p)^T\boldsymbol{\eta}^{1-p} + \mu\hat{f}(\boldsymbol{\eta}),
\end{aligned}$$

where the last equality holds since $|x_i| = \hat{\lambda}\eta_i q|s_i^\star|^{q-1}, \forall\eta_i \neq 0$, otherwise $s_i^\star = 0$ if $x_i = 0$ and ∞ otherwise. $(\star)$ holds by Sion's minimax theorem [S+58, Corollary 3.3]. Note then that the minimizer $\boldsymbol{\eta}^*$ (if it exists) satisfies $\mathrm{supp}(\boldsymbol{x}) \subseteq \mathrm{supp}(\boldsymbol{\eta}^\star)$. Finally, note that if we take the limit as $p \to \infty$, we recover $\Theta_\infty(\boldsymbol{x}) = \inf_{\boldsymbol{\eta}\in[0,1]^d}\{f^-(\boldsymbol{\eta}) : \boldsymbol{\eta} \geq |\boldsymbol{x}|\}$. □

The variational form (3.16) in lemma 6 follows from proposition 11 for the choice $\mu = \frac{1}{q}, \lambda = \frac{1}{p}$.

The following proposition derives the variational form (3.15) for $p = \infty$.

Proposition 12. *Given any set function $F : 2^V \to \mathbb{R}\cup\{+\infty\}$, and its corresponding convex closure f^-, Θ_∞ can be written $\forall\boldsymbol{x} \in \mathrm{dom}(\Theta_\infty)$ as*

$$\begin{aligned}
\Theta_\infty(\boldsymbol{x}) &= \max_{\boldsymbol{\kappa}\in\mathbb{R}^d_+}\{\boldsymbol{\kappa}^T|\boldsymbol{x}| + \min_{S\subseteq V} F(S) - \boldsymbol{\kappa}(S)\} \\
&= \max_{\boldsymbol{\kappa}\in\mathbb{R}^d_+}\{\boldsymbol{\kappa}^T|\boldsymbol{x}| + \min_{S\subseteq\mathrm{supp}(\boldsymbol{x})} F(S) - \boldsymbol{\kappa}(S)\} \qquad \text{(if } F \text{ is monotone)}
\end{aligned}$$

Similarly $\forall\boldsymbol{x} \in \mathrm{dom}(f^-)\cap[0,1]^d$ we can write

$$\begin{aligned}
f^-(\boldsymbol{x}) &= \max_{\boldsymbol{\kappa}\in\mathbb{R}^d}\{\boldsymbol{\kappa}^T\boldsymbol{x} + \min_{S\subseteq V} F(S) - \boldsymbol{\kappa}(S)\} \\
&= \Theta_\infty(\boldsymbol{x}) = \max_{\boldsymbol{\kappa}\in\mathbb{R}^d_+}\{\boldsymbol{\kappa}^T\boldsymbol{x} + \min_{S\subseteq\mathrm{supp}(\boldsymbol{x})} F(S) - \boldsymbol{\kappa}(S)\} \qquad \text{(if } F \text{ is monotone)}
\end{aligned}$$

Proof. $\forall \boldsymbol{x} \in \mathrm{dom}(\Theta_\infty)$, strong duality holds by Slater's condition, hence

$$\begin{aligned}
\Theta_\infty(\boldsymbol{x}) &= \min_{\alpha}\{\sum_{S\subseteq V}\alpha_S F(S) : \sum_{S\subseteq V}\alpha_S \mathbb{1}_S \geq |\boldsymbol{x}|, \sum_{S\subseteq V}\alpha_S = 1, \alpha_S \geq 0\}. \\
&= \min_{\alpha\geq 0}\max_{\rho\in\mathbb{R},\boldsymbol{\kappa}\in\mathbb{R}^d_+}\{\sum_{S\subseteq V}\alpha_S F(S) + \boldsymbol{\kappa}^T(|\boldsymbol{x}| - \sum_{S\subseteq V}\alpha_S\mathbb{1}_S) + \rho(1-\sum_{S\subseteq V}\alpha_S)\}. \\
&= \max_{\rho\in\mathbb{R},\boldsymbol{\kappa}\in\mathbb{R}^d_+}\min_{\alpha\geq 0}\{\boldsymbol{\kappa}^T|\boldsymbol{x}| + \sum_{S\subseteq V}\alpha_S(F(S) - \boldsymbol{\kappa}^T\mathbb{1}_S - \rho) + \rho\}. \\
&= \max_{\rho\in\mathbb{R},\boldsymbol{\kappa}\in\mathbb{R}^d_+}\{\boldsymbol{\kappa}^T|\boldsymbol{x}| + \rho : F(S) \geq \boldsymbol{\kappa}^T\mathbb{1}_S + \rho)\}. \\
&= \max_{\boldsymbol{\kappa}\in\mathbb{R}^d_+}\{\boldsymbol{\kappa}^T|\boldsymbol{x}| + \min_{S\subseteq V} F(S) - \boldsymbol{\kappa}(S)\}.
\end{aligned}$$

Let $J = \mathrm{supp}(|\boldsymbol{x}|)$ then $\boldsymbol{\kappa}^\star_{J^c} = 0$. Then for monotone functions $F(S) - \boldsymbol{\kappa}^\star(S) \geq F(S\cap J) - \boldsymbol{\kappa}^\star(S)$, so we can restrict the minimum to $S \subseteq J$. The same proof holds for f^-, with the Lagrange multiplier $\boldsymbol{\kappa} \in \mathbb{R}^d$ not constrained to be positive. □

The following Corollary derives the variational form (3.15) for $p \in [1, \infty]$.

Corollary 11. *Given any set function $F : 2^V \to \mathbb{R}\cup\{+\infty\}$, Θ_p can be written $\forall \boldsymbol{x} \in \mathrm{dom}(\Theta_p)$ as*

$$\begin{aligned}
\Theta_p(\boldsymbol{x}) &= \max_{\boldsymbol{\kappa}\in\mathbb{R}^d}\sum_{j=1}^d \psi_j(\kappa_j, x_j) + \min_{S\subseteq V} F(S) - \boldsymbol{\kappa}(S). \\
&= \max_{\boldsymbol{\kappa}\in\mathbb{R}^d_+}\sum_{j=1}^d \psi_j(\kappa_j, x_j) + \min_{S\subseteq V} F(S) - \boldsymbol{\kappa}(S). \qquad \text{(if } F \text{ is monotone)}
\end{aligned}$$

where

$$\psi_j(\kappa_j, x_j) := \begin{cases} \kappa_j^{1/q}|x_j| & \textit{if } |x_j| \leq \kappa_j^{1/p}, \kappa_j \geq 0 \\ \frac{1}{p}|x_j|^p + \frac{1}{q}\kappa_j & \textit{otherwise} \end{cases}$$

Proof. By Propositions 11 and 12, we have $\forall \boldsymbol{x} \in \mathrm{dom}(\Theta_p)$, i.e., $\exists \boldsymbol{\eta} \in [0,1]^d$, s.t $\mathrm{supp}(\boldsymbol{x}) \subseteq \mathrm{supp}(\boldsymbol{\eta}), \boldsymbol{\eta} \in \mathrm{dom}(f^-)$,

$$\begin{aligned}
\Theta_p(\boldsymbol{x}) &= \inf_{\boldsymbol{\eta}\in[0,1]^d}\frac{1}{p}\sum_{j=1}^d\frac{|x_j|^p}{\eta_j^{p-1}} + \frac{1}{q}f^-(\boldsymbol{\eta}) \\
&= \inf_{\boldsymbol{\eta}\in[0,1]^d}\frac{1}{p}\sum_{j=1}^d\frac{|x_j|^p}{\eta_j^{p-1}} + \frac{1}{q}\max_{\rho\in\mathbb{R},\boldsymbol{\kappa}\in\mathbb{R}^d}\{\boldsymbol{\kappa}^T\boldsymbol{\eta} + \rho : F(S) \geq \boldsymbol{\kappa}^T\mathbb{1}_S + \rho\}. \\
&\overset{\star}{=} \max_{\rho\in\mathbb{R},\boldsymbol{\kappa}\in\mathbb{R}^d}\inf_{\boldsymbol{\eta}\in[0,1]^d}\{\frac{1}{p}\sum_{j=1}^d\frac{|x_j|^p}{\eta_j^{p-1}} + \frac{1}{q}\boldsymbol{\kappa}^T\boldsymbol{\eta} + \rho : F(S) \geq \boldsymbol{\kappa}^T\mathbb{1}_S + \rho\}.
\end{aligned}$$

($\star$) holds by Sion's minimax theorem [S$^+$58, Corollary 3.3]. Note also that

$$\inf_{\eta_j \in [0,1]} \frac{1}{p}\frac{|x_j|^p}{\eta_j^{p-1}} + \frac{1}{q}\kappa_j \eta_j = \begin{cases} \kappa_j^{1/q}|x_j| & \text{if } |x_j| \leq \kappa_j^{1/p}, \kappa_j \geq 0 \\ \frac{1}{p}|x_j|^p + \frac{1}{q}\kappa_j & \text{otherwise} \end{cases} := \psi_j(\kappa_j, x_j)$$

where the minimum is $\eta_j^\star = 1$ if $\kappa_j \leq 0$. If $\kappa_j \geq 0$, the infimum is zero if $x_j = 0$. Otherwise, the minimum is achieved at $\eta_j^\star = \min\{\frac{|x_j|}{\kappa_j^{1/p}}, 1\}$ (if $\kappa_j = 0, \eta_j^\star = 1$). Hence,

$$\Theta_p(\boldsymbol{x}) = \max_{\boldsymbol{\kappa} \in \mathbb{R}^d} \sum_{j=1}^{d} \psi_j(\kappa_j, x_j) + \min_{S \subseteq V} F(S) - \boldsymbol{\kappa}(S).$$

□

3.7.2 Proof of Theorem 2

Before proving Theorem 2, we need the following technical Lemma.

Lemma 7. *Given $J \subset V$ and a vector $\boldsymbol{x}$ s.t $\operatorname{supp}(\boldsymbol{x}) \subseteq J$, if Φ is not decomposable at $\boldsymbol{x}$ w.r.t J, then $\exists i \in J^c$ such that the i-th component of all subgradients at $\boldsymbol{x}$ is zero; $0 = [\partial\Phi(\boldsymbol{x})]_i$.*

Proof. If Φ is not decomposable at $\boldsymbol{x}$ and $0 \neq [\partial\Phi(\boldsymbol{x})]_i, \forall i \in J^c$, then $\forall M_J > 0, \exists \Delta \neq 0, \operatorname{supp}(\Delta) \subseteq J^c$ s.t., $\Phi(\boldsymbol{x} + \Delta) < \Phi(\boldsymbol{x}) + M_J\|\Delta\|_\infty$. In particular, we can choose $M_J = \inf_{i \in J^c, \boldsymbol{v} \in \partial\Phi(\boldsymbol{x}_J), v_i \neq 0} |v_i| > 0$, if the inequality holds for some $\Delta \neq 0$, then let $i_{\max}$ denote the index where $|\Delta_{i_{\max}}| = \|\Delta\|_\infty$. Then given any $\boldsymbol{v} \in \partial\Phi(\boldsymbol{x})$ s.t., $v_{i_{\max}} \neq 0$, we have

$$\begin{aligned} \Phi(\boldsymbol{x} + \|\Delta\|_\infty \mathbb{1}_{i_{\max}}) \leq \Phi(\boldsymbol{x} + \Delta) &< \Phi(\boldsymbol{x}) + M_J\|\Delta\|_\infty \\ &\leq \Phi(\boldsymbol{x}) + \langle \boldsymbol{v}, \|\Delta\|_\infty \mathbb{1}_{i_{\max}} \operatorname{sign}(v_{i_{\max}})\rangle \\ &\leq \Phi(\boldsymbol{x} + \|\Delta\|_\infty \mathbb{1}_{i_{\max}}) \end{aligned}$$

which leads to a contradiction. □

Theorem 2 (Necessary conditions[2]). *The minimizer $\boldsymbol{x}^\star$ of $\min_{\boldsymbol{x} \in \mathbb{R}^d} L(\boldsymbol{x}) - \boldsymbol{z}^\top \boldsymbol{x} + \lambda\Phi(\boldsymbol{x})$, where L is a strongly-convex and smooth loss function and $\boldsymbol{z} \in \mathbb{R}^d$ has a continuous density w.r.t to the Lebesgue measure, has a* weakly stable *support w.r.t. Φ, with probability one.*

Proof. We will show in particular that Φ is decomposable at $\boldsymbol{x}^\star$ w.r.t $\operatorname{supp}(\boldsymbol{x}^\star)$. Since L is strongly-convex, given $\boldsymbol{z}$ the corresponding minimizer $\boldsymbol{x}^\star$ is unique, then the function $h(\boldsymbol{z}) :=$

[2]This theorem and its proof are due primarily to F. Bach.

$\arg\min_{\boldsymbol{x}\in\mathbb{R}^d} L(\boldsymbol{x}) - \boldsymbol{z}^T\boldsymbol{x} + \lambda\Phi(\boldsymbol{x})$ is well defined. We want to show that

$$\begin{aligned}
&P(\forall \boldsymbol{z},\ \Phi \text{ is decomposable at } h(\boldsymbol{z}) \text{ w.r.t } \operatorname{supp}(h(\boldsymbol{z}))\ ) \\
&= 1 - P(\exists \boldsymbol{z}, \text{s.t, } \Phi \text{ is not decomposable at } h(\boldsymbol{z}) \text{ w.r.t } \operatorname{supp}(h(\boldsymbol{z}))\ ) \\
&\geq 1 - P(\exists \boldsymbol{z},\ \text{s.t., } \exists i \in (\operatorname{supp}(h(\boldsymbol{z})))^c, [\partial\Phi(h(\boldsymbol{z}))]_i = 0) && \text{by lemma 7} \\
&= 1.
\end{aligned}$$

Given fixed $i \in V$, we show that the set $S_i := \{\boldsymbol{z} : i \in (\operatorname{supp}(h(\boldsymbol{z})))^c, [\partial\Phi(h(\boldsymbol{z}))]_i = 0\}$ has measure zero. Then, taking the union of the finitely many sets $S_i, \forall i \in V$, all of zero measure, we have $P(\exists \boldsymbol{z},\ \text{s.t., } \exists i \in (\operatorname{supp}(h(\boldsymbol{z})))^c, [\partial\Phi(h(\boldsymbol{z}))]_i = 0) = 0$.

To show that the set S_i has measure zero, let $\boldsymbol{z}_1, \boldsymbol{z}_2 \in S_i$ and denote by $\mu > 0$ the strong convexity constant of L. We have by convexity of Φ:

$$\begin{aligned}
\Big(\big(\boldsymbol{z}_1 - \nabla L(h(\boldsymbol{z}_1))\big) - \big(\boldsymbol{z}_2 - \nabla L(h(\boldsymbol{z}_2))\big)\Big)^\top \Big(h(\boldsymbol{z}_1) - h(\boldsymbol{z}_2)\Big) \geq 0 \\
(\boldsymbol{z}_1 - \boldsymbol{z}_2)^\top (h(\boldsymbol{z}_1) - h(\boldsymbol{z}_2)) \geq \big(\nabla L(h(\boldsymbol{z}_1)) - \nabla L(h(\boldsymbol{z}_2))\big)^\top \big(h(\boldsymbol{z}_1) - h(\boldsymbol{z}_2)\big) \\
(\boldsymbol{z}_1 - \boldsymbol{z}_2)^\top (h(\boldsymbol{z}_1) - h(\boldsymbol{z}_2)) \geq \mu \|h(\boldsymbol{z}_1) - h(\boldsymbol{z}_2)\|_2^2 \\
\frac{1}{\mu}\|\boldsymbol{z}_1 - \boldsymbol{z}_2\|_2 \geq \|h(\boldsymbol{z}_1) - h(\boldsymbol{z}_2)\|_2
\end{aligned}$$

Thus h is a deterministic Lipschitz-continuous function of $\boldsymbol{z}$. Let $J = \operatorname{supp}(h(\boldsymbol{z}))$, then by optimality conditions $\boldsymbol{z}_J - \nabla L(h(\boldsymbol{z}_J)) \in \partial\Phi(h(\boldsymbol{z}_J))$ (since $h(\boldsymbol{z}) = h(\boldsymbol{z}_J)$), then $z_i - \nabla L(h(\boldsymbol{z}_J))_i = 0$ since $[\partial\Phi(h(\boldsymbol{z}_J))]_i = 0$. Thus z_i is a Lipschitz-continuous function of $\boldsymbol{z}_J$, which can only happen with zero measure. □

3.7.3 Proof of Lemma 4 and Theorem 3

Lemma 4. *Let Φ be a monotone convex function, $\Phi(|\boldsymbol{x}|^\alpha)$ admits the following majorizer, $\forall \boldsymbol{x}^0 \in \mathbb{R}^d$, $\Phi(|\boldsymbol{x}|^\alpha) \leq (1-\alpha)\Phi(|\boldsymbol{x}^0|^\alpha) + \alpha\Phi(|\boldsymbol{x}^0|^{\alpha-1} \circ |\boldsymbol{x}|)$, which is tight at $\boldsymbol{x}^0$.*

Proof. The function $x \to x^\alpha$ is concave on $\mathbb{R}_+ \setminus \{0\}$, hence

$$\begin{aligned}
|x_j|^\alpha &\leq |x_j^0|^\alpha + \alpha|x_j^0|^{\alpha-1}(|x_j| - |x_j|^0) \\
|x_j|^\alpha &\leq (1-\alpha)|x_j^0|^\alpha + \alpha|x_j^0|^{\alpha-1}|x_j| \\
\Phi(|\boldsymbol{x}|^\alpha) &\leq \Phi((1-\alpha)|\boldsymbol{x}^0|^\alpha + \alpha|\boldsymbol{x}^0|^{\alpha-1} \circ |x_j|) && \text{(by monotonicity)} \\
\Phi(|\boldsymbol{x}|^\alpha) &\leq (1-\alpha)\Phi(|\boldsymbol{x}^0|^\alpha) + \alpha\Phi(|\boldsymbol{x}^0|^{\alpha-1} \circ |\boldsymbol{x}|) && \text{(by convexity)}
\end{aligned}$$

If $x_j = 0$ for any j, the upper bound goes to infinity and hence it still holds. □

Theorem 3 (Consistency and Support Recovery)**.** *Let $\Phi : \mathbb{R}^d \to \overline{\mathbb{R}}_+$ be a proper normalized absolute monotone convex function and denote by J the true support $J = \operatorname{supp}(\boldsymbol{x}^\natural)$. If $|\boldsymbol{x}^\natural|^\alpha \in \operatorname{int}\operatorname{dom}\Phi$, J is* strongly stable *with respect to Φ and λ_n satisfies $\frac{\lambda_n}{\sqrt{n}} \to 0$, $\frac{\lambda_n}{n^{\alpha/2}} \to \infty$, then the estimator* (3.6) *is consistent and asymptotically normal, i.e., it satisfies*

$$\sqrt{n}(\boldsymbol{x}^\star_J - \boldsymbol{x}^\natural_J) \xrightarrow{d} \mathcal{N}(0, \sigma^2 \boldsymbol{Q}_{JJ}^{-1}), \tag{3.7}$$

and

$$P(\operatorname{supp}(\boldsymbol{x}^\star) = J) \to 1. \tag{3.8}$$

Proof. We will follow the proof in [Zou06]. We write $\boldsymbol{x}^\star = \boldsymbol{x}^\natural + \frac{\boldsymbol{u}^\star}{\sqrt{n}}$ and $\Psi_n(\boldsymbol{u}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}(\boldsymbol{x}^\natural + \frac{\boldsymbol{u}}{\sqrt{n}})\|_2^2 + \lambda_n \Phi(\boldsymbol{c} \circ |\boldsymbol{x}^\natural + \frac{\boldsymbol{u}}{\sqrt{n}}|)$, where $\boldsymbol{c} = |\boldsymbol{x}^0|^{\alpha-1}$. Then $\boldsymbol{u}^\star = \arg\min_{\boldsymbol{u}\in\mathbb{R}^d} \Psi_n(\boldsymbol{u})$. Let $V_n(\boldsymbol{u}) = \Psi_n(\boldsymbol{u}) - \Psi_n(0)$, then

$$V_n(\boldsymbol{u}) = \frac{1}{2}\boldsymbol{u}^T\boldsymbol{Q}\boldsymbol{u} - \boldsymbol{\varepsilon}^T\frac{\boldsymbol{A}\boldsymbol{u}}{\sqrt{n}} + \lambda_n\big(\Phi(\boldsymbol{c} \circ |\boldsymbol{x}^\natural + \frac{\boldsymbol{u}}{\sqrt{n}}|) - \Phi(\boldsymbol{c} \circ |\boldsymbol{x}^\natural|)\big)$$

Since x^0 is a $\sqrt{n}$-consistent estimator to $\boldsymbol{x}^\natural$, then $\sqrt{n}x^0_{J^c} = O_p(1)$ and $n^{\frac{1-\alpha}{2}}\boldsymbol{c}_{J^c}^{-1} = O_p(1)$. Since $\frac{\lambda_n}{n^{\alpha/2}} \to \infty$, by stability of J, we have

$$\begin{aligned}
&\lambda_n\big(\Phi(\boldsymbol{c} \circ |\boldsymbol{x}^\natural + \frac{\boldsymbol{u}}{\sqrt{n}}|) - \Phi(\boldsymbol{c} \circ |\boldsymbol{x}^\natural|)\big) \\
&= \lambda_n\big(\Phi(\boldsymbol{c}_J \circ |\boldsymbol{x}^\natural_J + \frac{\boldsymbol{u}_J}{\sqrt{n}}| + \boldsymbol{c}_{J^c} \circ \frac{|\boldsymbol{u}_{J^c}|}{\sqrt{n}}) - \Phi(\boldsymbol{c}_J \circ |\boldsymbol{x}^\natural_J|)\big) \\
&\geq \lambda_n\big(\Phi(\boldsymbol{c}_J \circ |\boldsymbol{x}^\natural_J + \frac{\boldsymbol{u}_J}{\sqrt{n}}|) + M_J\|\boldsymbol{c}_{J^c} \circ \frac{|\boldsymbol{u}_{J^c}|}{\sqrt{n}}\|_\infty - \Phi(\boldsymbol{c}_J \circ |\boldsymbol{x}^\natural_J|)\big) \\
&= \lambda_n\big(\Phi(\boldsymbol{c}_J \circ |\boldsymbol{x}^\natural_J + \frac{\boldsymbol{u}_J}{\sqrt{n}}|) - \Phi(\boldsymbol{c}_J \circ |\boldsymbol{x}^\natural_J|)\big) + M_J\|\lambda_n n^{-\alpha/2} n^{\frac{\alpha-1}{2}}\boldsymbol{c}_{J^c} \circ |\boldsymbol{u}_{J^c}|\|_\infty \\
&\xrightarrow{p} \infty \quad \text{if } \boldsymbol{u}_{J^c} \neq 0
\end{aligned} \tag{3.17}$$

Otherwise, if $\boldsymbol{u}_{J^c} = 0$, we argue that

$$\lambda_n\big(\Phi(\boldsymbol{c} \circ |\boldsymbol{x}^\natural + \frac{\boldsymbol{u}}{\sqrt{n}}|) - \Phi(\boldsymbol{c} \circ |\boldsymbol{x}^\natural|)\big) = \lambda_n(\Phi(\boldsymbol{c}_J \circ |\boldsymbol{x}^\natural_J + \frac{\boldsymbol{u}_J}{\sqrt{n}}|) - \Phi(\boldsymbol{c}_J \circ |\boldsymbol{x}^\natural_J|)) \xrightarrow{p} 0. \tag{3.18}$$

To see this note first that since x^0 is a $\sqrt{n}$-consistent estimator to $\boldsymbol{x}^\natural$, then $\boldsymbol{c}_J = |x^0_J|^{\alpha-1} \xrightarrow{p} |\boldsymbol{x}^\natural_J|^{\alpha-1}$, $\boldsymbol{c}_J \circ |\boldsymbol{x}^\natural_J| \xrightarrow{p} |\boldsymbol{x}^\natural_J|^\alpha$ and $\boldsymbol{c}_J \circ |\boldsymbol{x}^\natural_J + \frac{\boldsymbol{u}_J}{\sqrt{n}}| \xrightarrow{p} |x^\star_J|^\alpha$. Then by the assumption $|\boldsymbol{x}^\natural|^\alpha \in \operatorname{int}\operatorname{dom}\Phi$, we have that both $\boldsymbol{c}_J \circ |\boldsymbol{x}^\natural_J|, \boldsymbol{c}_J \circ |\boldsymbol{x}^\natural_J + \frac{\boldsymbol{u}_J}{\sqrt{n}}| \in \operatorname{int}\operatorname{dom}\Phi$ with probability going to one. By convexity, we then have:

$$\begin{aligned}
\lambda_n(\Phi(\boldsymbol{c}_J \circ |\boldsymbol{x}^\natural_J + \frac{\boldsymbol{u}_J}{\sqrt{n}}|) - \Phi(\boldsymbol{c}_J \circ |\boldsymbol{x}^\natural_J|)) &\geq \langle \nabla\Phi(\boldsymbol{c}_J \circ |\boldsymbol{x}^\natural_J|), \lambda_n\frac{\boldsymbol{u}_J}{\sqrt{n}}\rangle \\
\lambda_n(\Phi(\boldsymbol{c}_J \circ |\boldsymbol{x}^\natural_J + \frac{\boldsymbol{u}_J}{\sqrt{n}}|) - \Phi(\boldsymbol{c}_J \circ |\boldsymbol{x}^\natural_J|)) &\leq \langle \nabla\Phi(\boldsymbol{c}_J \circ |\boldsymbol{x}^\natural_J + \frac{\boldsymbol{u}_J}{\sqrt{n}}|), \lambda_n\frac{\boldsymbol{u}_J}{\sqrt{n}}\rangle
\end{aligned}$$

where $\nabla\Phi(\boldsymbol{x})$ denotes a subgradient of Φ at x.

For all $\boldsymbol{x} \in \operatorname{int}\operatorname{dom}\Phi$ where Φ is convex, monotone and normalized, we have that $\|\boldsymbol{z}\|_\infty < \infty, \forall \boldsymbol{z} \in \partial\Phi(\boldsymbol{x})$. To see this, note that since $\boldsymbol{x} \in \operatorname{int}\operatorname{dom}\Phi$, $\exists\delta > 0$ s.t., $\forall \boldsymbol{x}' \in B_\delta(\boldsymbol{x}), \Phi(\boldsymbol{x}') < +\infty$. Let $\boldsymbol{x}' = \boldsymbol{x} + \operatorname{sign}(\boldsymbol{z})\mathbb{1}_{i_{\max}}\delta$, where $i_{\max}$ denotes the index where $|z_{i_{\max}}| = \|\boldsymbol{z}\|_\infty$ then by convexity we have

$$\begin{aligned} \Phi(\boldsymbol{x}') &\geq \Phi(\boldsymbol{x}) + \langle \boldsymbol{z}, \boldsymbol{x}' - \boldsymbol{x}\rangle, && \forall \boldsymbol{z} \in \partial\Phi(\boldsymbol{x}) \\ +\infty > \Phi(\boldsymbol{x}') &\geq \|\boldsymbol{z}\|_\infty\delta, && \forall \boldsymbol{z} \in \partial\Phi(\boldsymbol{x}), \quad (\text{since } \Phi(\boldsymbol{x}) \geq 0) \end{aligned}$$

Since $\frac{\lambda_n}{\sqrt{n}} \to 0$, we can then conclude by Slutsky's theorem that (3.18) holds.

Hence by (3.17) and (3.18),

$$\lambda_n\big(\Phi(\boldsymbol{c} \circ |\boldsymbol{x}^\natural + \frac{\boldsymbol{u}}{\sqrt{n}}|) - \Phi(\boldsymbol{c} \circ |\boldsymbol{x}^\natural|)\big) \xrightarrow{p} \begin{cases} 0 & \text{if } \boldsymbol{u}_{J^c} = 0 \\ \infty & \text{Otherwise} \end{cases}. \tag{3.19}$$

By CLT, $\frac{\boldsymbol{A}^\top\boldsymbol{\varepsilon}}{\sqrt{n}} \xrightarrow{d} \boldsymbol{W} \sim \mathcal{N}(0, \sigma^2\boldsymbol{Q})$, it follows then that $V_n(\boldsymbol{u}) \xrightarrow{d} V(\boldsymbol{u})$, where

$$V(\boldsymbol{u}) = \begin{cases} \frac{1}{2}\boldsymbol{u}_J^T\boldsymbol{Q}_{JJ}\boldsymbol{u}_J - \boldsymbol{W}_J^T\boldsymbol{u}_J & \text{if } \boldsymbol{u}_{J^c} = 0 \\ \infty & \text{Otherwise} \end{cases}.$$

V_n is convex and the unique minimum of V is $\boldsymbol{u}_J = \boldsymbol{Q}_{JJ}^{-1}\boldsymbol{W}_J, \boldsymbol{u}_{J^c} = 0$, hence by epi-convergence results (see [Zou06])

$$\boldsymbol{u}_J^\star \xrightarrow{d} \boldsymbol{Q}_{JJ}^{-1}\boldsymbol{W}_J \sim \mathcal{N}(0, \sigma^2\boldsymbol{Q}_{JJ}^{-1}), \quad \boldsymbol{u}_{J^c}^\star \xrightarrow{d} 0. \tag{3.20}$$

Since $\boldsymbol{u}^\star = \sqrt{n}(\boldsymbol{x}^\star - \boldsymbol{x}^\natural)$, then it follows from (3.20) that

$$\boldsymbol{x}_J^\star \xrightarrow{p} \boldsymbol{x}_J^\natural, \quad \boldsymbol{x}_{J^c}^\star \xrightarrow{p} 0 \tag{3.21}$$

Hence, $P(\operatorname{supp}(\boldsymbol{x}^\star) \supseteq J) \to 1$ and it is sufficient to show that $P(\operatorname{supp}(\boldsymbol{x}^\star) \subseteq J) \to 1$ to complete the proof.

For that denote $J^\star = \operatorname{supp}(\boldsymbol{x}^\star)$ and let's consider the event $J^\star \setminus J \neq \emptyset$. By optimality conditions, we know that

$$-\boldsymbol{A}_{J^\star\setminus J}^T(\boldsymbol{A}\boldsymbol{x}^\star - \boldsymbol{y}) \in \lambda_n[\partial\Phi(\boldsymbol{c} \circ \cdot)(\boldsymbol{x}^\star)]_{J^\star\setminus J}$$

Note, that $-\frac{\boldsymbol{A}_{J^\star\setminus J}^T(\boldsymbol{A}\boldsymbol{x}^\star - \boldsymbol{y})}{\sqrt{n}} = \frac{\boldsymbol{A}_{J^\star\setminus J}^T\boldsymbol{A}(\boldsymbol{x}^\star - \boldsymbol{x}^\natural)}{\sqrt{n}} - \frac{\boldsymbol{A}_{J^\star\setminus J}^T\boldsymbol{\varepsilon}}{\sqrt{n}}$. By CLT, $\frac{\boldsymbol{A}_{J^\star\setminus J}^T\boldsymbol{\varepsilon}}{\sqrt{n}} \xrightarrow{d} \boldsymbol{W} \sim \mathcal{N}(0, \sigma^2\boldsymbol{Q}_{J^\star\setminus J, J^\star\setminus J})$

and by (3.21) $\boldsymbol{x}^\star - \boldsymbol{x}^\natural \xrightarrow{p} 0$ then $-\frac{\boldsymbol{A}^T_{J^\star\setminus J}(\boldsymbol{A}\boldsymbol{x}^\star - \boldsymbol{y})}{\sqrt{n}} = O_p(1)$.

On the other hand, $\frac{\lambda_n \boldsymbol{c}_{J^\star\setminus J}}{\sqrt{n}} = \lambda_n n^{\frac{1-\alpha}{2}} n^{\frac{\alpha-1}{2}} \boldsymbol{c}_{J^\star\setminus J} \to \infty$, hence $\frac{\lambda_n \boldsymbol{c}_{J^\star\setminus J}}{\sqrt{n}} \boldsymbol{c}^{-1}_{J^\star\setminus J} \boldsymbol{v}_{J^\star\setminus J} \to \infty$, $\forall \boldsymbol{v} \in \partial\Phi(\boldsymbol{c}\circ\cdot)(\boldsymbol{x}^\star)$, since $\boldsymbol{c}^{-1}_{J^\star\setminus J}\boldsymbol{v}_{J^\star\setminus J} = O_p(1)^{-1}$. To see this, let $\boldsymbol{x}'_J = \boldsymbol{x}^\star_J$ and 0 elsewhere. Note that by definition of the subdifferential and the stability assumption on J, there must exists $M_J > 0$ s.t

$$\begin{aligned}
\Phi(\boldsymbol{c}\circ\boldsymbol{x}') &\geq \Phi(\boldsymbol{c}\circ\boldsymbol{x}^\star) + \langle \boldsymbol{v}_{J^\star\setminus J}, -\boldsymbol{x}^\star_{J^\star\setminus J}\rangle \\
&\geq \Phi(\boldsymbol{c}\circ\boldsymbol{x}') + M_J\|\boldsymbol{c}_{J^\star\setminus J}\circ\boldsymbol{x}^\star_{J^\star\setminus J}\|_\infty - \|\boldsymbol{c}^{-1}_{J^\star\setminus J}\circ\boldsymbol{v}_{J^\star\setminus J}\|_1\|\boldsymbol{c}_{J^\star\setminus J}\circ\boldsymbol{x}^\star_{J^\star\setminus J}\|_\infty \\
\|\boldsymbol{c}^{-1}_{J^\star\setminus J}\circ\boldsymbol{v}_{J^\star\setminus J}\|_1 &\geq M_J
\end{aligned}$$

We deduce then that $P(\mathrm{supp}(\boldsymbol{x}^\star) \subseteq J) = 1 - P(J^\star \setminus J \neq \emptyset) \to 1$. □

3.7.4 Proof of Proposition 8 and relation to weak submodularity

Proposition 8. *If F is a finite-valued monotone function, F is ρ-submodular iff discrete weak stability is equivalent to strong stability.*

Proof. If F is ρ-submodular and J is weakly stable, then $\forall A \subseteq J, \forall i \in J^c, 0 < \rho[F(J\cup\{i\}) - F(J)] \leq F(J\cup\{i\}) - F(J)$, i.e., J is strongly stable w.r.t. F. If F is such that any weakly stable set is also strongly stable, then if F is not ρ-submodular, then $\forall\rho \in (0,1]$ there must exists a set $B \subseteq V$, s.t., $\exists A \subseteq B, i \in B^c$, s.t., $\rho[F(B\cup\{i\}) - F(B)] > F(A\cup\{i\}) - F(A) \geq 0$. Hence, $F(B\cup\{i\}) - F(B) > 0$, i.e., B is weakly stable and thus it is also strongly stable and we must have $F(A\cup\{i\}) - F(A) > 0$. Choosing then in particular, $\rho = \min_{B\subseteq V}\min_{A\subseteq B, i\in B^c}\frac{F(A\cup\{i\})-F(A)}{F(B\cup\{i\})-F(B)} \in (0,1]$, leads to a contradiction; $\min_{A\subseteq B, i\in B^c} F(A\cup\{i\}) - F(A) \geq \rho[F(B\cup\{i\}) - F(B)] > F(A\cup\{i\}) - F(A)$. □

We show that ρ-submodularity is a stronger condition than weak submodularity. First, we recall the definition of weak submodular functions.

Definition 14 (Weak Submodularity (see [DK11, EKDN16])). *A function F is weakly submodular if $\forall S, L, S\cap L = \emptyset, F(L\cup S) - F(L) > 0$,*

$$\gamma_{S,L} = \frac{\sum_{i\in S} F(L\cup\{i\}) - F(L)}{F(L\cup S) - F(L)} > 0$$

Proposition 13. *If F is ρ-submodular then F is weakly submodular. But the converse is not true.*

Proof. If F is ρ-submodular then $\forall S, L, S \cap L = \emptyset, F(L \cup S) - F(L) > 0$, let $S = \{i_1, i_2, \cdots, i_r\}$

$$\begin{aligned} F(L \cup S) - F(L) &= \sum_{k=1}^{r} F(L \cup \{i_1, \cdots, i_k\}) - F(L \cup \{i_1, \cdots, i_{k-1}\}) \\ &\leq \sum_{k=1}^{r} \frac{1}{\rho}(F(L \cup \{i_k\}) - F(L)) \\ \Rightarrow \gamma_{S,T} &= \rho > 0. \end{aligned}$$

We show that the converse is not true by giving a counter-example: Consider the function defined on $V = \{1, 2, 3\}$, where $F(\{i\}) = 1, \forall i, F(\{1, 2\}) = 1, F(\{2, 3\}) = 2, F(\{1, 3\}) = 2, F(\{1, 2, 3\}) = 3$. Then note that this function is weakly submodular. We only need to consider sets $|S| \geq 2$, since otherwise $\gamma_{S,T} > 0$ holds trivially. Accordingly, we also only need to consider L which is the empty set or a singleton. In both cases $\gamma_{S,T} > 0$. However, this F is not ρ-submodular, since $F(1, 2) - F(1) = 0 < \rho(F(1, 2, 3) - F(1, 3)) = \rho$ for any $\rho > 0$. □

3.7.5 Proof of Propositions 9 and 10, and Corollary 9

First, we present a useful simple lemma, which provides an equivalent definition of decomposability for monotone function.

Lemma 8. *Given $\boldsymbol{x} \in \mathbb{R}^d, J \subseteq J, \mathrm{supp}(\boldsymbol{x}) = J$, if Φ is a monotone function, then Φ is decomposable at $\boldsymbol{x}$ w.r.t J iff $\exists M_J > 0, \forall \delta > 0, i \in J^c$, s.t,*

$$\Phi(\boldsymbol{x} + \delta \mathbb{1}_i) \geq \Phi(\boldsymbol{x}) + M_J \delta.$$

Proof. By definition 11, $\exists M_J > 0, \forall \Delta \in \mathbb{R}^d, \mathrm{supp}(\Delta) \subseteq J^c$,

$$\Phi(\boldsymbol{x} + \Delta) \geq \Phi(\boldsymbol{x}) + M_J \|\Delta\|_\infty.$$

in particular this must hold for $\Delta = \delta \mathbb{1}_i$. On the other hand, if the inequality hold for all $\delta \mathbb{1}_i$, then given any Δ s.t. $\mathrm{supp}(\Delta) \subseteq J^c$ let $i_{\max}$ be the index where $\Delta_{i_{\max}} = \|\Delta\|_\infty$ and let $\delta = \|\Delta\|_\infty$, then

$$\Phi(\boldsymbol{x} + \Delta) \geq \Phi(\boldsymbol{x} + \delta_{i_{\max}}) \geq \Phi(\boldsymbol{x}) + M_J \delta = \Phi(\boldsymbol{x}) + M_J \|\Delta\|_\infty.$$

□

Proposition 9. *Given any monotone set function F, all sets $J \subseteq V$ strongly stable w.r.t to F are also strongly stable w.r.t Ω_p and Θ_p.*

Proof. We make use of the variational form (3.11). Given a set J stable w.r.t to F and $\mathrm{supp}(\boldsymbol{x}) \subseteq J$, let $\boldsymbol{\kappa}^\star \in \arg\max_{\boldsymbol{\kappa} \in \mathbb{R}^d_+} \{\sum_{i \in J} \kappa_i^{1/q} |x_i| : \boldsymbol{\kappa}(A) \leq F(A), \forall A \subseteq V\}$, then $\Omega(\boldsymbol{x}) =$

$|\boldsymbol{x}_J|^T(\boldsymbol{\kappa}^\star_J)^{1/q}$. Note that $\forall A \subseteq J, F(A \cup i) > F(A)$, by definition 12. Hence, $\forall i \in J^c$, we can define $\boldsymbol{\kappa}' \in \mathbb{R}^d_+$ s.t., $\boldsymbol{\kappa}'_J = \boldsymbol{\kappa}^\star_J$, $\boldsymbol{\kappa}'_{(J\cup i)^c} = 0$ and $\kappa'_i = \min_{A\subseteq J} F(A\cup i) - F(A) > 0$. Note that $\boldsymbol{\kappa}'$ is feasible, since $\forall A \subseteq J, \boldsymbol{\kappa}'(A) = \boldsymbol{\kappa}^\star(A) \leq F(A)$ and $\boldsymbol{\kappa}'(A+i) = \boldsymbol{\kappa}^\star(A) + \kappa'_i \leq F(A) + F(A \cup i) - F(A) = F(A \cup i)$. For any other set $\boldsymbol{\kappa}'(A) = \boldsymbol{\kappa}'(A \cap (J+i)) \leq F(A \cap (J+i)) \leq F(A)$, by monotonicity. It follows then that $\Omega(\boldsymbol{x} + \delta \mathbb{1}_i) = \max_{\boldsymbol{\kappa}\in\mathbb{R}^d_+}\{\sum_{i\in J\cup i}^d \kappa_i^{1/q}|x_i| : \boldsymbol{\kappa}(A) \leq F(A), \forall A \subseteq V\} \geq |\boldsymbol{x}_J|^T(\boldsymbol{\kappa}^\star_J)^{1/q} + \delta(\kappa'_i)^{1/q} \geq \Omega(\boldsymbol{x}) + \delta M$, with $M = (\kappa'_i)^{1/q} > 0$. The proposition then follows by lemma 8.

The proof for Θ_p follows in a similar fashion. We make use of the variational form (3.15). Given a set J stable w.r.t to F and $\mathrm{supp}(\boldsymbol{x}) \subseteq J$, first note that this implicitly implies that $F(J) < +\infty$ and hence $\Theta_p(\boldsymbol{x}) < +\infty$. Let $\kappa^\star \in \arg\max_{\boldsymbol{\kappa}\in\mathbb{R}^d_+} \sum_{j=1}^d \psi_j(\kappa_j, x_j) + \min_{S\subseteq V} F(S) - \boldsymbol{\kappa}(S)$ and $S^\star \in \arg\min_{S\subseteq J} F(S) - \boldsymbol{\kappa}^\star(S)$. Note that $\forall S \subseteq J, \forall i \in J^c, F(S \cup i) > F(S)$, by definition 12. Hence, $\forall i \in J^c$, we can define $\boldsymbol{\kappa}' \in \mathbb{R}^d_+$ s.t., $\boldsymbol{\kappa}'_J = \boldsymbol{\kappa}^\star_J$, $\boldsymbol{\kappa}'_{(J\cup i)^c} = 0$ and $\kappa'_i = \min_{S\subseteq J} F(S \cup i) - F(S) > 0$. Note that $\forall S \subseteq J, F(S) - \boldsymbol{\kappa}'(S) = F(S) - \boldsymbol{\kappa}^\star(S) \geq F(S^\star) - \boldsymbol{\kappa}^\star(S^\star)$ and $F(S+i) - \boldsymbol{\kappa}'(S+i) = F(S+i) - \boldsymbol{\kappa}^\star(S) - \kappa'_i \geq F(S+i) - \boldsymbol{\kappa}^\star(S) - F(S+i) + F(S) \geq F(S^\star) - \boldsymbol{\kappa}^\star(S^\star)$. Note also that $\psi_i(\kappa'_i, \delta) = (\kappa'_i)^{1/q}\delta$ if $\delta \leq (\kappa'_i)^{1/p}$, and $\psi_i(\kappa'_i, \delta) = \frac{1}{p}\delta^p + \frac{1}{q}\kappa'_i = \delta(\frac{1}{p}\delta^{p-1} + \frac{1}{q}\kappa'_i\delta^{-1}) \geq \delta(\kappa'_i)^{1/q}$ otherwise. It follows then that $\Theta_p(\boldsymbol{x} + \delta\mathbb{1}_i) \geq \sum_{j\in J}\psi_j(\kappa_j, x_j) + (\kappa'_i)^{1/q}\delta + \min_{S\subseteq J\cup i} F(S) - \boldsymbol{\kappa}'(S) \geq \sum_{j\in J}\psi_j(\kappa_j, x_j) + (\kappa'_i)^{1/q}\delta + \min_{S\subseteq J} F(S) - \boldsymbol{\kappa}^\star(S) = \Theta_p(\boldsymbol{x}) + \delta M$ with $M = (\kappa'_i)^{1/q} > 0$. The proposition then follows by lemma 8. □

Proposition 10. *If $F = F_-$ and J is strongly stable w.r.t Ω_∞, then J is strongly stable w.r.t F. Similarly, for any monotone F, if J is strongly stable w.r.t Θ_∞, then J is strongly stable w.r.t F.*

Proof. $F(A+i) = \Omega_\infty(\mathbb{1}_A + \mathbb{1}_i) = \Theta_\infty(\mathbb{1}_A + \mathbb{1}_i) > \Omega_\infty(\mathbb{1}_A) = \Theta_\infty(\mathbb{1}_A) = F(A), \forall A \subseteq J$. □

Corollary 9. *If F is monotone submodular and J is weakly stable w.r.t $\Omega_\infty = \Theta_\infty$ then J is weakly stable w.r.t F.*

Proof. If F is a monotone submodular function, then $\Omega_\infty(\boldsymbol{x}) = \Theta_\infty(\boldsymbol{x}) = f_L(|\boldsymbol{x}|)$. If J is not weakly stable w.r.t F, then $\exists i \in J^c$ s.t., $F(J \cup \{i\}) = F(J)$. Thus, given any $\boldsymbol{x}, \mathrm{supp}(\boldsymbol{x}) = J$, choosing $0 < \delta < \min_{i\in J}|x_i|$, result in $f_L(|\boldsymbol{x}| + \delta\mathbb{1}_i) = f_L(|\boldsymbol{x}|)$, which contradicts the weak stability of J w.r.t to $\Omega_\infty = \Theta_\infty$. □

4 Non-Euclidean Convex Composite Optimization

4.1 Introduction

In the previous two chapters, we studied how to relax discrete descriptions of structured sparsity models to convex ones. The obtained convex penalties are naturally non-smooth. In this chapter, we are interested in optimizing the resulting convex composite minimization problem:

$$F^\star = \min_{\boldsymbol{x}\in\mathbb{R}^d} \left\{ F(\boldsymbol{x}) := f(\boldsymbol{x}) + g(\boldsymbol{x}) \right\}, \tag{4.1}$$

where f is a smooth convex loss function, and g is a non-smooth convex regularizer, which acts as a structure prior. Such problems are prevalent in machine learning and signal processing, beyond structured sparsity problems.

The proximal gradient method and its accelerated variant (see Section 1.4) are the methods of choice for solving (4.1), whenever the proximal operator of g can be computed efficiently:

$$\mathrm{prox}_g^{\ell_2}(\boldsymbol{u}, \boldsymbol{z}, L_2) := \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} \boldsymbol{z}^T\boldsymbol{x} + \frac{L_2}{2}\|\boldsymbol{x}-\boldsymbol{u}\|_2^2 + g(\boldsymbol{x}), \tag{4.2}$$

where the constant L_2 is the Lipschitz constant of ∇f with respect to the ℓ_2-norm, and $\boldsymbol{z}$ is the gradient of f at the current iterate $\boldsymbol{u}$ within the proximal gradient method.

The accelerated variant is known to converge with a rate of $O(1/k^2)$, which is optimal in terms of dependence on k, for first-order methods in [NYD83]. However, the choice of the ℓ_2-norm in these methods, may lead to suboptimal convergence, in terms of dimension dependence, for problems which are not "well-behaved" in the ℓ_2-norm. We thus consider here their extension to a General Proximal gradient Method (GPM), using the following operator:

$$\mathrm{prox}_{\mathrm{g}}(\boldsymbol{u}, \boldsymbol{z}, L) \in \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} \boldsymbol{z}^T\boldsymbol{x} + \frac{L}{2}\|\boldsymbol{x}-\boldsymbol{u}\|^2 + g(\boldsymbol{x}), \tag{4.3}$$

where $\|\cdot\|$ is any norm and L is the Lipschitz constant of ∇f with respect to the chosen norm.

Note that, unlike (4.2) where the solution is unique, (4.3) can be set valued.

The interest in this generalization stems from the benefit it can entail on the convergence, in terms of dimension dependence, as observed in the context of (projected) gradient descent method, for e.g, in [KLOS14, Nes05, BWB14, dGJ13]. In this chapter, we further identify two additional benefits; an appropriate choice of the norm can also lead, in some cases, to cheaper and sparse updates.

4.1.1 Related work

The convergence rate of GPM with general norms was studied in [RT14]. However, the general proximal operator (4.3), to be solved at each iteration in GPM, is not well-studied outside the Euclidean setting. Indeed, for non-Euclidean norms, it was only shown to be tractable in the special case where $\|\cdot\|$ is the ℓ_1-norm, and g is the ℓ_1-norm [SCJX17], or the indicator function of the simplex [Nes05].

Other extensions of proximal gradient methods to non-Euclidean settings, where the ℓ_2-norm in (4.2) is replaced by a Bregman divergence, were considered for example in [Tse08, Lan12]. However, as Bregman divergences are required to be strongly convex in the underlying norm, they also can introduce unnecessary dimension dependence terms in the convergence rate.

On the other hand, generalized conditional gradient (GCG) method provides an attractive alternative to solve (4.1), with cheaper and sparse updates, and a performance invariant to the choice of the norm used to measure properties of f, but at a slower convergence rate in general (see Section 1.4.2). Moreover, this method is only applicable when g has bounded domain. Variants of GCG able to handle the special case where g is a norm or a gauge, where proposed in [HJN15] and [YZS17], respectively. These variants require an additional exact line search step at each iteration, which can be expensive in general, and converge at the slower rate of $O(1/k)$.

4.1.2 Contributions

In this chapter, we establish the tractability of GPM for a broad class of regularizers and norms, including examples where the Euclidean proximal operator is not known to be efficently computable, such as latent group Lasso, exclusive Lasso, and general overlapping group Lasso. In addition, we propose the first—to our knowledge—accelerated variant of GPM for the general composite problem (4.1). Our specific contributions can be summarized as follows:

- We introduce a polynomial-time method, which performs a logarithmic number of linear optimization steps, to approximately compute the non-Euclidean proximal operator (4.3), for any *polyhedral* norm and regularizer (Section 4.3.1).
- For a special class of regularizers, namely for atomic norms with linearly independent atoms, and a matching choice of the norm, we design an efficient greedy algorithm to

compute (4.3) exactly (Section 4.3.2). The resulting iterates, in this approach, form a sparse convex combination of only few "atoms", which is a desirable property in several applications, and particularly in structured sparsity problems.

- We propose an accelerated variant of GPM (accGPM), with a small extra computational cost. Existing acceleration schemes in the non-Euclidean setting require an additional proximal/projection operation with respect a strongly convex *prox function*. We introduce a new type of estimate sequences which allow us to avoid such computation (Section 4.4).

- We illustrate our results on the Lasso and latent group Lasso problems (Section 4.5).

This chapter is based on the joint work with Ya-Ping Hsieh, Bang Vu, Quang Nguyen, and Volkan Cevher [EHHV+17].

In the sequel, we defer all proofs to the appendix of this chapter.

4.1.3 Preliminaries

We use the set Γ_0 to denote all proper lower semi-continuous convex functions on $\mathbb{R}^d$. We consider problems of the form (4.1), whose set of minimizers $\mathscr{X}^\star$ is assumed to be non-empty with $f, g \in \Gamma_0$.

We further assume that the gradient of f is L-Lipschitz continuous with respect to $\|\cdot\|$, i.e., $\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|_* \leq L\|\boldsymbol{x} - \boldsymbol{y}\|$, where $\|\cdot\|_*$ is the dual norm of $\|\cdot\|$. This property implies the following majorizer for any $\gamma \in (0, 1/L]$:

$$f(\boldsymbol{x}) \leq f(\boldsymbol{y}) + \langle \boldsymbol{x} - \boldsymbol{y}, \nabla f(\boldsymbol{y})\rangle + \frac{1}{2\gamma}\|\boldsymbol{x} - \boldsymbol{y}\|^2. \tag{4.4}$$

A function f is μ-strongly convex with respect to $\|\cdot\|$ if, $\forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d, \forall \mathbf{p} \in \partial f(\boldsymbol{y})$, it holds that

$$f(\boldsymbol{x}) \geq f(\boldsymbol{y}) + \langle \boldsymbol{x} - \boldsymbol{y}, \mathbf{p}\rangle + \frac{\mu}{2}\|\boldsymbol{x} - \boldsymbol{y}\|^2. \tag{4.5}$$

4.2 Generalized proximal gradient method: Warm-up

The general proximal gradient method (GPM) is the iterative scheme where $\boldsymbol{x}^{k+1} \in \mathrm{prox}_{\mathrm{g}}(\boldsymbol{x}^k, \nabla f(\boldsymbol{x}^k), L)$. For completeness, we state below its basic convergence result.

Theorem 4. *The iterates $\boldsymbol{x}^k$ of GPM satisfy $\forall k \in \mathbb{N}$:*

$$F(\boldsymbol{x}^k) - F^\star \leq \frac{2\max\{L\mathcal{R}(\boldsymbol{x}^0), F(\boldsymbol{x}^0) - F^\star\}}{k}$$

where $\mathcal{R}(\boldsymbol{x}^0) = \max_{\{\boldsymbol{x}: F(\boldsymbol{x}) \leq F(\boldsymbol{x}^0)\}} \min_{\boldsymbol{x}^\star \in \mathcal{X}^\star} \|\boldsymbol{x} - \boldsymbol{x}^\star\|^2$. If, in addition, $f(\boldsymbol{x})$ is μ-strongly

convex w.r.t. norm $\|\cdot\|$*, then GPM satisfies*

$$F(\boldsymbol{x}^k) - F^\star \leq (1 - \frac{\mu}{L})^k \big(F(\boldsymbol{x}^0) - F^\star\big).$$

It is easy to see that the convergence rate of GPM depends on the choice of norm. Choosing a non-Euclidean norm can lead in some cases to smaller Lipschitz constant L and level set radius $\mathcal{R}(\boldsymbol{x}^0)$, as well as larger (restricted) strong convexity constant μ (see Section 4.5.1 and [KLOS14, Nes05, BWB14]), thus yielding faster convergence.

Theorem 4 is not new; GPM has been analyzed in the context of randomized coordinate descent [RT14]. However, the primary interest of [RT14] is the weighted ℓ_2-norm, and the broader tractability question of the non-Euclidean norm choices is not addressed. We fill this gap in Section 4.3.

4.3 Tractability of the generalized proximal operator

To our knowledge, the computation of the proximal operator (4.3) for non-Euclidean norms is not addressed so far, except for the special case where $\|\cdot\|$ is the ℓ_1-norm, and g is the ℓ_1-norm [SCJX17], or the indicator function of the simplex [Nes05].

Section 4.3.1 shows that prox_g can be approximated in polynomial time, for the class of *polyhedral* functions g, if the norm is chosen to be an *atomic* norm $\|\cdot\|_{\mathcal{A}}$, with finitely many atoms (see Section 1.3.2 and [CRPW12]). In Section 4.3.1, we propose an efficient greedy algorithm to compute prox_g exactly, in the special case where g corresponds to an atomic norm, with *linear independent atoms*, and the norm in prox_g is chosen to be the same. In the simultaneous independent work of [SCJX17], another greedy algorithm, with the same cost as ours, was proposed to compute prox_g, in the special case where g and $\|\cdot\|$ are both the ℓ_1-norm.

We start by introducing a Moreau-like decomposition which relates, as in the Euclidean case (see Section 1.4.1), prox_g to the proximal operator of the Fenchel conjugate g^*, with respect to the dual norm $\|\cdot\|_*$, denoted by $\mathrm{prox}^*_{\mathrm{g}^*}$.

Proposition 14. *Generalized Moreau's decomposition Given* $g \in \Gamma_0$ *and its Fenchel* g^**, we have*

$$\mathbf{p} - \boldsymbol{z} \in \mathrm{prox}^*_{\mathrm{g}^*}(-\boldsymbol{z}, -\boldsymbol{u}, 1/L) \text{ and } \boldsymbol{x}^\star - \boldsymbol{u} \in -\partial\big(\frac{L}{2}\|\cdot\|_*^2\big)(\mathbf{p}) \cap \big(-\boldsymbol{u} + \partial g(\mathbf{p} - \boldsymbol{z})\big) \quad (4.6)$$

where $\mathbf{p} \in -\partial\big(\frac{L}{2}\|\cdot\|^2\big)(\boldsymbol{x}^\star - \boldsymbol{u}) \cap \big(\boldsymbol{z} + \partial g(\boldsymbol{x}^\star)\big)$ *and* $\boldsymbol{x}^\star \in \mathrm{prox}_{\mathrm{g}}(\boldsymbol{u}, \boldsymbol{z}, L)$.

This relation allow us to efficiently compute both proximal operators, whenever one of them is efficiently computable, and finding an element in the intersection of the two subdifferential sets (4.6) is easy. Such operation is also required in the acceleration of GPM. Section 4.4 describes how to find such an element for some examples of interest.

A simple but key observation to our proposed framework is given below:

Lemma 9. *Let* $h(t) = \min_{\|\boldsymbol{x}-\boldsymbol{u}\|\leq t} \boldsymbol{z}^T\boldsymbol{x} + g(\boldsymbol{x})$ *and* $t^\star \in \frac{\partial h(t^\star)}{L}$ *then*

$$\boldsymbol{x}^\star \in \mathrm{prox_g}(\boldsymbol{u}, \boldsymbol{z}, L) \Leftrightarrow \boldsymbol{x}^\star \in \underset{\|\boldsymbol{x}-\boldsymbol{u}\|\leq t^\star}{\arg\min}\ \boldsymbol{z}^T\boldsymbol{x} + g(\boldsymbol{x}). \tag{4.7}$$

Computing $\mathrm{prox_g}$ can be seen then as computing the Fenchel conjugate operator[1] of g *locally*, by restricting $\boldsymbol{x}$ in the norm ball of radius $t^\star$ around $\boldsymbol{u}$. Hence, we denote this operator by

$$\mathrm{lconj_g}(\boldsymbol{u}, \boldsymbol{z}, t) := \underset{\|\boldsymbol{x}-\boldsymbol{u}\|\leq t}{\arg\min}\ \boldsymbol{z}^T\boldsymbol{x} + g(\boldsymbol{x}).$$

Here, we note a close connection between our local Fenchel conjugate operator and the local linear minimization oracle proposed by [GH16]. The latter is a special case of $\mathrm{lconj_g}$, with $g = \iota_P$ for a polytope P.

4.3.1 Atomic proximal operator of polyhedral functions

In this section, we propose a polynomial time approach to approximately compute $\mathrm{prox_g}$ for any *polyhedral* function g, i.e., $P_g := \mathrm{epi}(g)$ is a polytope. Examples where g is a polyhedral function are abundant, including structured sparsity-inducing penalties (see Sections 1.3.1 and 2.5)[2] and general atomic norms [CRPW12]. For further examples, see [Jag13, GH16, LJJ15].

We choose the norm in $\mathrm{prox_g}$ to be any atomic norm, i.e., $\|\boldsymbol{x}\|_{\mathcal{A}} = \inf_{t>0}\{t : \boldsymbol{x} \in t\ \mathrm{conv}(\mathcal{A})\}$, where the atomic set $\mathcal{A}$ is centrally symmetric with finitely many atoms. We denote the polytope $P_{\mathcal{A}} := \mathrm{conv}(\mathcal{A})$ and the resulting proximal operator by $\mathrm{prox}_g^{\mathcal{A}}$.

Our choice of the atomic norm is motivated by the following observation.

$$h(t) = \min_{\|\boldsymbol{x}-\boldsymbol{u}\|_{\mathcal{A}}\leq t} \boldsymbol{z}^T\boldsymbol{x} + g(\boldsymbol{x}) = \min_{\substack{\boldsymbol{x}-\boldsymbol{u}\in tP_{\mathcal{A}} \\ (\boldsymbol{x},y)\in P_g}} \boldsymbol{z}^T\boldsymbol{x} + y. \tag{4.8}$$

Hence, h is a non-increasing piecewise linear function and $h(t)$ can be computed, for any t, by a linear program (LP). We will assume P_g and $P_{\mathcal{A}}$ are solvable polytopes, i.e., they have efficient linear minimization oracles. Hence, the LP (4.8) can be solved in polynomial time.

Since $h(t)$ is a non-increasing piecewise-linear function, its subdifferential can be approximated by $\partial h(t) \simeq [\frac{h(t)-h(t+\epsilon)}{\epsilon}, \frac{h(t-\epsilon)-h(t)}{\epsilon}]$ for a small enough $\epsilon > 0$. If t is a differentiable point of $h(t)$, the interval would correspond to a unique value. The optimal $t^\star$ can then be obtained via binary search over the interval $t^\star \in [t_{\min}, t_{\max}]$ where $t_{\min} = \min_{(\boldsymbol{x},y)\in P_g} \|\boldsymbol{x}-\boldsymbol{u}\|_{\mathcal{A}}$ and

[1] Recall from section 1.4.2, that the Fenchel conjugate operator is the operator used in GCG.

[2] In fact, we can see from the variational forms presented in Chapter 3 (Section 3.7.1), that any homogeneous or non-homogeneous convex envelope of a combinatorial function, over the unit ℓ_∞-ball, is polyhedral.

$t_{\max} = \|\boldsymbol{x}_{\min} - \boldsymbol{u}\|_{\mathcal{A}}$ where $\boldsymbol{x}_{\min} \in \arg\min_{\boldsymbol{x}} \boldsymbol{z}^T\boldsymbol{x} + g(\boldsymbol{x})$. By Lemma 9, we reach the optimal $t^\star$ when $t^\star \in \frac{\partial h(t)}{L}$. Algorithm 1 provides a pseudocode for this approach.

Algorithm 1 Atomic prox of polyhedral functions

Input: $t_{\min} > 0, t_{\max} > 0, \delta > 0, \epsilon > 0$
while $|t_{\max} - t_{\min}| > \delta$ **do**
  $t = (t_{\min} + t_{\max})/2$;
  $\text{slope}_1 = \frac{h(t)-h(t+\epsilon)}{\epsilon}$
  $\text{slope}_2 = \frac{h(t-\epsilon)-h(t)}{\epsilon}$
  if $\text{slope}_1 \leq Lt \leq \text{slope}_2$ **then**
    break
  else if $t - \text{slope}_1/L > 0$ **then**
    $t_{\max} = t$
  else
    $t_{\min} = t$
  end if
end while
Return: $\boldsymbol{x}^{k+1} \in \arg\min_{\|\boldsymbol{x}-\boldsymbol{u}\|_{\mathcal{A}} \leq t} \boldsymbol{z}^T\boldsymbol{x} + g(\boldsymbol{x})$

The binary search approach provides a simple strategy to compute $\text{prox}_g^{\mathcal{A}}$ approximately by a logarithmic number of LPs, for any polyhedral function g, including examples where the standard $\text{prox}_g^{\ell_2}$ is costly. Prominent examples include the ℓ_∞-latent group Lasso, ℓ_∞-exclusive Lasso, and general overlapping ℓ_∞-group Lasso, for which existing approaches to compute $\text{prox}_g^{\ell_2}$ are inefficient (see Section 1.4.1).

Note that the convergence analysis we provide in Sections 4.2 and 4.4 holds only for exact proximal operators. While the study of inexact GPM is straightforward (the gradient method is known to forgive inexact proximal operator calculations), the inexactness must be controlled for its acceleration, which is already a well-studied topic. We will ignore these issues in the sequel.

4.3.2 Proximal operator of atomic norms with linearly independent atoms

In this section, we consider the special case of polyhedral functions where g is the indicator function of an atomic norm with *linearly independent atoms*, i.e., $g = \iota_{\|\cdot\|_{\mathcal{A}} \leq \lambda}$, where $\mathcal{A} := \{\boldsymbol{a}_1, \cdots, \boldsymbol{a}_{2m}\}$, $(\boldsymbol{a}_i)_1^m$'s are linearly independent and $\boldsymbol{a}_i = -\boldsymbol{a}_{m+i}, \forall i = 1, \cdots, m$. To simplify the notation, we use cyclic indexing, i.e., $\boldsymbol{a}_{2m+i} = \boldsymbol{a}_i$. For example, for the ℓ_1-norm, $(\boldsymbol{a}_i)_1^m$ are the standard basis vectors.

We choose the matching norm in prox_g, i.e., $\|\cdot\| = \|\cdot\|_{\mathcal{A}}$. In this case, computing $h(t)$ corresponds to solving an LP over the intersection of the polytope $P_{\mathcal{A}} = \text{conv}(\mathcal{A})$ and its (scaled)

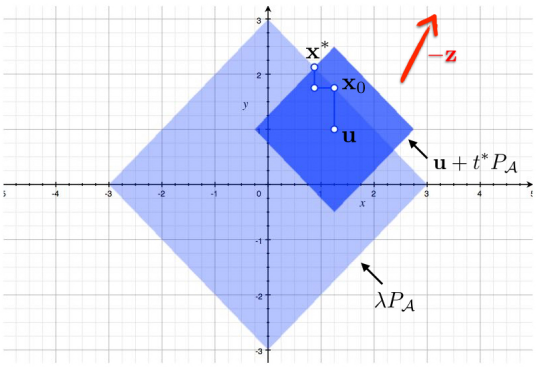


Figure 4.1: Illustration of the greedy Algorithm 2

translation by $\boldsymbol{u}$:

$$h(t) = \min_{\|\boldsymbol{x}-\boldsymbol{u}\|_{\mathcal{A}} \leq t} \boldsymbol{z}^T \boldsymbol{x} + g(\boldsymbol{x}) = \min_{\substack{\boldsymbol{x}-\boldsymbol{u} \in tP_{\mathcal{A}} \\ \boldsymbol{x} \in \lambda P_{\mathcal{A}}}} \boldsymbol{z}^T \boldsymbol{x}. \tag{4.9}$$

By the definition, we can represent $\boldsymbol{x} = \sum_{i=1}^{2m} c_i^x \boldsymbol{a}_i$, where $\boldsymbol{c}^x \geq 0$ such that $\sum_{i=1}^{2m} c_i^x = \|\boldsymbol{x}\|_{\mathcal{A}}$.

Lemma 10 shows that only linearly independent atoms are active in such a *unique* decomposition. We call this then a “minimal representation" decomposition and denote it by $\boldsymbol{c}^x = \mathrm{MR}(\boldsymbol{x})$.

Lemma 10. *Given* $\boldsymbol{x} = \sum_{i=1}^{2m} c_i^x \boldsymbol{a}_i, \boldsymbol{c}^x \geq 0$, *then* $\sum_{i=1}^{2m} c_i^x = \|\boldsymbol{x}\|_{\mathcal{A}} \Leftrightarrow \forall i, c_i^x = 0$ *or* $c_{i+m}^x = 0$.

Representing vectors in this fashion allows us to make the following simple observation.

Lemma 11. *Given* $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, *s.t* $\boldsymbol{c}^x = \mathrm{MR}(\boldsymbol{x}), \boldsymbol{c}^y = \mathrm{MR}(\boldsymbol{y})$, *we have* $\|\boldsymbol{x}-\boldsymbol{y}\|_{\mathcal{A}} = \|\boldsymbol{c}^x - \boldsymbol{c}^y\|_1$.

Based on these observations, computing $\mathrm{prox}_g^{\mathcal{A}}(\boldsymbol{u}, \boldsymbol{z}, L)$ reduces to the case where $\|\cdot\|_{\mathcal{A}} = \|\cdot\|_1$. We present in Algorithm 2 a fast greedy method that computes $\mathrm{prox}_g^{\mathcal{A}}(\boldsymbol{u}, \boldsymbol{z}, L)$ exactly and which only requires access to a linear minimization oracle $\mathrm{LMO}_{\mathcal{A}}(\boldsymbol{z}) \in \arg\min_{\boldsymbol{a} \in \mathcal{A}} \boldsymbol{z}^T \boldsymbol{a}$.

Note first that computing $t_{\min}$ and $t_{\max}$ is easy in this case: $t_{\min} = \min_{\boldsymbol{x} \in \lambda P_{\mathcal{A}}} \|\boldsymbol{x} - \boldsymbol{u}\|_{\mathcal{A}} = \min_{\|\boldsymbol{c}^x\|_1 \leq \lambda} \|\boldsymbol{c}^x - \boldsymbol{c}^u\|_1 = \max\{\|\boldsymbol{u}\|_{\mathcal{A}} - \lambda, 0\}$ (by lemma 11) and $t_{\max} = \max\{-\boldsymbol{z}^T \boldsymbol{a}_{i_{\min}}/L, t_{\min}\}$ where $\boldsymbol{a}_{i_{\min}} := \mathrm{LMO}_{\mathcal{A}}(\boldsymbol{z})$ and $-\boldsymbol{z}^T \boldsymbol{a}_{i_{\min}}/L$ corresponds to the largest slope of $h(t)$. For simplicity, Algorithm 2 presented here assumes the input is feasible, i.e., $\boldsymbol{u} \in \lambda P_{\mathcal{A}}$ and $t_{\min} = 0$. This is true for the iterates of GPM, but not for accGPM. The general algorithm is presented in the Appendix.

At a high level, Algorithm 2 acts the following way (see Figure 4.1 for an illustration): Assuming the optimal $t^\star$ is known, the algorithm starts at $\boldsymbol{u}$ and moves in the direction of the best atom

Algorithm 2 Prox of linearly independent atomic norms: $\mathrm{prox}_g^{\mathcal{A}}(\boldsymbol{u}, \boldsymbol{z}, L)$

1: **Input:** $\boldsymbol{c}^u = \mathrm{MR}(\boldsymbol{u})$.
2: **Initialize:** $\boldsymbol{x}^0 = \boldsymbol{u}, \boldsymbol{c}^x = \boldsymbol{c}^u, t_u^0 = 0$.
3: Let $\boldsymbol{a}_{i_{\min}} := \mathrm{LMO}_{\mathcal{A}}(\boldsymbol{z})$
4: Guess $t_l^0 = \max\{-\boldsymbol{z}^T\boldsymbol{a}_{i_{\min}}/L, 0\}$.
5: Sort $\boldsymbol{z}^T\boldsymbol{a}_i$ for active atoms: $\boldsymbol{z}^T\boldsymbol{a}_{j_1} \geq \boldsymbol{z}^T\boldsymbol{a}_{j_2} \geq \cdots, \forall c_j^u > 0$.
6: Let $\delta_0 = \max_{\delta>0}\{\delta\colon \boldsymbol{u} + \delta\boldsymbol{a}_{j_{\min}} \in \lambda P_{\mathcal{A}} \cap (t_l^0 P_{\mathcal{A}} + \boldsymbol{u})\} = \min\{t_l^0, \lambda - \|\boldsymbol{u}\|_{\mathcal{A}} + 2c_{i_{\min}+m}^u\}$
7: Update $\boldsymbol{x}^0 = \boldsymbol{u} + \delta_0\boldsymbol{a}_{i_{\min}}$.
8: Update weights: $c_{i_{\min}}^x = \max\{\delta_0 + c_{i_{\min}}^u - c_{i_{\min}+m}^u, 0\}, c_{i_{\min}+m}^x = -\min\{\delta_0 + c_{i_{\min}}^u - c_{i_{\min}+m}^u, 0\}$
9: Update $t_u^0 = \delta_0, t_l^r = t_l^0 - t_u^0$.
10: **while** $k = 1, \cdots, d$ and $t_l^k \geq 0$ **do**
11: Update guess $t_l^k = \max\{-0.5\boldsymbol{z}^T(\boldsymbol{a}_{i_{\min}} - \boldsymbol{a}_{j_k})/L - t_u^k, 0\}$.
12: Let $\delta_k = \max_{\delta>0}\{\delta\colon \boldsymbol{x}^{k-1} + \delta(\boldsymbol{a}_{i_{\min}} - \boldsymbol{a}_{j_k}) \in \lambda P_{\mathcal{A}} \cap (t_l^k P_{\mathcal{A}} + \boldsymbol{u})\} = \min\{c_{j_k}^x, t_l^k/2\}$
13: Update $\boldsymbol{x}^k = \boldsymbol{x}^{k-1} + \delta_k(\boldsymbol{a}_{i_{\min}} - \boldsymbol{a}_{j_k})$
14: Update $t_l^{k+1} = t_l^k - 2\delta_k$
15: **end while**
16: **Return:** $\boldsymbol{x}^k$

$\boldsymbol{a}_{i_{\min}}$, i.e., the one with the smallest product $\boldsymbol{z}^T\boldsymbol{a}$ (line 3), until it hits the boundary of one the two polytopes (lines 6 - 7). If the boundary reached is of $t^\star P_{\mathcal{A}} + \boldsymbol{u}$, we are done. Otherwise, we are at the boundary of $\lambda P_{\mathcal{A}}$.

The algorithm then improves on the solution by moving the largest amount of weight, which will not violate the constraints, from other active atoms to $\boldsymbol{a}_{i_{\min}}$, starting from the least beneficial active atom in terms of their product with $\boldsymbol{z}$. The algorithm stops when it runs out of active atoms or it reaches $\|\boldsymbol{x} - \boldsymbol{u}\|_{\mathcal{A}} = t^\star$ (lines 10 -15). Note the similarity with Away step FW [LJJ15], which only reduces the weight of the worst active atom.

Note that Algorithm 2 actually minimizes the objective along the path of possible values of $t^\star = \|\boldsymbol{x}^\star - \boldsymbol{u}\|_{\mathcal{A}}$ from $t = 0$ to $t = t_{\max}$. Indeed, the iterates satisfy $\boldsymbol{x}^k \in \mathrm{lconj_g}(\boldsymbol{u}, \boldsymbol{z}, t_u^k), \forall k$, where t_u^k (budget used) and t_l^k (budget left) keep track, respectively, of how far we are from $\boldsymbol{u}$, $\|\boldsymbol{x}^k - \boldsymbol{u}\|_{\mathcal{A}} = t_u^k$ and how far we "guess" we are from the boundary of $t^\star P_{\mathcal{A}} + \boldsymbol{u}$, where the guess of $Lt^\star$ corresponds to the current slope of $h(t_u^k)$. Unlike the general case where we are computing $h(t)$ using a black box optimizer, we actually can compute explicitly the slopes of the different pieces of $h(t)$, given by $\boldsymbol{z}^T\boldsymbol{a}_{i_{\min}}, 0.5(\boldsymbol{z}^T\boldsymbol{a}_{i_{\min}} - \boldsymbol{z}^T\boldsymbol{a}_{j_1}), 0.5(\boldsymbol{z}^T\boldsymbol{a}_{i_{\min}} - \boldsymbol{z}^T\boldsymbol{a}_{j_2}), \cdots, 0.5(\boldsymbol{z}^T\boldsymbol{a}_{i_{\min}} - \boldsymbol{z}^T\boldsymbol{a}_{j_d})$.

Proposition 15. *Algorithm 2 returns $\boldsymbol{x} \in \mathrm{prox}_g^{\mathcal{A}}(\boldsymbol{u}, \boldsymbol{z}, L)$ in $O(d\mathcal{T} + d\log d)$ time, where $\mathcal{T}$ is the time to compute $\boldsymbol{z}^T\boldsymbol{a}$ for any atom $\boldsymbol{a} \in \mathcal{A}$.*

Sketch of Proof Assuming $t^\star$ is guessed correctly, then if the maximal feasible step $\delta_0 = t^\star$, $\boldsymbol{x}^0$ is optimal. Otherwise $\|\boldsymbol{x}^0\|_{\mathcal{A}} = \lambda$ and there exists an optimal solution $\boldsymbol{x}^\star$ s.t. $\|\boldsymbol{x}^\star\|_{\mathcal{A}} = \lambda$

and $\|\boldsymbol{x}^\star - \boldsymbol{x}^0\|_{\mathcal{A}} \leq t^\star - \delta_0$. Then by Lemma 11, we can now solve instead: $\min_{\boldsymbol{c}^x \geq 0}\{\tilde{\boldsymbol{z}}^T \boldsymbol{c}^x : \mathbb{1}^T \boldsymbol{c}^x = \lambda, \|\boldsymbol{c}^x - \boldsymbol{c}^{x^0}\|_1 \leq t\}$ where $\tilde{\boldsymbol{z}}_i = \boldsymbol{z}^T \boldsymbol{a}_i$. This has been considered by [GH16], to obtain a local linear minimization oracle. The rest of our algorithm, i.e., after entering the for loop on line 10, reduces to theirs. We refer the reader to their proof of correctness [GH16, Lemma 5.2]. The correctness of the search for $t^\star$ follows from the correctness of this greedy approach. Finally, it is clear that the most expensive step in Algorithm 2 is the sorting operation on line 5, and hence its time complexity is $O(d\mathcal{T} + d \log d)$.

Remark 4. *If $g(\boldsymbol{x}) = \lambda \|\boldsymbol{x}\|_{\mathcal{A}}$ where $\mathcal{A} := \{\boldsymbol{a}_1, \cdots, \boldsymbol{a}_{2m}\}$, $(\boldsymbol{a}_i)_1^m$'s are linearly independent. Its Fenchel conjugate is given by $g^*(\boldsymbol{x}) = \iota_{\{\|\cdot\|_{\mathcal{A}^*} \leq \lambda\}}(\boldsymbol{x})$, where $\|\cdot\|_{\mathcal{A}^*}$ is the dual norm of $\|\cdot\|_{\mathcal{A}}$, then $\operatorname{prox}_g^{\mathcal{A}}$ can be obtained by computing $\operatorname{prox}_{g^*}^{\mathcal{A}^*}$ via Algorithm 2 and applying Proposition 14.*

Note that Algorithm 2 only adds *one* atom to the set of active atoms of $\boldsymbol{u}$ and possibly remove others, hence the corresponding iterates in GPM retain "sparsity".

4.4 Accelerated generalized proximal gradient method

In this section, we present an accelerated variant of GPM in Algorithm 3 and show that it has the same convergence rate as fast Euclidean proximal gradient methods, such as FISTA [BT09a].

The literature is vast on how to accelerate first order methods in non-Euclidean setting [Nes05, Tse08, Lan12, Aho16, AZO14, WWJ16]. However, unlike accGPM, these schemes require the computation of an additional proximal/projection operation with respect to a strongly convex prox-function D in each iteration. Similar to the classical fast methods, accGPM introduces a momentum term. However, a novel term $\mathbf{p}^k$ in line 10 of accGPM is essential in our analysis.

Algorithm 3 Accelerated proximal gradient method

1: **Input:** $L > 0$, $\mu > 0$, $\boldsymbol{x}^0 \in \mathbb{R}^d$, $\beta_0 > 0$.
2: **Initialization:** $\boldsymbol{w}^0 = \boldsymbol{x}^0$, $\boldsymbol{y}^0 = \boldsymbol{x}^0$.
3: **for** $k = 0, 1, \ldots$ **do**
4: $\quad \gamma_k \in (0, 1/L]$, $\alpha_k = \frac{1}{2}\big(\sqrt{\beta_k^2\gamma_k^2 + 4\beta_k\gamma_k} - \beta_k\gamma_k\big)$, $\beta_{k+1} = (1-\alpha_k)\beta_k + \alpha_k\tau_k\mu$
5: $\quad \boldsymbol{y}^{k+1} = (1-\alpha_k)\boldsymbol{x}^k + \alpha_k \boldsymbol{w}^k$
6: $\quad \boldsymbol{x}^{k+1} \in \operatorname{prox}(\boldsymbol{y}^{k+1}, \nabla f(\boldsymbol{y}^{k+1}), 1/\gamma_k)$
7: $\quad$ **if** $\boldsymbol{x}^{k+1} = \boldsymbol{y}^{k+1}$ **then**
8: $\quad\quad$ stop
9: $\quad$ **end if**
10: $\quad \mathbf{p}^k \in -\partial\big(\frac{1}{2\gamma_k}\|\cdot\|^2\big)(\boldsymbol{x}^{k+1} - \boldsymbol{y}^{k+1}) \cap \big(\nabla f(\boldsymbol{y}^{k+1}) + \partial g(\boldsymbol{x}^{k+1})\big)$
11: $\quad \boldsymbol{w}^{k+1} = \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} e_{k+1}(\boldsymbol{x})$
12: **end for**
13: **Return:** $\boldsymbol{x}^{k+1}$

Computation of $\mathbf{p}^k$: When $g = 0$, this term reduces to the gradient of f; $\mathbf{p}^k = \nabla f(\boldsymbol{y}^{k+1})$. When $\|\cdot\|^2$ or $g(\boldsymbol{x})$ is differentiable, $\mathbf{p}^k$ is unique. In general, since the subdifferential of any norm can be described by $\partial\|\boldsymbol{x}\| = \{\boldsymbol{z} : \boldsymbol{z}^T\boldsymbol{x} = \|\boldsymbol{x}\|, \|\boldsymbol{z}\|_* \leq 1\}$, then if g and $\|\cdot\|$ are atomic norms, $\mathbf{p}^k$ can be computed via a linear feasibility problem. In Sections 4.7.5 and 4.7.6, we show specifically how to compute $\mathbf{p}^k$ for the examples used in the numerical experiments; i.e, for the ℓ_1-norm and ℓ_∞-latent group Lasso norm.

A non-Euclidean projected gradient algorithm solving the special case of Problem (4.1), where $g = \iota_{\mathcal{X}}$ for a convex set $\mathcal{X}$, was proposed in [Nes05]. The analysis of this scheme is based on the concept of estimate sequences (cf., [Bae09, Nes04]). Algorithm 3 solves Problem (4.1) in the general setting, by constructing a novel estimate sequence e_k defined as follows.

Definition 15. *Let $(\alpha_k)_{k\in\mathbb{N}}$, $(\tau_k)_{k\in\mathbb{N}}$ and $(\gamma_k)_{k\in\mathbb{N}}$ be sequences in $(0, +\infty)$ and let $(\boldsymbol{x}^k)_{k\in\mathbb{N}}$, $(\boldsymbol{y}^k)_{k\in\mathbb{N}}$ and $(\mathbf{p}^k)_{k\in\mathbb{N}}$ be sequences in $\mathbb{R}^d$. We define the estimate sequence e_k recursively, with $e_0 := \dfrac{\beta_0}{\sigma}D + F(\boldsymbol{x}^0)$ and $e_{k+1} := (1-\alpha_k)e_k + \alpha_k\big((1-\tau_k)\psi_k + \tau_k\phi_k\big)$, where*

$$\psi_k := F(\boldsymbol{x}^{k+1}) + \left\langle \cdot - \boldsymbol{x}^{k+1}, \mathbf{p}^k \right\rangle - \frac{1}{2\gamma_k}\|\boldsymbol{x}^{k+1} - \boldsymbol{y}^{k+1}\|^2$$

and

$$\phi_k := f(\boldsymbol{y}^{k+1}) + \left\langle \cdot - \boldsymbol{y}^{k+1}, \nabla f(\boldsymbol{y}^{k+1}) \right\rangle + g.$$

The prox-function D is σ-strongly convex with respect to $\|\cdot\|$ and $\boldsymbol{x}^0 = \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} D(\boldsymbol{x})$, assuming without loss of generality that $D(\boldsymbol{x}^0) = 0$.

Note that the parameter τ_k allows us to choose between ψ_k and ϕ_k, depending on which is more suitable to the problem at hand. The estimate sequence resulting from ϕ_k ($\tau_k = 1$) is a direct extension of the one considered in [Nes05]. If g is strongly convex, this type of estimate sequence is preferable as it can exploit strong convexity, leading to a linear rate (see Theorem 5). However, this approach requires the minimization of a proximal-type subproblem involving the strongly-convex function D (line 11 in Algorithm 3). In fact, if $D = \frac{1}{2}\|\cdot\|_2^2$, this subproblem reduces to the Euclidean proximal operator of g.

On the other hand, choosing instead the novel estimate sequence resulting from ψ_k ($\tau_k = 0$) avoids such expensive subroutine. In this case, the minimization problem at line 11 is an instance of the Fenchel conjugate operator of D, which is usually easy to compute. For example, if $D = \frac{1}{2}\|\cdot\|_q^2, 1 < q \leq 2$, $\boldsymbol{w}^{k+1}$ can be computed in closed-form solution.

Theorem 5 (Convergence[3]). *Given g which is μ-strongly convex w.r.t. $\|\cdot\|$. If accGPM terminates at iteration k, i.e., $\boldsymbol{x}^{k+1} = \boldsymbol{y}^{k+1}$, then $\boldsymbol{x}^{k+1}$ is a solution to* (4.1). *Otherwise, let $x^\star \in \mathscr{X}^\star$, the iterates of accGPM satisfy the following.*

[3]This theorem and its proof are due primarily to B. Vu.

1. *If* $\mu = 0$. *Then* $\forall k \in \mathbb{N}$, *we have* $F(\boldsymbol{x}^{k+1}) - F^\star \leq \dfrac{4\big(\sigma(F(\boldsymbol{x}^0) - F^\star) + \beta_0 D(\boldsymbol{x}^\star)\big)}{\sigma\{2 + \sqrt{\beta_0}\sum_{i=0}^{k}\sqrt{\gamma_i}\}^2}$.

   Consequently, if $\forall k \in \mathbb{N}, \gamma_k = 1/L$, *then* $F(\boldsymbol{x}^{k+1}) - F^\star \leq \dfrac{4L\big(\sigma(F(\boldsymbol{x}^0) - F^\star) + \beta_0 D(\boldsymbol{x}^\star)\big)}{\sigma\{2\sqrt{L} + \sqrt{\beta_0}(k+1)\}^2}$.

2. *If* $\mu > 0$. *Set* $\tau = \inf_{k\in\mathbb{N}} \tau_k$, *and* $\rho = \frac{\tau\mu}{2L}\left\{\sqrt{1 + \frac{4L}{\beta_0+\mu}} - 1\right\}$. *If* $\beta_0 \geq \tau\mu$ *and* $\forall k \in \mathbb{N}, \gamma_k = 1/L$, *then we have* $F(\boldsymbol{x}^{k+1}) - F^\star \leq (1-\rho)^{k+1}\{F(\boldsymbol{x}^0) - F^\star + \frac{\beta_0}{\sigma} D(\boldsymbol{x}^\star)\}$.

Note that the choice of the norm in accGPM affects the Lipschitz constant L, and the strong convexity constant μ as in GPM, but also affects implicitly the term $D(\boldsymbol{x}^\star)/\sigma$.

4.5 Experiments

The purpose of this numerical section is to demonstrate how choosing a non-Euclidean norm in GPM leads in some cases to faster convergence, and in others to easier-to-solve proximal operators. To that end, we consider in Section 4.5.1, the classical Lasso problem [Tib96] and illustrate how ℓ_1-GPM improves the convergence rate, in a sparse setting. Then, in Section 4.5.2, we consider the latent group Lasso problem [OJV11] and illustrate how our results allow us to compute the non-Euclidean proximal operator of the ℓ_∞-latent group Lasso norm, significantly faster than state-of-the-art methods computing the corresponding Euclidean proximal operator.

4.5.1 Lasso

In this section, we consider the classical Lasso problem [Tib96]:

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} \frac{1}{2n}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{x}\|_1.$$

We propose to solve it with ℓ_1-GPM, i.e., with $\mathrm{prox}_{\ell_1}^{\ell_1}$. The motivation for this choice is two folds. The resulting iterates from $\mathrm{prox}_{\ell_1}^{\ell_1}$ are *sparse* (see Section 4.3.2) which is naturally preferred in this set-up. Also, the convergence rate of GPM is better when the ℓ_1-norm is chosen, as opposed to the ℓ_2-norm, in the sparse settings. Indeed, recall from Theorem 4, that GPM converges in $O(\frac{L_1\|\boldsymbol{x}^\star\|_1^2}{k})$ with the ℓ_1-norm, and in $O(\frac{L_2\|\boldsymbol{x}^\star\|_2^2}{k})$ with the ℓ_2-norm. In this case, $L_1 = \max_{i,j} |[\boldsymbol{A}^T\boldsymbol{A}]_{ij}|$ and $L_2 = \sigma_{\max}(\boldsymbol{A})^2$, where $\sigma_{\max}(\boldsymbol{A})$ is the largest singular value of $\boldsymbol{A}$. Hence, if $\boldsymbol{A}$ is a dense matrix, we can have $L_1 \ll L_2$, while $\|\boldsymbol{x}^\star\|_1 \simeq \|\boldsymbol{x}^\star\|_2$ when $\boldsymbol{x}^\star$ is sparse. Moreover, the *restricted strong convexity* parameter, which governs the learning quality of Lasso problems is also known to be better w.r.t. the ℓ_1-norm vs the ℓ_2-norm [VDGB09], implying faster convergence also in the estimation error. Our experiments verify these observations.

Unfortunately, some of these benefits are lost in the accelerated variant ℓ_1-accGPM, where the iterates are no longer sparse, and the strong convexity requirement on the prox-function

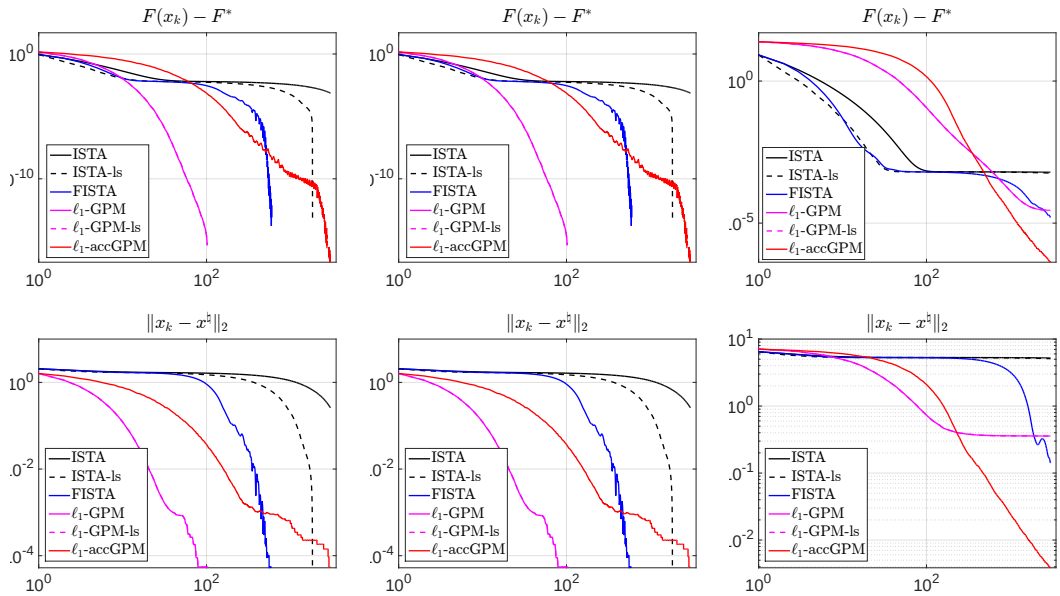


Figure 4.2: (1st row) Objective error and (2nd row) estimation error, with $d = 1000, n = 400$: (*Left*) $s = 10$, (*Middle*) $s = 50$, (*Right*) $s = 100$.

D in Definition 15, introduces a dimension-dependent term in the convergence rate. We use $D(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x}\|_{1+\epsilon}^2$ as the prox-function. This function is strongly convex in the ℓ_1-norm, when $\epsilon > 0$, with $\sigma = \epsilon/d^{\frac{2\epsilon}{1+\epsilon}}$. We choose $\epsilon = \log d - 1 - \sqrt{(\log d - 1)^2 - 1}$, to maximize σ. The convergence rate of ℓ_1-accGPM in this case is $O(\frac{L_1 \log^2(d)\|\boldsymbol{x}^\star\|_{1+\epsilon}^2}{k^2})$ by Theorem 5.

The standard $\mathrm{prox}_{\ell_1}^{\ell_2}$ can be computed in $O(d)$ using the soft thresholding operator [DJ95]. By Remark 4, the decomposition in Proposition 14 can be exploited to compute $\mathrm{prox}_{\ell_1}^{\ell_1}$ in $O(d \log d)$ time with the greedy Algorithm 2. We choose instead to solve it directly via another greedy algorithm, of the same "flavor" as Algorithm 2, presented in the Appendix. The momentum $\mathbf{p}^k$ for accGPM has a closed form solution in this case, also given in the Appendix.

We synthetically set up a linear model $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}^\natural + \boldsymbol{\varepsilon}$, where $\boldsymbol{x}^\natural$ is an s-sparse vector with normalized ℓ_1-norm. $\boldsymbol{A} \in \mathbb{R}^{n\times d}$ is an i.i.d Gaussian matrix and $\boldsymbol{\varepsilon}$ an i.i.d. Gaussian noise vector of variance σ^2 where $\sigma = 10^{-4}$. We fix $d = 1000$, $n = 400$, and vary the sparsity level s from 10 to 100. The number of samples is chosen to exceed the sample complexity [NRW+12], while approaching to the statistical phase transition as sparsity increases. The regularization parameter is set to $\lambda = \sigma\sqrt{\frac{\log d}{n}}$ according to the theory of [NRW+12].

We compare ISTA and FISTA to ℓ_1-GPM and ℓ_1-accGPM (with $\tau = 0$). Figure 4.2 plots (in log scale) the objective error and estimation error, in the different sparsity setups. We use an accuracy based stopping condition with `tol`$= 10^{-9}$ where the optimal objective value is obtained by the convex solver CVX [GB14]. We also use a 3000 iteration limit. We equip both ISTA and

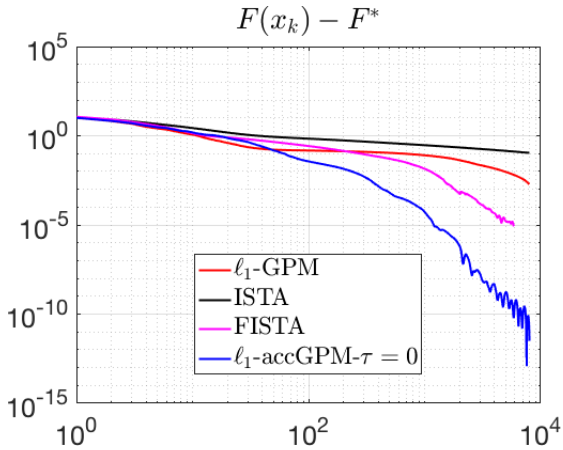


Figure 4.3: Objective error on the LEUKEMIA dataset.

Table 4.1: Running time (in sec) of $\mathrm{prox}_G^{\ell_2}$ (LHS) and $\mathrm{prox}_G^{\ell_\infty}$ + $\mathbf{p}_k$ (RHS), averaged over 10 runs.

	d = 64		d = 128		d = 256		d = 512	
tol= 10^{-2}	**0.055**	0.055 + 0.003	**0.103**	0.137+ 0.005	**0.192**	0.247 + 0.009	**0.461**	0.714 + 0.016
tol= 10^{-3}	0.502	**0.101 + 0.003**	0.944	**0.149 + 0.004**	2.038	**0.360 + 0.008**	4.213	**1.276 + 0.013**
tol= 10^{-4}	5.234	**0.252 + 0.006**	9.422	**0.203 + 0.004**	18.92	**0.460 + 0.006**	41.21	**1.857 + 0.016**
tol= 10^{-5}	42.62	**0.214 + 0.005**	98.13	**0.428 + 0.009**	170.6	**0.614 + 0.006**	377.5	**1.487 + 0.015**

ℓ_1-GPM with line-search.

ℓ_1-accGPM has a similar performance to FISTA, and is slower than its non-accelerated variant in this setting, due to the dimension-dependence of σ. Otherwise, a clear advantage of sparse updates in the sparse regime can be inferred from the leftmost pair, where ℓ_1-GPM significantly outperforms ISTA/FISTA. As the sparsity level increases, the benefits of sparse updates vanish (mid pair), and around the phase transition classical gradient methods perform better.

In the above setting, the reduction in the Lipschitz constant seems to be offset by the $\log^2(d)$ term in the convergence rate of ℓ_1-accGPM. We consider now another set-up, where we compare the same methods on the real dataset LEUKEMIA from LIBSVM [Cha00], which contains $n = 38$ samples and $d = 7129$ features. We use the same stopping conditions as before. The convergence behavior is depicted in Figure 4.3. In this case, we can see a clear advantage of using ℓ_1-accGPM.

4.5.2 Latent group Lasso

In this section, we consider the latent group Lasso (LGL) problem:

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2 + \lambda\|\boldsymbol{x}\|_{\mathfrak{G}},$$

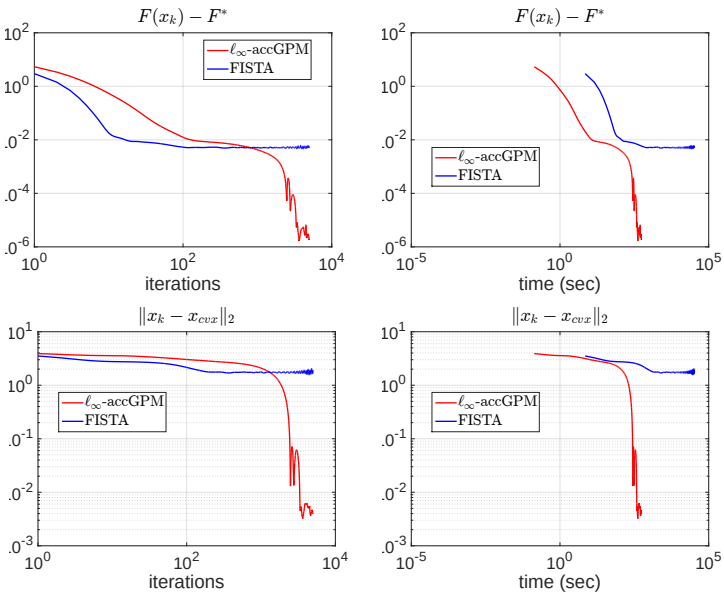


Figure 4.4: Objective error (top) and optimization error (bottom), for $d = 100, n = 50, s = 2$.

where $\|\boldsymbol{x}\|_{\mathfrak{G}}$ is the LGL-norm, proposed by [OJV11] to induce supports that corresponds to union of groups (see Section 1.3.1). Given a collection of groups $\mathfrak{G} = \{G_1, \cdots, G_M\}$, the ℓ_p-LGL norm is given by $\|\boldsymbol{x}\|_{\mathfrak{G}} = \min_{\boldsymbol{v}}\{\sum_{i=1}^M \|\boldsymbol{v}_{G_i}\|_p : |\boldsymbol{x}| = \sum_{i=1}^M v_{G_i}, \mathrm{supp}(\boldsymbol{v}_{G_i}) \subseteq G_i\}$. It is known that ℓ_p-LGL is an atomic norm, with atoms $\mathcal{A} = \{\boldsymbol{v} \in \mathbb{R}^d : \mathrm{supp}(\boldsymbol{v}_{G_i}) \subseteq G_i, \|\boldsymbol{v}_{G_i}\|_p \leq 1\}$ [OJV11]. We focus on the case $p = \infty$ with finitely many atoms. Recall that ℓ_∞-LGL is the convex envelope of the set cover function over the unit ℓ_∞-ball (see Sections 1.3.4 and 2.5.1).

As the goal here is to compare the time complexity of the proximal operator of the LGL norm, with a Euclidean vs. a non-Euclidean norm, we do not optimize the choice of the non-Euclidean norm to achieve the best convergence rate, and simply choose $\|\cdot\|$ to be the ℓ_∞-norm.

The atoms in $\mathcal{A}$ are not linearly independent, thus we cannot use the greedy Algorithm 2. Nevertheless, ℓ_∞-LGL is a polyhedral function, as required in Section 4.3.1. Hence, its proximal operator $\mathrm{prox}_G^{\ell_\infty}$, with respect to the ℓ_∞-norm, can be computed via Algorithm 1, and its $\mathbf{p}_k$ can be computed by a feasibility LP, given in the Appendix.We use Gurobi [Gur16] to solve the resulting LPs. We choose $D(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x}\|_2^2$ as the prox-function.

To the best of our knowledge, the only available approaches to compute the Euclidean proximal operator of ℓ_∞-LGL, i.e., $\mathrm{prox}_G^{\ell_2}$, is either via duplicating the variables in the overlapping groups, which is very inefficient for groups with substantial overlap, or via the cyclic projections approach proposed in [VRMV14], which is guaranteed to converge but with no convergence rate guarantees.

We first assess the time complexity of the proximal operator $\mathrm{prox}_G^{\ell_\infty}$ vs $\mathrm{prox}_G^{\ell_2}$. We fix the size of the groups to $|G_i| = 10$ and generate $M = 2.5d/10$ (to ensure substantial overlap) groups with randomly selected elements. The input $\boldsymbol{u} \in \mathbb{R}^d$ is generated as a random Gaussian vector. For

fairness, we set $\lambda = 0.8 \min_i \|\boldsymbol{u}_{G_i}\|_1$ to ensure all groups are active. We report in Table 4.1 the CPU time (in sec) of $\mathrm{prox}_G^{\ell_\infty}$ and $\mathrm{prox}_G^{\ell_2}$, as we vary the dimension d from 64 to 512 and the accuracy `tol` from 10^{-2} to 10^{-5}, where a true solution is obtained by the convex solver CVX [GB14]. $\mathrm{prox}_G^{\ell_\infty}$ provides up to 300× speed up, and the cost of computing $\mathbf{p}^k$ is negligible.

To assess if the slow performance of $\mathrm{prox}_G^{\ell_2}$ is compensated by a better convergence rate, we compare the performance of FISTA to ℓ_∞-accGPM on a synthetic learning problem, where the true vector $\boldsymbol{x}^\natural$ is given by the union of $s = 2$ randomly selected groups. We follow otherwise the same setup as in Section 4.5.1, with $d = 100, n = 50$ and the groups generated as before. We stop both proximal algorithms after 10^5 iterations, or when the distance between iterates reaches a precision, initialized to 10^{-5} and decreased linearly with iterations. For the outer algorithms, we use an accuracy based stopping condition with `tol`$= 10^{-9}$ where the optimal objective value is obtained by CVX. We also use a 5000 iteration limit. We choose the regularization parameter λ that yields the best performance on the CVX solution. Figure 4.4 plots (in logscale) the objective error and optimization error. FISTA indeed has a better convergence rate in this case, but this is undermined by the slow performance of $\mathrm{prox}_G^{\ell_2}$. Indeed, with the set iteration limit, $\mathrm{prox}_G^{\ell_2}$ is not able to reach the requested precision, and thus FISTA does not converge to the true solution.

4.6 Discussion

We presented two algorithms to compute the non-Euclidean proximal operator in GPM, for polyhedral regularizers and norms: One general polynomial-time method (Algorithm 1), which requires computing a logarithmic number of linear minimization problems over the intersection of two polytopes. The other is an efficient greedy method (Algorithm 2) which only applies to special cases where the regularizer is an atomic norm with linearly independent atoms. The cost of updating the iterates of GPM via this greedy method is almost as cheap as Frank-Wolfe (FW) iterates, and the resulting iterates are sparse; a very attractive property specially in the context of structured sparsity.

We also introduced an accelerated variant of GPM, which only requires an additional feasibility LP in general, as opposed to the additional proximal/projection operation typically required by other acceleration schemes. We showcased the benefit of using non-Euclidean norms in GPM numerically on two structured-sparsity examples, showing significant speed-up over state-of-the-art methods.

The discussion in this chapter raises the following open questions:

Open question 6. The general Algorithm 1, though only requiring linear minimization problems, can still be expensive in general; limiting the applicability of this approach. *Can we extend the greedy Algorithm 2 to handle any polyhedral function?*

A possible promising approach to achieve this is to leverage the analysis from [GH16], which considers a related problem to the local Fenchel conjugate operator (4.7) we use, namely a local

linear minimization oracle. In particular, in [GH16] the authors reduce the general polytope case of their problem to the simpler case of the simplex constraint, but loose a factor of d in their convergence rate in the process. It is worth investigating a similar reduction for the local Fenchel conjugate operator. We expect the factor d to be replaced by a factor ρ, relating the norm $\|\cdot\|$ chosen in GPM to the ℓ_1-norm, such that $\|\cdot\|_1^2 \leq \rho\|\cdot\|^2$.

Such extension would result in a proximal gradient method with an attractive trade-off: It would enjoy cheap and sparse iterates, which are only slightly more expensive than a linear minimization oracle, and a fast convergence rate, optimal in terms of iteration-dependence, via the accelerated variant we presented in this chapter.

Open question 7. One drawback to the acceleration method accGPM and any other acceleration method, is the requirement that the prox-function D should be *strongly convex*. This restriction on the choice of D then introduces dimension dependent terms in the convergence rate, which offset the benefit of choosing a suitable norm, as observed in Section 4.5.1. In the worst case, the dimension dependence can reach up to a factor d, if the chosen norm is the ℓ_∞-norm; this is known as the ℓ_∞- barrier [She17]. *Is this dimension-dependence necessary or it can be avoided by better acceleration schemes?*

A weaker notion than strong convexity, called *area convexity*, was introduced in a recent work [She17] considering bilinear saddle point problems. In this work, a modified version of Nesterov's dual extrapolation algorithm [Nes07] is presented, which only requires area-convexity for convergence. This approach is successfully applied to accelerate maxflow problems, without suffering any increase in dimension dependence. Such result suggest then that the dimension-dependence might not be necessary. It is worth then investigating acceleration schemes of GPM, under this weaker area convexity assumption on the prox-function D. Designing area-convex prox-functions suitable for structured sparsity problems is also needed to be able to apply this approach.

4.7 Appendix: Proofs

4.7.1 Proof of Theorem 4

Theorem 4. *The iterates $\boldsymbol{x}^k$ of GPM satisfy $\forall k \in \mathbb{N}$:*

$$F(\boldsymbol{x}^k) - F^\star \leq \frac{2\max\{L\mathcal{R}(\boldsymbol{x}^0), F(\boldsymbol{x}^0) - F^\star\}}{k}$$

where $\mathcal{R}(\boldsymbol{x}^0) = \max_{\{\boldsymbol{x}:F(\boldsymbol{x})\leq F(\boldsymbol{x}^0)\}} \min_{\boldsymbol{x}^\star\in\mathcal{X}^\star} \|\boldsymbol{x} - \boldsymbol{x}^\star\|^2$. If, in addition, $f(\boldsymbol{x})$ is μ-strongly convex w.r.t. norm $\|\cdot\|$, then GPM satisfies

$$F(\boldsymbol{x}^k) - F^\star \leq (1 - \frac{\mu}{L})^k \big(F(\boldsymbol{x}^0) - F^\star\big).$$

Proof. Without loss of generality, we assume that f is μ-strongly convex with $\mu \in [0, +\infty)$ ($\mu = 0$ corresponds to the case where f is convex). Fix $\boldsymbol{x}^\star$ a minimizer of F such that $\max_{\boldsymbol{x}:F(\boldsymbol{x})\leq F(\boldsymbol{x}^0)} \|\boldsymbol{x} - \boldsymbol{x}^\star\| \leq \mathcal{R}(\boldsymbol{x}^0)$, and $k \in \mathbb{N}$. If $\boldsymbol{x}^k$ is a minimizer of F then the claims are trivial. Otherwise, let us define

$$(\forall \boldsymbol{x} \in \mathbb{R}^d) \quad Q(\boldsymbol{x}, \boldsymbol{x}^k) = f(\boldsymbol{x}^k) + \Big\langle \boldsymbol{x} - \boldsymbol{x}^k, \nabla f(\boldsymbol{x}^k)\Big\rangle + g(\boldsymbol{x}) + \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{x}^k\|^2. \tag{4.10}$$

Then

$$\boldsymbol{x}^{k+1} \in \mathrm{prox}_{\mathrm{g}}(\nabla f(\boldsymbol{x}^k), \boldsymbol{x}^k, L) = \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} Q(\boldsymbol{x}, \boldsymbol{x}^k). \tag{4.11}$$

Since the gradient f is L-Lipschitz continuous,

$$f(\boldsymbol{x}^{k+1}) \leq f(\boldsymbol{x}^k) + \Big\langle \boldsymbol{x}^{k+1} - \boldsymbol{x}^k, \nabla f(\boldsymbol{x}^k)\Big\rangle + \frac{L}{2}\|\boldsymbol{x}^{k+1} - \boldsymbol{x}^k\|^2 \tag{4.12}$$

and hence (4.11) yields

$$F(\boldsymbol{x}^k) = Q(\boldsymbol{x}^k, \boldsymbol{x}^k) \geq Q(\boldsymbol{x}^{k+1}, \boldsymbol{x}^k) \geq F(\boldsymbol{x}^{k+1}). \tag{4.13}$$

By strongly convexity of f we have,

$$(\forall \boldsymbol{x} \in \mathbb{R}^d) \quad f(\boldsymbol{x}^k) + \Big\langle \boldsymbol{x} - \boldsymbol{x}^k, \nabla f(\boldsymbol{x}^k)\Big\rangle \leq f(\boldsymbol{x}) - \frac{\mu}{2}\|\boldsymbol{x} - \boldsymbol{x}^k\|^2. \tag{4.14}$$

Also by strongly convexity of F and by lemma 13 in [SSS07] we have

$$(\forall \alpha \in [0, 1]) \quad F(\alpha\boldsymbol{x}^\star + (1-\alpha)\boldsymbol{x}^k) \leq \alpha F(\boldsymbol{x}^\star) + (1-\alpha)F(\boldsymbol{x}^k) - \frac{\alpha(1-\alpha)\mu}{2}\|\boldsymbol{x}^\star - \boldsymbol{x}^k\|^2. \tag{4.15}$$

It hence follows from (4.11), (4.14), and (4.15) that

$$\begin{aligned}
Q(\boldsymbol{x}^{k+1}, \boldsymbol{x}^k) &= \min_{\boldsymbol{x}\in\mathbb{R}^d} \; f(\boldsymbol{x}^k) + \Big\langle \boldsymbol{x} - \boldsymbol{x}^k, \nabla f(\boldsymbol{x}^k) \Big\rangle + g(\boldsymbol{x}) + \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{x}^k\|^2 \\
&\leq \min_{\boldsymbol{x}\in\mathbb{R}^d} \; F(\boldsymbol{x}) + \frac{L-\mu}{2}\|\boldsymbol{x} - \boldsymbol{x}^k\|^2 \\
&\leq \min_{\alpha\in[0,1]} \; F(\alpha\boldsymbol{x}^\star + (1-\alpha)\boldsymbol{x}^k) + \frac{(L-\mu)\alpha^2}{2}\|\boldsymbol{x}^\star - \boldsymbol{x}^k\|^2 \\
&\leq \min_{\alpha\in[0,1]} \; \alpha F(\boldsymbol{x}^\star) + (1-\alpha)F(\boldsymbol{x}^k) - \frac{\alpha(1-\alpha)\mu}{2}\|\boldsymbol{x}^\star - \boldsymbol{x}^k\|^2 + \frac{(L-\mu)\alpha^2}{2}\|\boldsymbol{x}^k - \boldsymbol{x}^\star\|^2. \\
&\leq \min_{\alpha\in[0,1]} \; F(\boldsymbol{x}^k) + \alpha\big(F(\boldsymbol{x}^\star) - F(\boldsymbol{x}^k)\big) - \frac{\alpha(1-\alpha)\mu - (L-\mu)\alpha^2}{2}\|\boldsymbol{x}^\star - \boldsymbol{x}^k\|^2.
\end{aligned} \tag{4.16}$$

For $\mu = 0$, the function in (4.16) admits a minimizer at

$$\alpha_k^\star = \min\{\frac{F(\boldsymbol{x}^k) - F^\star}{L\|\boldsymbol{x}^k - \boldsymbol{x}^\star\|^2}, 1\} \in [0,1], \tag{4.17}$$

we deduce from (4.16) that

$$Q(\boldsymbol{x}^{k+1}, \boldsymbol{x}^k) - F^\star \leq \max\left\{1 - \frac{F(\boldsymbol{x}^k) - F^\star}{2L\|\boldsymbol{x}^k - \boldsymbol{x}^\star\|^2}, \frac{1}{2}\right\}\big(F(\boldsymbol{x}^k) - F^\star\big). \tag{4.18}$$

Let $\rho = 2\max\{L\mathcal{R}(\boldsymbol{x}^0), F(\boldsymbol{x}^0) - F^\star\}$. Consequently, (4.13) yields

$$F(\boldsymbol{x}^{k+1}) - F^\star \leq Q(\boldsymbol{x}^{k+1}, \boldsymbol{x}^k) - F^\star \leq \left(1 - \frac{F(\boldsymbol{x}^k) - F^\star}{\rho}\right)\big(F(\boldsymbol{x}^k) - F^\star\big). \tag{4.19}$$

Let $a_k = F(\boldsymbol{x}^k) - F^\star$. Since $a_k - a_{k+1} \geq \frac{a_k^2}{\rho}$, we obtain

$$\frac{1}{a_{k+1}} - \frac{1}{a_k} = \frac{a_k - a_{k+1}}{a_k a_{k+1}} \geq \frac{a_k^2}{\rho a_k^2} = \frac{1}{\rho}. \tag{4.20}$$

Consequently, $a_k \leq \frac{\rho}{k}$, which proves the first claim. For the second claim, we note that

$$(\forall \boldsymbol{x} \in \mathbb{R}^d) \quad \frac{\mu}{2}\|\boldsymbol{x} - \boldsymbol{x}^\star\|^2 \leq F(\boldsymbol{x}) - F^\star \tag{4.21}$$

and hence $\alpha_k^* = \dfrac{\mu}{L} \in (0,1]$. It then follows from (4.16) that $Q(\boldsymbol{x}^{k+1}, \boldsymbol{x}^k) \leq F(\boldsymbol{x}^k) - \alpha_k^*\big(F(\boldsymbol{x}^k) - F^\star\big)$, and hence,

$$F(\boldsymbol{x}^{k+1}) - F^\star \leq (1 - \alpha_k^*)\big(F(\boldsymbol{x}^k) - F^\star\big) = \left(1 - \frac{\mu}{L}\right)\big(F(\boldsymbol{x}^k) - F^\star\big). \tag{4.22}$$

□

4.7.2 Proof of Proposition 14 and Lemma 9

Proposition 14. *Generalized Moreau's decomposition Given $g \in \Gamma_0$ and its Fenchel g^*, we have*

$$\mathbf{p} - \boldsymbol{z} \in \mathrm{prox}^*_{\mathrm{g}^*}(-\boldsymbol{z}, -\boldsymbol{u}, 1/L) \text{ and } \boldsymbol{x}^\star - \boldsymbol{u} \in -\partial\big(\frac{L}{2}\|\cdot\|_*^2\big)(\mathbf{p}) \cap \big(-\boldsymbol{u} + \partial g(\mathbf{p} - \boldsymbol{z})\big) \quad (4.6)$$

where $\mathbf{p} \in -\partial\big(\frac{L}{2}\|\cdot\|^2\big)(\boldsymbol{x}^\star - \boldsymbol{u}) \cap \big(\boldsymbol{z} + \partial g(\boldsymbol{x}^\star)\big)$ *and* $\boldsymbol{x}^\star \in \mathrm{prox}_{\mathrm{g}}(\boldsymbol{u}, \boldsymbol{z}, L)$.

Proof. Recall that $\boldsymbol{y} \in \partial f(\boldsymbol{x}) \Leftrightarrow \boldsymbol{x} \in \partial f^*(\boldsymbol{y})$ for any $f \in \Gamma_0$ and its Fenchel conjugate f^*. Then since the Fenchel conjugate of $-\frac{L}{2}\|\cdot\|^2$ is given by $-\frac{1}{2L}\|\cdot\|_*^2$, we have

$$\begin{aligned}
\boldsymbol{x}^\star - \boldsymbol{u} &\in \partial\big(-\frac{L}{2}\|\cdot\|_*^2\big)(\mathbf{p}) \\
\boldsymbol{x}^\star &\in \partial g(\mathbf{p} - \boldsymbol{z}) \\
\Leftrightarrow \boldsymbol{x}^\star - \boldsymbol{u} &\in \partial\big(-\frac{L}{2}\|\cdot\|_*^2\big)(\mathbf{p} - \boldsymbol{z} + \boldsymbol{z}) \cap \big(-\boldsymbol{u} + \partial g(\mathbf{p} - \boldsymbol{z})\big) \\
\Leftrightarrow \mathbf{p} - \boldsymbol{z} &\in \mathrm{prox}^*_{\mathrm{g}^*}(-\boldsymbol{z}, -\boldsymbol{u}, 1/L)
\end{aligned}$$

□

Lemma 9. *Let* $h(t) = \min_{\|\boldsymbol{x}-\boldsymbol{u}\| \le t} \boldsymbol{z}^T\boldsymbol{x} + g(\boldsymbol{x})$ *and* $t^\star \in \frac{\partial h(t^\star)}{L}$ *then*

$$\boldsymbol{x}^\star \in \mathrm{prox}_{\mathrm{g}}(\boldsymbol{u}, \boldsymbol{z}, L) \Leftrightarrow \boldsymbol{x}^\star \in \operatorname*{arg\,min}_{\|\boldsymbol{x}-\boldsymbol{u}\| \le t^\star} \boldsymbol{z}^T\boldsymbol{x} + g(\boldsymbol{x}). \quad (4.7)$$

Proof. The two problems are related in the following way:

$$\begin{aligned}
&\min_{\boldsymbol{x} \in \mathbb{R}^d} \boldsymbol{z}^T\boldsymbol{x} + \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{u}\|^2 + g(\boldsymbol{x}) \\
&= \min_{t \ge 0} \frac{L}{2}t^2 + \min_{\|\boldsymbol{x}-\boldsymbol{u}\| \le t} \boldsymbol{z}^T\boldsymbol{x} + g(\boldsymbol{x}) \\
&= \min_{t \ge 0} \frac{L}{2}t^2 + h(t)
\end{aligned}$$

The lemma follows by optimality conditions. □

4.7.3 Proof of Lemmas 10 and 11, and Proposition 15

Lemma 10. *Given* $\boldsymbol{x} = \sum_{i=1}^{2m} c_i^x \boldsymbol{a}_i, \boldsymbol{c}^x \ge 0$, *then* $\sum_{i=1}^{2m} c_i^x = \|\boldsymbol{x}\|_{\mathcal{A}} \Leftrightarrow \forall i, c_i^x = 0$ *or* $c_{i+m}^x = 0$.

Proof. Assume towards contradiction that $\exists i'$, such that $c_{i'}^x \neq 0, c_{i'+m}^x \neq 0$, then let $\tilde{c}_{i'}^x = c_{i'}^x - \min\{c_{i'}^x, c_{i'+m}^x\}, \tilde{c}_{i'+m}^x = c_{i'+m}^x - \min\{c_{i'}^x, c_{i'+m}^x\}$, which makes one of them zero and keep all other coefficients unchanged. Note then that $\boldsymbol{x} = \sum_{i=1}^{2m} \tilde{c}_i^x \boldsymbol{a}_i, \tilde{\boldsymbol{c}}^x \ge 0$ and $\mathbb{1}^T\tilde{\boldsymbol{c}}^x < \mathbb{1}^T\boldsymbol{c}^x =$

$\|\boldsymbol{x}\|_{\mathcal{A}}$ leading to a contradiction. The uniqueness follows from the linear independence of the atoms. The other direction follows from the uniqueness observation. □

Lemma 11. *Given $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, s.t $\boldsymbol{c}^x = \mathrm{MR}(\boldsymbol{x}), \boldsymbol{c}^y = \mathrm{MR}(\boldsymbol{y})$, we have $\|\boldsymbol{x}-\boldsymbol{y}\|_{\mathcal{A}} = \|\boldsymbol{c}^x - \boldsymbol{c}^y\|_1$.*

Proof. We can write $\boldsymbol{x}-\boldsymbol{y} = \sum_{i=1}^{2m} c_i^{x-y}\boldsymbol{a}_i$ where $\boldsymbol{c}^{x-y} = \mathrm{MR}(\boldsymbol{x}-\boldsymbol{y})$. By linear independence, we have $(c_i^x - c_{i+m}^x) - (c_i^y - c_{i+m}^y) = (c_i^{x-y} - c_{i+m}^{x-y})$. By lemma 10, we know that $\forall i$ either c_i^{x-y} or c_{i+m}^{x-y} is zero. It follows then that the other will be equal to $|(c_i^x - c_{i+m}^x) - (c_i^y - c_{i+m}^y)|$. Hence $\|(\boldsymbol{c}^x(1:m) - \boldsymbol{c}^y(1:m)) - (\boldsymbol{c}^x(m+1:2m) - \boldsymbol{c}^y(m+1:2m))\|_1 = \mathbb{1}^T\boldsymbol{c}^{x-y} = \|x-y\|_{\mathcal{A}}$.

By lemma 10, we only need to consider these cases:

c_i^x	c_{i+m}^x	c_i^y	c_{i+m}^y	$\lvert(c_i^x - c_{i+m}^x) - (c_i^y - c_{i+m}^y)\rvert$	$\lvert c_i^x - c_i^y\rvert + \lvert c_{i+m}^x - c_{i+m}^y\rvert$
>0	0	>0	0	$\lvert c_i^x - c_i^y\rvert$	$\lvert c_i^x - c_i^y\rvert$
>0	0	0	>0	$c_i^x + c_{i+m}^y$	$c_i^x + c_{i+m}^y$
0	>0	>0	0	$c_{i+m}^x + c_i^y$	$c_{i+m}^x + c_i^y$
0	>0	0	>0	$\lvert c_{i+m}^x - c_{i+m}^y\rvert$	$\lvert c_{i+m}^x - c_{i+m}^y\rvert$

Hence, $\|\boldsymbol{x}-\boldsymbol{y}\|_{\mathcal{A}} = \|(\boldsymbol{c}^x(1:m) - \boldsymbol{c}^y(1:m)) - (\boldsymbol{c}^x(m+1:2m) - \boldsymbol{c}^y(m+1:2m))\|_1 = \|\boldsymbol{c}^x(1:m) - \boldsymbol{c}^y(1:m)\|_1 + \|\boldsymbol{c}^x(m+1:2m) - \boldsymbol{c}^y(m+1:2m)\|_1 = \|\boldsymbol{c}^x - \boldsymbol{c}^y\|_1$. □

Algorithm 4 presents the general version of Algorithm 2 which can handle the case where $\|\boldsymbol{u}\|_{\mathcal{A}} > \lambda$. In the case where $\|\boldsymbol{u}\|_{\mathcal{A}} \leq \lambda$ and t is given, Algorithm 4 reduces to Algorithm 5.

Proposition 15. *Algorithm 2 returns $\boldsymbol{x} \in \mathrm{prox}_g^{\mathcal{A}}(\boldsymbol{u}, \boldsymbol{z}, L)$ in $O(d\mathcal{T} + d\log d)$ time, where $\mathcal{T}$ is the time to compute $\boldsymbol{z}^T\boldsymbol{a}$ for any atom $\boldsymbol{a} \in \mathcal{A}$.*

Proof. We know from lemma 9 that solving $\mathrm{prox}_g^{\mathcal{A}}(\boldsymbol{u}, \boldsymbol{z}, L)$ reduces to solving $\mathrm{lconj}_g^{\mathcal{A}}(\boldsymbol{u}, \boldsymbol{z}, t)$ with $t = t^\star$. We show first that given any $t \geq 0$ Algorithm 5 indeed returns $x^k \in \mathrm{lconj}_g^{\mathcal{A}}(\boldsymbol{u}, \boldsymbol{z}, t)$. Making use of lemma 10 and 11, we make the following observations:

- $\delta_0 = \max_{\delta>0}\{\delta : \boldsymbol{u} + \delta\boldsymbol{a}_{j_{\min}} \in \lambda\,\mathrm{conv}(\mathcal{A}) \cap (t\,\mathrm{conv}(\mathcal{A}) + \boldsymbol{u})\}$.
  To see this note that for any $\delta > 0$ s.t. $\boldsymbol{x} = \boldsymbol{u} + \delta\boldsymbol{a}_{j_{\min}}$ is feasible, we need to have $\|\boldsymbol{x}-\boldsymbol{u}\|_{\mathcal{A}} = \delta \leq t$ and $\|\boldsymbol{x}\|_{\mathcal{A}} \leq \lambda$, i.e., $\sum_{i \neq i_{\min}, i_{\min}+m} c_i^u + |\delta + c_{i_{\min}}^u - c_{i_{\min}+m}^u| \leq \lambda$ (by lemma 10). Since $\mathbb{1}^T\boldsymbol{c}^u = \|\boldsymbol{u}\|_{\mathcal{A}} \leq c$, we deduce the following constraint (note that we don't need to consider cases where $\delta + c_{i_{\min}}^u - c_{i_{\min}+m}^u \leq 0$ since in that case $\|x\|_{\mathcal{A}} \leq \lambda$ is trivially satisfied for any $\delta \geq 0$), $\delta \leq \lambda - \|\boldsymbol{u}\|_{\mathcal{A}} + 2c_{i_{\min}+m}^u$. Hence, δ_0 is indeed the maximal feasible step in this direction.

- $\delta_0 = t$ then $\boldsymbol{x}^0$ is optimal.
  Given any $\boldsymbol{x} \in \mathbb{R}^d$ s.t., $\|\boldsymbol{x}-\boldsymbol{u}\|_{\mathcal{A}} \leq t$, i.e., $\boldsymbol{x}-\boldsymbol{u} \in t\,\mathrm{conv}(\mathcal{A})$, we can write it as $\boldsymbol{x}-\boldsymbol{u} = \sum_{i=1}^{2m} c_i^{x-u}\boldsymbol{a}_i$ with $\boldsymbol{c}^{x-u} \geq 0$ and $\mathbb{1}^T\boldsymbol{c}^{x-u} = t$ (not necessarily a minimal

Algorithm 4 Prox of linearly independent atomic norms: $\mathrm{prox}_g^{\mathcal{A}}(\boldsymbol{u}, \boldsymbol{z}, L)$

Input: $\boldsymbol{c}^u = \mathrm{MR}(\boldsymbol{u})$.
Initialize: $\boldsymbol{x}^0 = \boldsymbol{u}, \boldsymbol{c}^x = \boldsymbol{c}^u, t_u^0 = 0, r = 1$.
$t_{\min} = \max\{\|\boldsymbol{u}\|_{\mathcal{A}} - \lambda, 0\}$
$\boldsymbol{a}_{i_{\min}} := \mathrm{LMO}_{\mathcal{A}}(\boldsymbol{z})$
$t_l^0 = \max\{-\boldsymbol{z}^T \boldsymbol{a}_{i_{\min}}/L, t_{\min}\}$.
Sort $\boldsymbol{z}^T \boldsymbol{a}_i$ for active atoms: $\boldsymbol{z}^T \boldsymbol{a}_{j_1} \geq \boldsymbol{z}^T \boldsymbol{a}_{j_2} \geq \cdots, \forall c_j^u > 0$.
if $t_{\min} > 0$ **then**
  Let r be the smallest integer s.t. $\sum_{k=1}^r c_{j_k}^u \geq t_{\min}$.
  for $k = 1, \cdots, r-1$ **do**
    $\boldsymbol{x}^0 = \boldsymbol{x}^0 - c_{j_k}^u \boldsymbol{a}_{j_k}, c_{j_k}^x = 0$.
  end for
  $\boldsymbol{x}^0 = \boldsymbol{x}^0 - (t_{\min} - \sum_{i=1}^k c_{j_k}^u) \boldsymbol{a}_{j_k}$
  $c_{j_k}^x = c_{j_k}^u - (t_{\min} - \sum_{i=1}^k c_{j_k}^u)$.
  $\delta_0 = t_{\min}$.
else
  $\delta_0 = \min\{t_l^0, \lambda - \|\boldsymbol{u}\|_{\mathcal{A}} + 2c_{i_{\min}+m}^u\}$
  $\boldsymbol{x}^0 = \boldsymbol{u} + \delta_0 \boldsymbol{a}_{i_{\min}}$.
  $c_{i_{\min}}^x = \max\{\delta_0 + c_{i_{\min}}^u - c_{i_{\min}+m}^u, 0\}, c_{i_{\min}+m}^x = -\min\{\delta_0 + c_{i_{\min}}^u - c_{i_{\min}+m}^u, 0\}$
end if
$t_u^0 = \delta_0, t_l^r = t_l^0 - t_u^0$.
while $k = r, \cdots, d$ and $t_l^k \geq 0$ **do**
  $t_l^k = \max\{-0.5\boldsymbol{z}^T(\boldsymbol{a}_{i_{\min}} - \boldsymbol{a}_{j_k})/L - t_u^k, 0\}$.
  $\delta_k = \min\{c_{j_k}^x, t_l^k/2\}$
  $\boldsymbol{x}^k = \boldsymbol{x}^{k-1} + \delta_k(\boldsymbol{a}_{i_{\min}} - \boldsymbol{a}_{j_k})$
  $t_l^{k+1} = t_l^k - 2\delta_k$
end while
Return: $\boldsymbol{x}^k$

Algorithm 5 Local Conjugate of linearly independent atomic norms: $\mathrm{lconj}_g^{\mathcal{A}}(\boldsymbol{u}, \boldsymbol{z}, t)$

Input: $\boldsymbol{c}^u = \mathrm{MR}(\boldsymbol{u}), t \geq 0$
Initialize: $\boldsymbol{x}^0 = \boldsymbol{u}, \boldsymbol{c}^x = \boldsymbol{c}^u, t_l^0 = t$
$\boldsymbol{a}_{i_{\min}} \in \arg\min_{\boldsymbol{a} \in \mathcal{A}} \boldsymbol{z}^T \boldsymbol{a}$
Sort: $\boldsymbol{z}^T \boldsymbol{a}_{j_1} \geq \boldsymbol{z}^T \boldsymbol{a}_{j_2} \geq \cdots, \forall c_j^u > 0$.
$\delta_0 = \min\{t_l^0, \lambda - \|\boldsymbol{u}\|_{\mathcal{A}} + 2c_{i_{\min}+m}^u\}$
$\boldsymbol{x}^0 = \boldsymbol{u} + \delta_0 \boldsymbol{a}_{i_{\min}}$.
$c_{i_{\min}}^x = \max\{\delta_0 + c_{i_{\min}}^u - c_{i_{\min}+m}^u, 0\}, c_{j_{\min}+m}^x = -\min\{\delta_0 + c_{i_{\min}}^u - c_{i_{\min}+m}^u, 0\}$
$t_l^1 = t_l^0 - \delta_0$.
while $k = 1, \cdots, m$ and $t_l^k \geq 0$ **do**
  $\delta_k = \min\{c_{j_k}^x, t_l^k/2\}$
  $\boldsymbol{x}^k = \boldsymbol{x}^{k-1} + \delta_k(\boldsymbol{a}_{i_{\min}} - \boldsymbol{a}_{j_k})$
  $t_l^{k+1} = t_l^k - 2\delta_k$
end while
Return: $\boldsymbol{x}^k$

representation). If $t = \delta_0$ then $\boldsymbol{z}^T(\boldsymbol{x} - \boldsymbol{u}) = \sum_{i=1}^{2m} c_i^{x-u} \boldsymbol{z}^T \boldsymbol{a}_i \geq t\boldsymbol{z}^T \boldsymbol{a}_{i_{\min}} = \boldsymbol{z}^T(\boldsymbol{x}^0 - \boldsymbol{u})$, so $\boldsymbol{x}^0$ is optimal.

- If $\delta_0 \neq t$, we have $\|x^0\|_{\mathcal{A}} = \lambda$.
  We prove this by contradiction. Assume $\|\boldsymbol{x}^0\|_{\mathcal{A}} < \lambda$ and let $\delta = \min\{\lambda - \|\boldsymbol{x}^0\|_{\mathcal{A}}, t - \delta_0\} > 0$, and let $\boldsymbol{x}' = \boldsymbol{u} + (\delta + \delta_0)\boldsymbol{a}_{i_{\min}} \neq \boldsymbol{x}^0$. $\boldsymbol{x}'$ is feasible since $\|\boldsymbol{x}' - \boldsymbol{u}\|_{\mathcal{A}} = \|(\delta + \delta_0)\boldsymbol{a}_{i_{\min}}\|_{\mathcal{A}} = \delta + \delta_0 \leq t$ and $\|\boldsymbol{x}'\|_{\mathcal{A}} \leq \|\boldsymbol{x}^0\|_{\mathcal{A}} + \|\delta \boldsymbol{a}_{i_{\min}}\|_{\mathcal{A}} \leq \lambda$ (by triangle inequality). This contradicts the above observation about δ^0.

- If $\delta_0 \neq t$, then there exists an optimal solution $\boldsymbol{x}^\star$ s.t. $\|\boldsymbol{x}^\star\|_{\mathcal{A}} = \lambda$.
  To see this let $\delta = \min\{(\lambda - \|\boldsymbol{x}^\star\|_{\mathcal{A}})/2, c_j^{x^\star - u}\} > 0$, where $\boldsymbol{c}^{x^\star - u} = \mathrm{MR}(\boldsymbol{x}^\star - \boldsymbol{u})$, and j any index that satisfies $j \neq i_{\min}, c_j^{x^\star - u} > 0$. Such index exists unless $\boldsymbol{x}^\star = \boldsymbol{u} + c_{i_{\min}}^{x^\star - u} \boldsymbol{a}_{i_{\min}}$, in which case $\boldsymbol{x}^0$ is optimal. Let $\boldsymbol{x}' = \boldsymbol{x}^\star + \delta(\boldsymbol{a}_{i_{\min}} - \boldsymbol{a}_j) \neq \boldsymbol{x}^\star$, $\|\boldsymbol{x}' - \boldsymbol{u}\|_{\mathcal{A}} = \|(c_{i_{\min}}^{x^\star - u} + \delta - c_{i_{\min}+m}^{x^\star - u})\boldsymbol{a}_{i_{\min}} + (c_j^{x^\star - u} - \delta))\boldsymbol{a}_j + \sum_{i \neq j, i_{\min}, i_{\min}+m} c_i^{x^\star - u} \boldsymbol{a}_i\|_{\mathcal{A}} \leq \mathbb{1}^T \boldsymbol{c}^{x^\star - u} \leq t$, $\|\boldsymbol{x}'\|_{\mathcal{A}} \leq \|\boldsymbol{x}^\star\|_{\mathcal{A}} + \|\delta \boldsymbol{a}_{i_{\min}}\|_{\mathcal{A}} + \|\delta(-\boldsymbol{a}_{i_j})\|_{\mathcal{A}} = \|\boldsymbol{x}^\star\|_{\mathcal{A}} + 2\delta \leq \lambda$. So $\boldsymbol{x}'$ is feasible and has a better objective than $\boldsymbol{x}^\star$ leading to a contradiction.

- There exists an optimal solution s.t. $\|\boldsymbol{x}^\star - \boldsymbol{x}^0\|_{\mathcal{A}} \leq t - \delta_0$.
  By the above observation, this is trivial if $t = \delta_0$. It also holds trivially if $\delta_0 = 0$. Otherwise, it is enough to show that $c_{i_{\min}}^{x^\star - u} \geq \delta_0$, where $\boldsymbol{c}^{x^\star - u} = \mathrm{MR}(\boldsymbol{x}^\star - \boldsymbol{u})$. Since $\|\boldsymbol{x}^\star - \boldsymbol{x}^0\|_{\mathcal{A}} = \|(c_{i_{\min}}^{x^\star - u} - c_{i_{\min}+m}^{x^\star - u} - \delta_0)\boldsymbol{a}_{i_{\min}} + \sum_{i \neq i_{\min}, i_{\min}+m} c_i^{x^\star - u} \boldsymbol{a}_i\|_{\mathcal{A}}$, if $c_{i_{\min}}^{x^\star - u} \geq \delta_0 > 0$, then by lemma 10, $c_{i_{\min}+m}^{x^\star - u} = 0$ and $\|\boldsymbol{x}^\star - \boldsymbol{x}^0\|_{\mathcal{A}} = \sum_{i \neq i_{\min}, i_{\min}+m} c_i^{x^\star - u} + (c_{i_{\min}}^{x^\star - u} - \delta_0) = \mathbb{1}^T \boldsymbol{c}^{x^\star - u} - \delta_0 \leq t - \delta_0$.
  To show that $c_{i_{\min}}^{x^\star - u} \geq \delta_0$, assume towards contradiction that $c_{i_{\min}}^{x^\star - u} < \delta_0$, and let j be an index where $c_j^{x^\star - u} > 0$ and $c_j^{x^\star - u} - c_{j+m}^u > 0$ and $j \neq i_{\min}$. Such index must exists, since otherwise $\forall i \neq i_{\min}$ where $c_i^{x^\star - u} > 0$, we'll have $0 < c_i^{x^\star - u} \leq c_{i+m}^u$, hence by lemma 10 $c_i^u = 0$. Then we can write $\boldsymbol{x}^\star = \boldsymbol{u} + \sum_i c_i^{x^\star - u} \boldsymbol{a}_i = \sum_{c_i^{x^\star - u} > 0, i \neq i_{\min}} (-c_{i+m}^u + c_i^{x^\star - u})\boldsymbol{a}_i + |c_{i_{\min}}^{x^\star - u} - c_{i_{\min}+m}^u| \boldsymbol{a}_{i_{\min}}$. We assume that $\exists i \neq i_{\min}, c_i^{x^\star - u} > 0$, otherwise $\boldsymbol{x}^\star = \boldsymbol{x}^0$. Hence, we'll have the following 2 cases:

$$
\begin{aligned}
\lambda &= \|\boldsymbol{x}^\star\|_{\mathcal{A}} \\
&= \begin{cases} \sum_{c_i^{x^\star - u} > 0} c_{i+m}^u - \sum_{c_i^{x^\star - u} > 0} c_i^{x^\star - u} & \text{if } c_{i_{\min}}^{x^\star - u} - c_{i_{\min}+m}^u \leq 0 \\ \sum_{c_i^{x^\star - u} > 0} c_{i+m}^u - \sum_{c_i^{x^\star - u} > 0, i \neq i_{\min}} c_i^{x^\star - u} + c_{i_{\min}}^{x^\star - u} - 2c_{i_{\min}+m}^u & \text{otherwise} \end{cases} \\
&< \begin{cases} \|\boldsymbol{u}\|_{\mathcal{A}} & \text{if } c_{i_{\min}}^{x^\star - u} - c_{i_{\min}+m}^u \leq 0 \\ \|\boldsymbol{u}\|_{\mathcal{A}} + \delta_0 - 2c_{i_{\min}+m}^u & \text{otherwise} \end{cases} \qquad (\text{since } c_{i_{\min}}^{x^\star - u} < \delta_0) \\
&\leq \lambda
\end{aligned}
$$

which leads to a contradiction. Hence, such index must exists. Then let $\delta = c_j^{x^\star - u} - c_{j+m}^u > 0$ and $\boldsymbol{x}' = \boldsymbol{x}^\star + \delta(\boldsymbol{a}_{i_{\min}} - \boldsymbol{a}_j) \neq \boldsymbol{x}^\star$. We show that $\boldsymbol{x}'$ is feasible. First note that $-\boldsymbol{u} = \sum_i c_i^u(-\boldsymbol{a}_i) = \sum_i c_{i+m}^u \boldsymbol{a}_i$ and hence by lemma 11, $\|\boldsymbol{x}^\star\|_{\mathcal{A}} = \|\boldsymbol{x}^\star - \boldsymbol{u} - (-\boldsymbol{u})\|_{\mathcal{A}} = \|\boldsymbol{c}^{x^\star - u} - \tilde{\boldsymbol{c}}^u\|_1 = \lambda$ where $\tilde{\boldsymbol{c}}_i^u = \boldsymbol{c}_{i+m}^u$. Then, we have $\|\boldsymbol{x}' - \boldsymbol{u}\|_{\mathcal{A}} = \|(c_{i_{\min}}^{x^\star - u} +$

$\delta - c^{x^\star - u}_{i_{\min}+m})\boldsymbol{a}_{i_{\min}} + (c^{x^\star-u}_j - \delta))\boldsymbol{a}_j + \sum_{i\neq j, i_{\min}, i_{\min}+m} c^{x^\star-u}_i \boldsymbol{a}_i\|_{\mathcal{A}} \leq \mathbb{1}^T \boldsymbol{c}^{x^\star-u} \leq t$ and $\|\boldsymbol{x}'\|_{\mathcal{A}} = \|\sum_{i\neq j, j+m}(c^{x^\star-u}_i + c^u_i)\boldsymbol{a}_i + (c^{x^\star-u}_j + c^u_j - c^u_{j+m} - \delta)\boldsymbol{a}_j + \delta \boldsymbol{a}_{i_{\min}}\|_{\mathcal{A}} \leq \|\sum_{i\neq j, j+m} c^{x^\star-u}_i \boldsymbol{a}_i + (c^{x^\star-u}_j - c^u_{j+m} - \delta)\boldsymbol{a}_j - (-\boldsymbol{u})\|_{\mathcal{A}} + \|\delta \boldsymbol{a}_{i_{\min}}\|_{\mathcal{A}} = \|\boldsymbol{c}^{x^\star-u} - \tilde{\boldsymbol{c}}^u\|_1 - \delta + \delta = \lambda$, by lemma 13. Finally note that $\boldsymbol{z}^T\boldsymbol{x}' \leq \boldsymbol{z}^T\boldsymbol{x}^\star$ leading to a contradiction.

Note that in Algorithm 5 we enter the loop only if $t \neq \delta_0$. So if we stop before that then we have found an optimal solution $\boldsymbol{x}^0$. Otherwise, there exists an optimal solution s.t. $\|\boldsymbol{x}^\star\|_{\mathcal{A}} = \lambda$ and $\|\boldsymbol{x}^\star - \boldsymbol{x}^0\|_{\mathcal{A}} \leq t - \delta_0$, so we can now solve this problem instead:

$$\min_{\substack{\|\boldsymbol{x}\|_{\mathcal{A}}=\lambda \\ \|\boldsymbol{x}-\boldsymbol{x}^0\|_{\mathcal{A}}\leq t}} \boldsymbol{z}^T\boldsymbol{x} \tag{4.23}$$

We know though by lemma 11 that $\|\boldsymbol{x}-\boldsymbol{x}^0\|_{\mathcal{A}} = ||\boldsymbol{c}^x - \boldsymbol{c}^{x^0}\|_1$, for $\boldsymbol{c}^x = \mathrm{MR}(\boldsymbol{x}), \boldsymbol{c}^{x^0} = \mathrm{MR}(\boldsymbol{x}^0)$. Hence we can further reformulate problem 4.23 as:

$$\min_{\substack{\mathbb{1}^T\boldsymbol{c}^x=\lambda, \boldsymbol{c}^x\geq 0 \\ \|\boldsymbol{c}^x-\boldsymbol{c}^{x^0}\|_1\leq t}} \tilde{\boldsymbol{z}}^T\boldsymbol{c}^x \tag{4.24}$$

where $\tilde{\boldsymbol{z}}_i = \boldsymbol{z}^T\boldsymbol{a}_i$. This problem has been considered by [GH16], to obtain a local linear oracle. The rest of our algorithm, i.e., after entering the loop, reduces to their algorithm. So we refer the reader to their proof of correctness [GH16, Lemma 5.2]. This concludes the proof that algorithm 5 returns $x^k \in \mathrm{lconj}^{\mathcal{A}}_g(\boldsymbol{u}, \boldsymbol{z}, t)$.

Now we argue that Algorithm 2 returns $x^k \in \mathrm{prox}^{\mathcal{A}}_g(\boldsymbol{u}, \boldsymbol{z}, L)$. Recall from section 4.3.1 that $h(t)$ is a non-increasing piecewise linear function. But unlike the general case where we're computing $h(t)$ using a black box optimizer, we actually can compute the slopes of the different pieces of $h(t)$ explicitly. In fact, $h'(t^\star)$ belongs to one of these intervals: $[\boldsymbol{z}^T\boldsymbol{a}_{i_{\min}}, \infty], [0.5(\boldsymbol{z}^T\boldsymbol{a}_{i_{\min}} - \boldsymbol{z}^T\boldsymbol{a}_{j_2}), 0.5(\boldsymbol{z}^T\boldsymbol{a}_{i_{\min}} - \boldsymbol{z}^T\boldsymbol{a}_{j_1})], \cdots$. Note that Algorithm 5 is actually minimizing the objective along the path of possible values of $t' = \|\boldsymbol{x} - \boldsymbol{u}\|_{\mathcal{A}}$ from $t' = t_{\min}$ to $t' = t$. In fact, $\boldsymbol{x}^k \in \mathrm{lconj_g}(\boldsymbol{u}, \boldsymbol{z}, t^k_u), \forall k$ in Algorithm 5. Hence, it's easy to incorporate the search for $t^\star$ without increasing the time complexity.
Finally, it is clear that the most expensive step in Algorithm 2 is the sorting operation on line 5, and hence it's time complexity is $O(d\mathcal{T} + d\log d)$. Handling the case where $\|\boldsymbol{u}\|_{\mathcal{A}} > \lambda$ (c.f., lines 7 -14) follows using similar arguments. □

4.7.4 Proof of Theorem 5

First, we present a technical lemma, which can be found in [Cio90, Example 2.9]. We provide a proof of it here for completeness. The term $\mathbf{p}^k$ satisfies the following property, which is useful to handle general norms.

Lemma 12 (*cf.*, [Cio90, Example 2.9])**.** $\|\cdot\|^2$ *is differentiable at zero with* $\partial(\frac{1}{2}\|\cdot\|^2)(\mathbf{0}) = 0$

and $\forall \boldsymbol{x} \in \mathbb{R}^d$ *and* $\mathbf{p} \in \partial(\frac{1}{2}\|\cdot\|^2)(\boldsymbol{x})$*, we have*

$$\langle \boldsymbol{x}, \mathbf{p} \rangle = \|\boldsymbol{x}\|^2 = \|\mathbf{p}\|_*^2. \tag{4.25}$$

Proof. Note that $\forall \boldsymbol{x} \in \mathbb{R}^d$

$$\lim_{t \to 0} \frac{\|\mathbf{0} + t\boldsymbol{z}\|^2 - \|\mathbf{0}\|^2}{t} = \lim_{t \to 0} t\|\boldsymbol{x}\|^2 = 0, \tag{4.26}$$

which implies that $\|\cdot\|^2$ is differentiable at $\mathbf{0}$. Hence if $\boldsymbol{x} = 0$ then $\mathbf{p} = 0$ and (4.25) trivially holds. Otherwise if $\boldsymbol{x} \neq 0$, note that since $\|\cdot\|^2$ is positively homogeneous of degree 2 and is locally Lipschitz, then by [YWW10] Euler's identity holds

$$\|\boldsymbol{x}\|^2 = \langle \boldsymbol{x}, \mathbf{p} \rangle \leq \|\boldsymbol{x}\|\|\mathbf{p}\|_*, \tag{4.27}$$

which implies that $\|\boldsymbol{x}\| \leq \|\mathbf{p}\|_*$. The subdifferential of $\|\cdot\|$ exists at every point (see [Zal02]) and $\mathbf{p}/\|\boldsymbol{x}\| \in \partial\|\boldsymbol{x}\|$. Then since $\|\cdot\| \in \Gamma_0$, it follows by Fenchel-Young equality,

$$\|\mathbf{p}/\|\boldsymbol{x}\|\|^* + \|\boldsymbol{x}\| = \langle \boldsymbol{x}, \mathbf{p}/\|\boldsymbol{x}\| \rangle = \|\boldsymbol{x}\|, \tag{4.28}$$

where $\|\cdot\|^*$ is the Fenchel conjugate of $\|\cdot\|$. This implies that $\|\mathbf{p}/\|\boldsymbol{x}\|\|^* = \iota_{\|\cdot\|_* \leq 1}(\mathbf{p}/\|\boldsymbol{x}\|) = 0$ and hence $\|\mathbf{p}\|_* \leq \|\boldsymbol{x}\|$, and thus (4.25) holds.

□

Theorem 5 (Convergence[4])**.** *Given g which is μ-strongly convex w.r.t. $\|\cdot\|$. If accGPM terminates at iteration k, i.e., $\boldsymbol{x}^{k+1} = \boldsymbol{y}^{k+1}$, then $\boldsymbol{x}^{k+1}$ is a solution to* (4.1)*. Otherwise, let $x^\star \in \mathscr{X}^\star$, the iterates of accGPM satisfy the following.*

1. *If $\mu = 0$. Then $\forall k \in \mathbb{N}$, we have* $F(\boldsymbol{x}^{k+1}) - F^\star \leq \dfrac{4\big(\sigma(F(\boldsymbol{x}^0) - F^\star) + \beta_0 D(\boldsymbol{x}^\star)\big)}{\sigma\{2 + \sqrt{\beta_0}\sum_{i=0}^k \sqrt{\gamma_i}\}^2}$.

   Consequently, if $\forall k \in \mathbb{N}, \gamma_k = 1/L$, then $F(\boldsymbol{x}^{k+1}) - F^\star \leq \dfrac{4L\big(\sigma(F(\boldsymbol{x}^0) - F^\star) + \beta_0 D(\boldsymbol{x}^\star)\big)}{\sigma\{2\sqrt{L} + \sqrt{\beta_0}(k+1)\}^2}$.

2. *If $\mu > 0$. Set $\tau = \inf_{k \in \mathbb{N}} \tau_k$, and* $\rho = \frac{\tau\mu}{2L}\left\{\sqrt{1 + \frac{4L}{\beta_0 + \mu}} - 1\right\}$*. If $\beta_0 \geq \tau\mu$ and $\forall k \in \mathbb{N}, \gamma_k = 1/L$, then we have* $F(\boldsymbol{x}^{k+1}) - F^\star \leq (1-\rho)^{k+1}\{F(\boldsymbol{x}^0) - F^\star + \frac{\beta_0}{\sigma} D(\boldsymbol{x}^\star)\}$.

Proof. If there exists $k \in \mathbb{N}$ such that $x^{k+1} = y^{k+1}$ then it follows from Step 6 of Algorithm 3 and Fermat's rule that

$$0 \in \partial g(\boldsymbol{x}^{k+1}) + \nabla f(\boldsymbol{x}^{k+1}) + \partial(\frac{1}{2\gamma_k}\|\cdot\|^2)(\mathbf{0}) \tag{4.29}$$

[4]This theorem and its proof are due primarily to B. Vu.

By lemma 12, (4.29) yields $0 \in \partial g(\boldsymbol{x}^{k+1}) + \nabla f(\boldsymbol{x}^{k+1})$ and thus $\boldsymbol{x}^k$ is a minimizer of F. We now suppose that $\forall k \in \mathbb{N}, x^{k+1} \neq y^{k+1}$. Step 10 of Algorithm 3 yields

$$(\forall k \in \mathbb{N}) \quad \mathbf{p}^k - \nabla f(\boldsymbol{y}^{k+1}) \in \partial g(\boldsymbol{x}^{k+1}) \tag{4.30}$$

It follows then that

$$g(\boldsymbol{x}) \geq g(\boldsymbol{x}^{k+1}) + \Big\langle \boldsymbol{x} - \boldsymbol{x}^{k+1}, \mathbf{p}^k - \nabla f(\boldsymbol{y}^{k+1}) \Big\rangle. \tag{4.31}$$

Since ∇f is L-Lipschitz and since $\forall k \in \mathbb{N}, \gamma_k \in (0, 1/L]$, it follows from that

$$f(\boldsymbol{x}^{k+1}) \leq f(\boldsymbol{y}^{k+1}) + \big\langle \boldsymbol{x}^{k+1} - \boldsymbol{y}^{k+1}, \nabla f(\boldsymbol{y}^{k+1}) \big\rangle + \frac{1}{2\gamma_k} \|\boldsymbol{x}^{k+1} - \boldsymbol{y}^{k+1}\|^2. \tag{4.32}$$

In turn the convexity of f implies that

$$\begin{aligned} f(\boldsymbol{x}) &\geq f(\boldsymbol{y}^{k+1}) + \big\langle \boldsymbol{x} - \boldsymbol{y}^{k+1}, \nabla f(\boldsymbol{y}^{k+1}) \big\rangle \\ &\geq f(\boldsymbol{x}^{k+1}) + \big\langle \boldsymbol{y}^{k+1} - \boldsymbol{x}^{k+1}, \nabla f(\boldsymbol{y}^{k+1}) \big\rangle - \frac{1}{2\gamma_k} \|\boldsymbol{x}^{k+1} - \boldsymbol{y}^{k+1}\|^2 + \big\langle \boldsymbol{x} - \boldsymbol{y}^{k+1}, \nabla f(y^{k+1}) \big\rangle \\ &= f(\boldsymbol{x}^{k+1}) + \big\langle \boldsymbol{x} - \boldsymbol{x}^{k+1}, \nabla f(\boldsymbol{y}^{k+1}) \big\rangle - \frac{1}{2\gamma_k} \|\boldsymbol{x}^{k+1} - \boldsymbol{y}^{k+1}\|^2. \end{aligned} \tag{4.33}$$

Adding (4.31) and (4.33) we get

$$F(\boldsymbol{x}) \geq F(\boldsymbol{x}^{k+1}) + \Big\langle \boldsymbol{x} - \boldsymbol{x}^{k+1}, \mathbf{p}^k \Big\rangle - \frac{1}{2\gamma_k} \|\boldsymbol{x}^{k+1} - \boldsymbol{y}^{k+1}\|^2. \tag{4.34}$$

Hence, for every $\boldsymbol{x} \in \mathbb{R}^d$,

$$\begin{aligned} e_{k+1}(\boldsymbol{x}) - F(\boldsymbol{x}) &= (1 - \alpha_k)(e_k(\boldsymbol{x}) - F(\boldsymbol{x})) + \alpha_k \big((1 - \tau_k)(\psi_k(\boldsymbol{x}) - F(\boldsymbol{x})) + \tau_k(\phi_k(\boldsymbol{x}) - F(\boldsymbol{x}))\big) \\ &\leq (1 - \alpha_k)(e_k(\boldsymbol{x}) - F(\boldsymbol{x})). \end{aligned} \tag{4.35}$$

Since d is σ-strongly convex and g is μ-strongly convex, it follows by induction that e_k is β_k-strongly convex. Next, let us show that

$$(\forall k \in \mathbb{N}) \quad e_k(\boldsymbol{w}^k) \geq F(\boldsymbol{x}^k). \tag{4.36}$$

Note that $e_0(\boldsymbol{w}^0) \geq F(\boldsymbol{x}^0)$. Suppose that $e_k(\boldsymbol{w}^k) \geq F(\boldsymbol{x}^k)$ for some $k \in \mathbb{N}$. Then it follows from (4.34) that

$$e_k(\boldsymbol{w}^k) \geq F(\boldsymbol{x}^k) \geq \psi_k(\boldsymbol{x}^k)$$

Hence, since e_k is β_k-strongly convex, we have

$$\begin{aligned} e_k(\boldsymbol{w}^{k+1}) &\geq e_k(\boldsymbol{w}^k) + \frac{\beta_k}{2} \|\boldsymbol{w}^{k+1} - \boldsymbol{w}^k\|^2 \\ &\geq \psi_k(\boldsymbol{x}^k) + \frac{\beta_k}{2} \|\boldsymbol{w}^{k+1} - \boldsymbol{w}^k\|^2. \end{aligned} \tag{4.37}$$

However, since $\mathbf{p}^k - \nabla f(\boldsymbol{y}^{k+1}) \in \partial g(\boldsymbol{x}^{k+1})$,

$$g(\boldsymbol{x}) + \left\langle \boldsymbol{x} - \boldsymbol{x}^{k+1}, \nabla f(\boldsymbol{y}^{k+1}) - \mathbf{p}^k \right\rangle \geq g(\boldsymbol{x}^{k+1}), \tag{4.38}$$

and hence we deduce from (4.32) that

$$\phi_k(\boldsymbol{x}) \geq \psi_k(\boldsymbol{x}). \tag{4.39}$$

In turn, we deduce from (4.37) that

$$\begin{aligned} e_{k+1}(\boldsymbol{w}^{k+1}) &\geq (1-\alpha_k)e_k(\boldsymbol{w}^{k+1}) + \alpha_k\psi_k(\boldsymbol{w}^{k+1}) \\ &= F(\boldsymbol{x}^{k+1}) + \frac{(1-\alpha_k)\beta_k}{2}\|\boldsymbol{w}^{k+1} - \boldsymbol{w}^k\|^2 + \left\langle \boldsymbol{y}^{k+1} - \boldsymbol{x}^{k+1}, \mathbf{p}^k \right\rangle \\ &+ \left\langle \alpha_k(\boldsymbol{w}^{k+1} - \boldsymbol{w}^k), \mathbf{p}^k \right\rangle - \frac{1}{2\gamma_k}\|\boldsymbol{x}^{k+1} - \boldsymbol{y}^{k+1}\|^2. \end{aligned} \tag{4.40}$$

It follows from definition of $\mathbf{p}^k$ and Lemma 12 that

$$\left\langle \boldsymbol{y}^{k+1} - \boldsymbol{x}^{k+1}, \mathbf{p}^k \right\rangle = \frac{1}{\gamma_k}\|\boldsymbol{x}^{k+1} - \boldsymbol{y}^{k+1}\|^2 = \gamma_k\| - \mathbf{p}^k\|_*^2. \tag{4.41}$$

On the other hand, the Cauchy-Schwarz inequality yields

$$\alpha_k\langle \boldsymbol{w}^{k+1} - \boldsymbol{w}^k, \mathbf{p}^k\rangle \geq -\frac{(1-\alpha_k)\beta_k}{2}\|\boldsymbol{w}^{k+1} - \boldsymbol{w}^k\|^2 - \frac{\alpha_k^2}{2(1-\alpha_k)\beta_k}\|\mathbf{p}^k\|_*^2. \tag{4.42}$$

Consequently, we deduce from (4.40) and (4.41) that

$$\begin{aligned} e_{k+1}(\boldsymbol{w}^{k+1}) &\geq F(\boldsymbol{x}^{k+1}) + \frac{1}{2\gamma_k}\|\boldsymbol{x}^{k+1} - \boldsymbol{y}^{k+1}\|^2 - \frac{\alpha_k^2}{2(1-\alpha_k)\beta_k\gamma_k}\|\boldsymbol{x}^{k+1} - \boldsymbol{y}^{k+1}\|^2 \\ &\geq F(\boldsymbol{x}^{k+1}) + \frac{1}{2\gamma_k}\left\{1 - \frac{\alpha_k^2}{(1-\alpha_k)\beta_k\gamma_k}\right\}\|\boldsymbol{x}^{k+1} - \boldsymbol{y}^{k+1}\|^2 \\ &= F(\boldsymbol{x}^{k+1}) \end{aligned} \tag{4.43}$$

which proves (4.36). Finally, we derive from the definition of $\boldsymbol{w}_{k+1}$ and (4.35) that

$$(\forall k \in \mathbb{N}) \quad F(\boldsymbol{x}^{k+1}) - F^\star \leq e_{k+1}(\boldsymbol{w}^{k+1}) - F^\star \leq e_{k+1}(\boldsymbol{x}^\star) - F^\star \leq (1-\alpha_k)\big(e_k(\boldsymbol{x}^\star) - F^\star\big), \tag{4.44}$$

where $x^\star$ is a minimizer of F. Hence, by induction,

$$F(\boldsymbol{x}^{k+1}) - F^\star \leq \prod_{i=0}^{k}(1-\alpha_i)\big(e_0(\boldsymbol{x}^\star) - F^\star\big). \tag{4.45}$$

(1): Note that $\forall k \in \mathbb{N}$

$$\alpha_k^2 = (1-\alpha_k)\beta_k\gamma_k \quad \text{and} \quad \beta_{k+1} = (1-\alpha_k)\beta_k$$

Hence, it follows from Lemma 2.2 in [Gül92] that

$$\prod_{i=0}^{k}(1-\alpha_i) \leq \frac{1}{\left(1+\sqrt{\beta_0}/2\sum_{i=0}^{k}\sqrt{\gamma_i}\right)^2}. \tag{4.46}$$

Consequently, the assertion follows from (4.45).

(2): First we note that by induction,

$$(\forall k \in \mathbb{N}) \quad \tau\mu \leq \beta_k \leq \beta_0 + \mu. \tag{4.47}$$

Therefore,

$$\alpha_{k+1} = \frac{\beta_k}{2L}\left\{\sqrt{1+\frac{4L}{\beta_k}}-1\right\} \geq \frac{\tau\mu}{2L}\left\{\sqrt{1+\frac{4L}{\beta_0+\mu}}-1\right\} \tag{4.48}$$

and hence the assertion follows from (4.44). □

4.7.5 Solving $\mathrm{prox}_{\ell_1}^{\ell_1}$ and $\mathbf{p}^k$ for Section 4.5.1

Computing ℓ_1-proximal operator

Computing the standard $\mathrm{prox}_{\ell_1}^{\ell_2}$ of ℓ_1-norm can be computed efficiently in $O(d)$ using the so-called soft thresholding operator [DJ95], $S(\boldsymbol{z},\lambda) = \mathrm{sign}(\boldsymbol{z}) \circ \max\{|\boldsymbol{z}| - \lambda, 0\}$.

As explained in Remark 4, $\mathrm{prox}_{\ell_1}^{\ell_1}$ can be solved by computing the prox over the ℓ_∞ ball by the greedy Algorithm 4 and applying the decomposition in 14. By proposition 15, $\mathrm{prox}_{\ell_1}^{\ell_1}$ can then be computed in $O(d\log d)$ time. However, $\mathrm{prox}_{\ell_1}^{\ell_1}$ is simple enough that we opt for a direct way to solve it using again an intuitive greedy algorithm, of the same “flavor" as Algorithm 2.

Proposition 16. *Algorithm 6 returns $\boldsymbol{x} \in \mathrm{prox}_{\ell_1}^{\ell_1}(\boldsymbol{u},\boldsymbol{z},L)$ in $O(d\log d)$ time.*

The high level idea of Algorithm 10 is the following. By lemma 11, we know that computing $\mathrm{prox}_{\ell_1}^{\ell_1}(\boldsymbol{u},\boldsymbol{z},L)$ is equivalent to computing $\mathrm{lconj}_{\ell_1}^{\ell_1}(\boldsymbol{u},\boldsymbol{z},L,t^\star)$, where recall from Section 4.3.2, $h(t)$ is a non-increasing piecewise linear function, whose slopes can be computed explicitly. Hence, the search for $t^\star$ is done in Algorithm 10 the same way as in Algorithm 2. Algorithm 10 then solves $\mathrm{lconj}_{\ell_1}^{\ell_1}(\boldsymbol{u},\boldsymbol{z},L,t)$ along the path of possible t values. Note that given t and the signs of $\boldsymbol{x}$, the objective in $\mathrm{lconj}_{\ell_1}^{\ell_1}(\boldsymbol{u},\boldsymbol{z},L,t)$ reduces to minimizing a linear function $\mathrm{sign}(\boldsymbol{x}) \circ (\boldsymbol{z} + \lambda\,\mathrm{sign}(\boldsymbol{x}))$ over the ℓ_1-ball $\|\boldsymbol{x}-\boldsymbol{u}\|_1 \leq t^\star$ and the signs constraint, whose solution is simple (c.f., lines 10 and 16). To guess the optimal signs, we start by the feasible ones, $\mathrm{sign}(\boldsymbol{u})$, and modify them gradually along the greedy solution path. It's clear that the time complexity of

Algorithm 6 ℓ_1-prox of ℓ_1-norm

1: **Input:** $\boldsymbol{u} \in \mathbb{R}^d, L_1 > 0$
2: **Initialization:** $\boldsymbol{x}^0 = \boldsymbol{u}, t_u^0 = 0, \boldsymbol{s}^0 = \operatorname{sign}(\boldsymbol{x}^0), k = 0$
3: $s_i^0 = -\operatorname{sign}(S(z_i, \lambda)), \forall i$ s.t. $x_i^0 = 0$.
4: $\boldsymbol{w} = [\boldsymbol{s}^0 \circ (\boldsymbol{z} + \lambda \boldsymbol{s}^0), \boldsymbol{s}^0 \circ (\boldsymbol{z} - \lambda \boldsymbol{s}^0)]$
5: Sort: $|w_{i_1}| \geq |w_{i_2}| \geq \cdots |w_{i_{2p}}|$
6: **while** $k = 1, \cdots, d+1$ and $t_l^k \geq 0$ **do**
7: **if** $w_{i_k} = s_{i_k}^k \circ (z_{i_k} + \lambda s_{i_k}^k)$ **then**
8: $t_l^{k+1} \leftarrow \max\{|w_{i_k}^k|/L_1 - t_u^k, 0\}$
9: **if** $\operatorname{sign}(w_{i_k}^k) > 0$ **then**
10: $x_{i_k}^{k+1} = s_{i_k}^k \max\{|x_{i_k}^k| - t_l^k, 0\}$
11: $t_u^{k+1} = t_u^k - |x_{i_k}^{k+1}| + |x_{i_k}^k|$
12: **if** $x_{i_k}^{k+1} = 0$ **then**
13: $s_{i_k}^{k+1} = -\operatorname{sign}(S(z_i, \lambda))$
14: **end if**
15: **else**
16: $x_{i_k}^{k+1} = s_{i_k}^k (|x_{i_k}^k| + t_l^k)$
17: $t_u^{k+1} = t_u^k + |x_{i_k}^{k+1}| - |x_{i_k}^k|$
18: **end if**
19: **end if**
20: **end while**
Return: $\boldsymbol{x}^{k+1}$

Algorithm 10 is dominated by the sorting operation on line 5, leading to a worst case complexity of $O(d \log d)$. However, in practice, we notice that we rarely do more than one iteration. In fact, when Algorithm 10 is executed within GPM and accGPM, it's not hard to see that doing more than one iteration requires $\|\nabla f(\boldsymbol{x}^k)\|_\infty \leq \lambda$, and since λ is usually small, this condition implies that we're already near convergence, which is exactly what we observe in our experiments (c.f., section 4.5.1). Hence, in our implementation we choose instead to compute the maximum value of $\boldsymbol{w}$ at each iteration instead of sorting, leading to an expected complexity of $O(d)$. This observation is interesting, since it implies that running FW with a carefully chosen step-size, approximate running a proximal gradient method.

Finally note that the updates generated by $\boldsymbol{x} \in \operatorname{prox}_{\ell_1}^{\ell_1}(\boldsymbol{u}, \boldsymbol{z}, L)$ are always sparse, i.e., given an s-sparse vector $\boldsymbol{u}$, $\boldsymbol{x}$ is at most $s+1$-sparse. The proof of proposition 16 follows by similar arguments as in proposition 15.

Computing the momentum term $\mathbf{p}^k$

Recall that accGPM required the computation of a momentum term $\mathbf{p}^k$ at each iteration k (c.f., line 10). Below, we show that the computation of $\mathbf{p}^k$, in this setting, has a closed form solution.

Proposition 17. *Given $\boldsymbol{x} \in \operatorname{prox}_{\ell_1}^{\ell_1}(\boldsymbol{u}, \boldsymbol{z}, L)$ generated by Algorithm 6, to have $\mathbf{p} \in \partial\left(-\frac{L}{2}\|\boldsymbol{x} - \boldsymbol{u}\|_1^2\right) \cap$*

$(\boldsymbol{z} + \lambda\partial\|\boldsymbol{x}\|_1)$, *we can choose*

$$p_i = \begin{cases} -L\|\boldsymbol{x}-\boldsymbol{u}\|_1 \operatorname{sign}(\boldsymbol{x}-\boldsymbol{u}) & \text{if } 0 = x_i \neq u_i \\ (s_i)^2(z_i + \lambda s_i) & \text{otherwise} \end{cases}$$

where $s_i = \operatorname{sign}(x_i)$ *if* $x_i \neq 0$ *and* $s_i = -\operatorname{sign}(S(z_i, \lambda))$ *otherwise.*

Proof. We note first than, by the optimality of $\boldsymbol{x}$, whenever one of the two sets consists of a unique element, then choosing this element must be a feasible choice. If $\|\boldsymbol{x}-\boldsymbol{u}\|_1 = 0$, then from Algorithm 6, we know that $\boldsymbol{s} = 0$ and hence the above choice will correspond to $\mathbf{p} = 0$, which is the unique choice here. If $0 = x_i \neq u_i$ or $x_i \neq 0$ the choice of p_i above is again unique. Otherwise, if $0 = x_i = u_i$, then the choice of p_i is a feasible one, since $(s_i)^2(z_i + \lambda s_i) \in [-L\|\boldsymbol{x}-\boldsymbol{u}\|_1, L\|\boldsymbol{x}-\boldsymbol{u}\|_1] \cap z_i + [-\lambda, \lambda]$. □

4.7.6 Solving $\mathbf{p}^k$ for $\operatorname{prox}_G^{\ell_\infty}$ in Section 4.5.2

Proposition 18. *Given* $\boldsymbol{x} \in \operatorname{prox}_G^{\ell_\infty}(\boldsymbol{u}, \boldsymbol{z}, L)$ *generated by Algorithm 1, to have* $\mathbf{p} \in \partial\left(-\frac{L}{2}\|\boldsymbol{x}-\boldsymbol{u}\|_\infty^2\right) \cap (\boldsymbol{z} + \lambda\partial\|\boldsymbol{x}\|_G)$, *where* $\|\boldsymbol{x}\|_G$ *is the* ℓ_∞*-LGL norm, we need to solve the following linear feasibility program.*

$$\begin{aligned} \mathbf{p} \in \operatorname*{arg\,min}_{\mathbf{p}\in\mathbb{R}^d} \quad & 0 \\ \textit{subject to} \quad & \mathbf{p}^T\left(\frac{\boldsymbol{x}-\boldsymbol{u}}{-Lt}\right) = t \\ & \left(\operatorname{sign}(\boldsymbol{x}-\boldsymbol{u}) \circ \frac{\mathbf{p}}{-Lt}\right)^T \mathbb{1} \leq 1 \\ & \boldsymbol{x}^T(\mathbf{p}-\boldsymbol{z}) = \lambda\|\boldsymbol{x}\|_G \\ & \mathbf{B}^T(\operatorname{sign}(\boldsymbol{x}) \circ (\mathbf{p}-\boldsymbol{z})) \leq \lambda \\ & \operatorname{sign}(\boldsymbol{x}-\boldsymbol{u}) = \operatorname{sign}(\mathbf{p}) \\ & \operatorname{sign}(\boldsymbol{x}) = \operatorname{sign}(\mathbf{p}-\boldsymbol{z}) \end{aligned}$$

where $t = \|\boldsymbol{x}-\boldsymbol{u}\|_\infty$ *and* $\mathbf{B}$ *is the matrix whose columns are the indicator vectors of the groups, i.e.,* $\mathbf{B}_i = \mathbb{1}_{G_i}$.

Proof. By definition of dual norms, the subdifferential of the $\partial\left(-\frac{L}{2}\|\boldsymbol{x}-\boldsymbol{u}\|_\infty^2\right) = \{-Lt\kappa : \kappa^T(\boldsymbol{x}-\boldsymbol{u}) = \|\boldsymbol{x}-\boldsymbol{u}\|_\infty, \|\kappa\|_1 \leq 1\}$. The dual of ℓ_∞-LGL norm is given by $\max_{i\in[1,\cdots,M]} \|\kappa_{G_i}\|_1$, hence $\lambda\partial\|\boldsymbol{x}\|_G = \{\kappa : \kappa^T\boldsymbol{x} = \lambda\|\boldsymbol{x}\|_G, \|\kappa_{G_i}\|_1 \leq \lambda, \forall i \in [1, \cdots, M]\}$. The proposition then follows directly. □

5 Non-Convex Proximal Method for Structured Sparsity

5.1 Introduction

Our discussion so far has focused on convex approaches to structured sparse learning. We saw that for structures that can be expressed by monotone submodular functions or monotone integral linear programs (i.e, ILP penalties introduced in Chapter 2), employing the convex envelope as a convex surrogate can yield efficient and accurate solutions. On the other hand, our analysis in Chapter 3 showed that convex relaxations fail to capture *non-monotone* structures. Furthermore, computing the convex envelope of structures outside the class of submodular and ILP penalties is in general intractable. To handle such cases, we adopt in this chapter a non-convex approach. We are thus interested in directly addressing the following optimization problem:

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} f(\boldsymbol{x}) + F(\mathrm{supp}(\boldsymbol{x})), \tag{5.1}$$

where $f : \mathbb{R}^d \to \mathbb{R}$ is a smooth convex loss function and $F : 2^V \to \overline{R}_+$, where $V = \{1, \cdots, d\}$, is a normalized (i.e., $F(\emptyset) = 0$) set function encoding a structured sparsity model.

5.1.1 Related work

A main instance of functions that do not admit meaningful convex envelopes are *symmetric*[1] functions. In the special case of symmetric submodular functions, [Bac11] proposed to use their Lovász extension as a convex regularizer, to impose prior knowledge on the level sets of $\boldsymbol{x}$, instead of its support. This choice is justified by showing that the Lovász extension can be viewed as the convex envelope of F defined over the level sets[2] of $\boldsymbol{x}$.

On the other hand, existing non-convex approaches (see Section 1.5) have mostly focused on the constrained formulation of the structured learning problem (5.1). In this setting, available

[1] A set function F is symmetric if it satisfies $F(S) = F(V \setminus S), \forall S \subseteq V$

[2] Level sets of a vector $\boldsymbol{x} \in \mathbb{R}^d$ are sets of indices above a given threshold α; i.e., $\{i : x_i \geq \alpha\}$.

algorithms require a *discrete projection* step, which is NP-Hard for general structures, including submodular and ILP structures. Existing non-convex approaches that consider the penalized formulation are limited to simple sparsity [HGT06] and non-overlapping group sparsity [BH18].

5.1.2 Contributions

This chapter presents preliminary results towards addressing problem (5.1), when disciplined convex approaches are inapplicable. In particular, we first demonstrate that the Lovász extension is not a suitable convex relaxation of symmetric submodular functions, in cases where we do not seek piecewise constant solutions. We propose to use instead a *discrete proximal gradient descent* method, which is a simple extension of the discrete projected gradient descent method, typically used in non-convex approaches (see Section 1.5), and is efficient for several classes of structures, including submodular, supermodular and ILP penalties. For concreteness, we consider regression with clustering penalties as a motivating example. We numerically illustrate that the proposed algorithm performs better than the alternative convex method, for this example.

This chapter is based on the joint work with Luca Baldassarre and Volkan Cevher [EHBC13].

5.2 Motivating example: Graph cuts

As a motivating example, we consider a compressive sensing scenario where we observe compressive samples $\boldsymbol{y} \in \mathbb{R}^n$ of a "clustered" sparse vector $\boldsymbol{x}^\natural \in \mathbb{R}^d$, through a dimensionality reducing matrix $\boldsymbol{A}$. In this case, $f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\|_2^2$. Then, the clustering model can be naturally enforced by a cut function $F_{\text{cut}}(S) = |\delta(S)|$, defined over an appropriate undirected graph $\mathcal{G} = (V, E)$, where $\delta(S) = \{(i,j) \in E : i \in S, j \notin S\}$ is the cut-set induced by S in $\mathcal{G}$. One can also consider a weighted version of F_{cut}, where each edge is associated with a positive weight. Cut functions favor sets whose elements are clustered on the given graph, which is a desired structure for example in several computer vision problems [CHDB09, KZ04]. For example, for background subtraction in images, the vertices of the graph are pixels of the image, and edges connect pixels next to each other, forming a lattice.

Convex envelope of graph cuts: Recall that the convex envelope of $F(\text{supp}(\boldsymbol{x}))$ over the unit ℓ_∞-ball is given by $\Theta_\infty(\boldsymbol{x}) = \inf_{\boldsymbol{s}\in[0,1]^d}\{f^-(\boldsymbol{x}) : \boldsymbol{s} \geq |\boldsymbol{x}|\}$ (see Lemma 1), where f^- is the *convex closure* of F, i.e., the largest convex function defined on $[0,1]^d$ that always lower bounds F (see Appendix A.2). For example, when F is a monotone submodular function, its convex closure is given by its Lovász extension $f^- = f_L$, and its convex envelope by $\Theta_\infty(\boldsymbol{x}) = f_L(|\boldsymbol{x}|)$. Unfortunately, for any symmetric functions, the resulting convex envelope is the zero function, i.e., $\Theta_\infty(\boldsymbol{x}) = 0, \forall \boldsymbol{x} \in \mathbb{R}^d$, since $f^-(\mathbb{1}) = F(V) = F(\emptyset) = 0$. In particular, the convex envelope of cut functions, which are symmetric submodular functions, is then the zero function.

As an alternative, one can use instead the convex closure of F as the second most natural convex relaxation. In the case of symmetric submodular functions, where $f^- = f_L$, such choice is

further motivated by the result of [Bac11], showing that f_L is the convex envelope of the function $\boldsymbol{x} \to \min_{\alpha\in\mathbb{R}} F(\{i : x_i \geq \alpha\})$ on the set $[0,1]^d + \mathbb{R}\mathbb{1} = \{\boldsymbol{x} \in \mathbb{R}^d : \max_{i\in V} x_i - \min_{i\in V} x_i \leq 1\}$.

Lovász extension of graph cuts: The Lovász extension of cut functions is known to be the total variation semi-norm (see, e.g., [Bac11]). We provide here an elementary proof of this fact.

Proposition 19. *The Lovász extension of F_{cut} is the anisotropic discrete Total Variation semi-norm $\|\boldsymbol{x}\|_{TV} = \sum_{(i,j)\in E} |x_i - x_j|$.*

Proof. Given $\boldsymbol{x} \in \mathbb{R}^d$, we sort its components in decreasing order $x_{j_1} \geq \cdots \geq x_{j_d}$. Let $m = |E|$ and $\forall k \in V$ and $\ell \in [1, m]$, let $\sigma_k(e_\ell) = 1$ if $e_\ell \in E$ is cut by $\{j_1, \ldots, j_k\}$ and $\sigma_k(e_\ell) = 0$ otherwise, then by definition of the Lovász extension (see Definition 17):

$$\begin{aligned} F_{\text{cut}}(\boldsymbol{x}) &= \sum_{k=1}^{d-1} |\delta(\{j_1, \ldots, j_k\})|(x_{j_k} - x_{j_{k+1}}) \\ &= \sum_{k=1}^{d-1} \sum_{\ell=1}^{m} \sigma_k(e_\ell)(x_{j_k} - x_{j_{k+1}}) \\ &= \sum_{\ell=1}^{m} \sum_{k\in[s_\ell,t_\ell]} (x_{j_k} - x_{j_{k+1}}) = \|\boldsymbol{x}\|_{TV} \ , \end{aligned}$$

where $[s_\ell, t_\ell]$ is the range of indices associated with each edge e_ℓ, such that e_ℓ is cut by $\{j_1, \ldots, j_k\}$ for $k \in [s_\ell, t_\ell]$. Let $e_\ell = (i, j)$, then assuming w.l.o.g that $x_i \geq x_j$, we have $j_{s_\ell} = i$, $j_{t_\ell+1} = j$, and $\sigma_k(e_\ell) = 1$ for $k \in [s_\ell, t_\ell]$ and 0 otherwise. □

The TV semi-norm is a classical regularizer commonly used in computer vision, which leads to piecewise constant solutions [TSR+05]. More generally, the Lovász extension of general symmetric submodular functions were also shown in [Bac11] to lead to solutions with many equal values. Hence, the structure induced by such convex relaxation differ significantly from the original structure encoded in F. Indeed, F_{cut} encourages the clustering of the support of $\boldsymbol{x}$, but not of its coefficient values.

To clarify how the convex relaxation can radically alter the solutions of (5.1) in this case, consider the simple denoising example where $\boldsymbol{A} = \boldsymbol{I}$, and assume that $\boldsymbol{y}$ has full support, i.e., $\text{supp}(\boldsymbol{y}) = V$. In this case, since $F_{\text{cut}}(V) = 0$, the minimizer of (5.1) for $f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{y} - \boldsymbol{x}\|_2^2$ is simply $\boldsymbol{x}^\star = \boldsymbol{y}$. While, if we instead substitute F_{cut} by its Lovász extension $\|\boldsymbol{x}\|_{TV}$, the solutions returned by the resulting convex problem will not be equal to $\boldsymbol{x}^\star$ (unless $\boldsymbol{y}$ has constant values).

Also note that $F_{\text{cut}}(\text{supp}(\boldsymbol{x}))$ is symmetric around the origin, while its Lovász extension is not (see Figure 5.1), which makes it vulnerable to sign flip errors. One can try to correct for this by composing the Lovász extension with the absolute value, but the resulting function is only guaranteed to be convex in the case of monotonic submodular functions. For example, $\||\boldsymbol{x}|\|_{TV}$

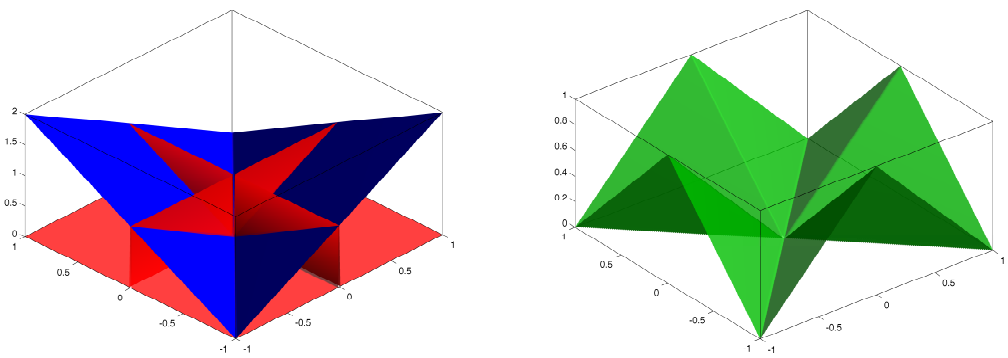


Figure 5.1: Cut function F_{cut} (red), its Lovász extension $\|x\|_{TV}$ (blue), and its Lovász extension composed with the absolute value $\||\boldsymbol{x}|\|_{TV}$ (green), in $\mathbb{R}^2$.

Algorithm 7 Discrete proximal gradient descent algorithm

1: **Input:** $\boldsymbol{x}^0 \in \mathbb{R}^d$
2: **while** *not converged* **do**
3: $\quad \boldsymbol{u} = \boldsymbol{x}^k - \frac{1}{L}\nabla f(\boldsymbol{x}^k)$
4: $\quad S = \arg\min_{A \subseteq V} F(A) - \frac{L}{2}\|\boldsymbol{u}_A\|_2^2$
5: $\quad \boldsymbol{x}^{k+1} = \boldsymbol{u}_S$
6: $\quad \boldsymbol{x}^{k+1} = \arg\min_{\boldsymbol{x}:\text{supp}(\boldsymbol{x}) \subseteq S} f(\boldsymbol{x})$ (optional full correction step)
7: **end while**

is not convex (see Figure 5.1). In the numerical experiments, we exploit this weakness to show that the TV semi-norm has poor performance when we perturb the signal with random sign flips.

5.3 Discrete proximal gradient descent method

We assume that f has an L-Lipschitz continuous gradient (i.e, $\forall\ \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d, \|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|_2 \leq L\|\boldsymbol{x} - \boldsymbol{y}\|_2$). We propose an iterative majorization-minimization scheme for solving (5.1) in Algorithm 7. Given its Lipschitz constant L, we have the following bound on $f(\boldsymbol{x})$:

$$f(\boldsymbol{x}) \leq f(\boldsymbol{x}') + \langle \nabla f(\boldsymbol{x}'), \boldsymbol{x} - \boldsymbol{x}' \rangle + \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{x}'\|_2^2 := Q(\boldsymbol{x}, \boldsymbol{x}') \tag{5.2}$$

for all $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^d$. Similar to the classical convex proximal gradient descent method (see Section 1.4.1), Algorithm 7 proceeds by minimizing at each iteration the majorizer $Q(\boldsymbol{x}, \boldsymbol{x}^k) + F(\text{supp}(\boldsymbol{x}))$ of $f(\boldsymbol{x}) + F(\text{supp}(\boldsymbol{x}))$. This corresponds to the iterates $\boldsymbol{x}^{k+1} \in \text{prox}_{F/L}(\boldsymbol{x}^k - \frac{1}{L}\nabla f(\boldsymbol{x}^k))$, where prox_F is a *discrete proximal operator*[3] defined by

$$\text{prox}_F(\boldsymbol{u}) = \arg\min_{\boldsymbol{x} \in \mathbb{R}^d} \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{u}\|_2^2 + F(\text{supp}(\boldsymbol{x})) \tag{5.3}$$

[3]When F is an indicator function, this reduces to the discrete projection operator (see Section 1.5).

The update $\widehat{\boldsymbol{x}} \in \operatorname{prox}_F(\boldsymbol{u})$ can be computed by finding an optimal set

$$S \in \arg\min_{A \subseteq V} F(A) - \frac{1}{2}\|\boldsymbol{u}_A\|_2^2 \tag{5.4}$$

and setting $\widehat{\boldsymbol{x}}_S = \boldsymbol{u}_S$. Note that the function $M(S) = \frac{1}{2}\|\boldsymbol{u}_S\|_2^2$ is a *modular* function.

The proximal update step (lines 3-5) is followed by an optional full correction step (line 6), where we minimize the original loss function f over the current estimate of the support. When it can be solved efficiently (e.g., when f is the least squares loss), the full correction step significantly improves the performance of Algorithm 7.

It is easy to see that the objective is guaranteed to be monotonically decreasing, which is characteristic of majorization-minimization algorithms. We defer the theoretical characterization of the solutions returned by Algorithm 7 to future work.

Proposition 20. *At each iteration, the new estimate $\boldsymbol{x}^{k+1}$ produced by Algorithm 7 satisfies $f(\boldsymbol{x}^{k+1}) + F(\operatorname{supp}(\boldsymbol{x}^{k+1})) \leq f(\boldsymbol{x}^k) + F(\operatorname{supp}(\boldsymbol{x}^k))$, which implies convergence.*

Proof. Let $S^{k+1} = \operatorname{supp}(\boldsymbol{x}^{k+1})$ and $S^k = \operatorname{supp}(\boldsymbol{x}^k)$. Based on the discussion above, we have

$$\begin{aligned} f(\boldsymbol{x}^{k+1}) + F(S^{k+1}) &\leq f(\boldsymbol{u}_{S^{k+1}}) + F(S^{k+1}) \\ &\leq Q(\boldsymbol{u}_{S^{k+1}}, \boldsymbol{x}^k) + F(S^{k+1}) \\ &\leq Q(\boldsymbol{u}_{S^k}, \boldsymbol{x}^k) + F(S^k) \\ &\leq Q(\boldsymbol{x}^k, \boldsymbol{x}^k) + F(S^k) \\ &= f(\boldsymbol{x}^k) + F(S^k), \end{aligned}$$

Therefore, at each iteration the objective value does not increase and it is lower bounded by 0, hence the algorithm converges. □

Computational complexity per iteration: Computing the discrete proximal operator is in general NP-Hard. But for several classes of structures, including structures that do not admit tractable or meaningful convex envelopes, it can be efficiently solved exactly or approximately.

- **Submodular penalties:** When F is a submodular function, problem (5.4) becomes a submodular minimization problem, which can be solved efficiently (see Section A.1). In the special case of cut functions, it can be solved even more efficiently with a min s-t-cut algorithm [GR98]. For further examples of structures in this class, see e.g., [Bac11] and [OB16]. It is worth noting here that projecting on a feasible set described by a submodular function, i.e., $\operatorname{prox}_{\iota_{F(\operatorname{supp}(\boldsymbol{x}) \leq \lambda)}}(\boldsymbol{u})$ is in contrast NP-Hard in general. As a result, existing non-convex methods are inapplicable.

Table 5.1: A summary of the regularizers used in experiments

Model	Regularizer
Sparse graph cut (GC + L0)	$\lambda F_{\text{cut}}(\text{supp}(\boldsymbol{x})) + \tau\lvert\,\text{supp}(\boldsymbol{x})\rvert$
Total Variation (TV)	$\lambda\lVert\boldsymbol{x}\rVert_{TV}$
Sparse TV (TV + L1)	$\lambda\lVert\boldsymbol{x}\rVert_{TV} + \tau\lVert\boldsymbol{x}\rVert_1$

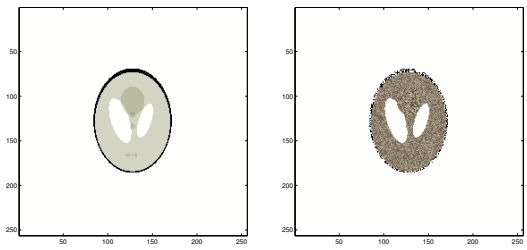


Figure 5.2: Shepp-Logan phantom: Original (left) and *Dirty*, i.e., with randomized signs (right).

- **ILP penalties:** When F can be expressed by an integral linear program as in Chapter 2, problem (5.4) becomes a linear program, which can be solved efficiently (see Section 2.4). For examples of structures in this class, see Section 2.5. Similar to submodular functions, projecting on a feasible set described by an ILP penalty is also NP-Hard in general.

- **Supermodular penalties:** When F is a supermodular[4] function, problem (5.4) becomes a submodular maximization problem, which can be approximated efficiently with a greedy algorithm, achieving $1/2$-approximation ratio [BFSS15]. Supermodular regularizers are used to promote diversity of features, leading to more representative features and better noise robustness. For examples of structures in this class, see e.g., [DDK12]. In contrast, computing the convex envelope of a supermodular functions is in general NP-Hard. Similarly, projecting on a feasible set described by a supermodular function is also NP-Hard.

- **Set cover penalties:** When F is the minimal weighted set cover penalty (see Def. 5), problem (5.4) is again a submodular maximization problem, and thus it can be approximated efficiently. In this case, projecting over group cover constraints can also be approximated efficiently by a greedy algorithm [JRD16]. In contrast, recall that computing the convex envelope of this function, for non-TU group structures, is NP-Hard (see Section 2.5.1).

5.4 Experiments

We perform a compressive sensing experiment to highlight the differences between solutions of (5.1) and its convex relaxation via the Lovász extension. We take dimensionality reducing measurements of a "clustered" sparse $\boldsymbol{x}^\natural$ via $\mathbf{A}$ and then seek to minimize $f(\boldsymbol{x}) = \frac{1}{2}\lVert\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}\rVert_2^2$ with a regularizer matched to the structure of $\boldsymbol{x}^\natural$. Our linear measurements $\mathbf{A}$ are randomly subsampled Fourier coefficients of $\boldsymbol{x}$, and hence, the Lipschitz constant of f is $L = 0.5$.

[4] A set function F is supermodular if it satisfies $F(B \cup \{i\}) - F(B) \geq F(A \cup \{i\}) - F(A), \forall A \subseteq B, i \in B^c$.

We compare the performance of three regularizers as summarized in Table 5.1. To promote sparsity, we add a cardinality term to F_{cut}. The corresponding Lovász extension is the sparse total variation regularizer, i.e., the TV semi-norm plus the ℓ_1-norm[5]. We also include the total variation regularizer alone to emphasize that its solutions are significantly different from regularization with F_{cut} directly.

We consider the standard Shepp-Logan phantom image of size 256×256 pixels. The resulting image is sparse ($s = 8084$ non-zero pixels) with its coefficients forming constant value clusters, see Figure 5.2(left). This image suits the TV models that encourage the signal coefficients to have constant values. We then randomly flip the signs of the coefficients, obtaining the *Dirty* phantom, see Figure 5.2(right). In this case, the TV models can enforce an incorrect structure as the true coefficient values are not smooth. However, the sign change does not affect the sparse graph cut penalty, since it is agnostic to the coefficient values and only cares about whether they cluster.

We show recovery performance using $n = 1.5s$ samples, which is less than the theoretically minimum number of samples for ℓ_0 recovery (i.e., $n = 2s$). Hence, without the structured sparsity model, it is impossible to do tractable guaranteed recovery. We measure performance with the relative recovery error, $E = \|\widehat{\boldsymbol{x}} - \boldsymbol{x}^\natural\|_2^2 / \|\boldsymbol{x}^\natural\|_2^2$, where $\widehat{\boldsymbol{x}}$ is the estimated image and $\boldsymbol{x}^\natural$ the original one. The regularization parameters λ and τ have been tuned according to E to yield the best possible result for each model. Since, the convex models might not yield exactly sparse solutions, we adopt the debiasing heuristic of finding the support of the coefficients that are greater in magnitude than a threshold. We do this by visually inspecting the histogram of the solutions. The debiased estimate is then given by the least squares solution on the estimated support. Figure 5.3 presents the recovered images for the original and *Dirty* phantom respectively, while Table 5.2 contains the relative recovery errors. In the figures, we use the log scale of the coefficients' absolute value to accentuate the errors.

The debiased estimates perform well in recovering the standard image, while the TV and TV + L1 penalties fail to set the background exactly to zero. Our method recovers the image with no need for debiasing since it correctly identifies the support of the signal. Note that in the original phantom, the values of the pixels inside the eyes of the phantom are not exactly zero. Hence the sparse graph cut model actually perfectly recovers the entire support. As expected, on the Dirty Shepp-Logan, TV does not perform well. Debiasing helps in this case too, but it is not able to recover the correct support.

5.5 Discussion

We proposed to use a simple non-convex iterative algorithm, *the discrete proximal gradient descent method*, to tackle directly structured sparse learning problems, without relying on convex

[5]Note that the convex envelope of $\lambda F_{\text{cut}}(\operatorname{supp}(\boldsymbol{x})) + \tau|\operatorname{supp}(\boldsymbol{x})|$ is not the zero function anymore (unless $\tau = 0$). However, as the aim of this experiment is to illustrate the effect of using the convex closure as a "heuristic" convex relaxation, when no meaningful convex envelope exists, we will not consider the true convex envelope here.

Table 5.2: Relative Recovery Errors

Model	Original phantom	*Dirty* phantom
TV	0.25	0.13
TV + L1	0.005	0.14
TV (debiased)	$\mathbf{8.8 * 10^{-12}}$	0.013
TV + L1 (debiased)	$\mathbf{8.8 * 10^{-12}}$	0.027
GC + L0	$1.2 * 10^{-10}$	$\mathbf{1.4 * 10^{-11}}$

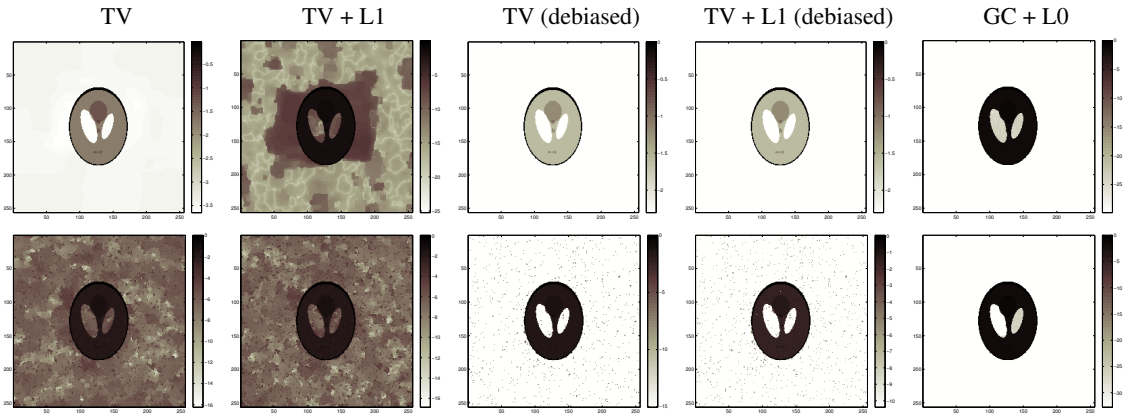


Figure 5.3: Top: Clean Phantom recovery. Bottom: Dirty Phantom recovery. The colorbar changes with each figure.

relaxations. Such algorithm is of particular interest, in cases where the convex envelope of the underlying structure is either intractable or trivial, and thus disciplined convex approaches are inapplicable. This algorithm provides also an alternative to the *the discrete projected gradient descent method*, for cases where the discrete projection is difficult. Our preliminary numerical results indicate that the solutions returned by the proposed algorithm upon convergence can have significant recovery benefits compared to solutions obtained by "heuristic" convex relaxations.

The missing piece in our discussion is the theoretical characterization of the quality of the solutions returned by the proposed algorithm, both with exact and approximate proximal operators.

Open question 8. Existing non-convex methods, based on variants of the discrete projected gradient descent method, guarantee near-optimal recovery, under some conditions on the design matrix, such as the restricted isometry property (RIP) [BCDH10]. Such results were further extended, e.g., in [HIS15a] and [JRD16], to allow for inexact projections. *Can similar theoretical guarantees be obtained for the discrete proximal gradient descent method?*
Recently, [BH18] studied Problem (5.1) when F is a set cover penalty or constraint (or both), with non-overlapping groups, and showed that the discrete proximal gradient descent method is guaranteed to return a stationary point in this setting. One promising starting point is then to characterize the properties of stationary points of Problem (5.1), under RIP-like conditions.

6 MAP Estimation for Mixture Models with Combinatorial Priors

6.1 Introduction

In this chapter, we continue with the non-convex approach to structured sparse learning, but from a probabilistic view. We have so far assumed that signals to be estimated are *exactly sparse*. Real-world signals though rarely satisfy this assumption. In this chapter, we consider instead signals consisting of a *mixture* of large and small coefficients, with only *few* large coefficients, and where the small coefficients are *not exactly zero*. We further assume that the large coefficients satisfy a structured sparsity model.

6.1.1 Related work

Most existing work in structured sparsity typically assume that small coefficients in signals of interest are negligible if not exactly sparse, and thus they do not explicitly account for them. In this setting, *compressible signals*, i.e., signals that are not exactly sparse, but are well-approximated by sparse signals, were considered in the compressive sensing (CS) literature. Compressible signals have coefficients that decay rapidly to zero, when sorted in order of decreasing magnitude[1]. Most of the theory on CS extends to this model. Furthermore, [DHCB09] also extended the non-convex structured sparsity framework of [BCDH10] to compressible signals. This framework guarantees recovery of structured compressible signals, but it does not explicitly account for small coefficients in its recovery algorithm. It also requires a *discrete projection* step, which is NP-Hard for general combinatorial structures, including ones expressed by submodular functions and integral linear programs (ILP).

In the Bayesian framework, the two-state mixture models we consider in this chapter are well-studied classical models. They were used in the context of structured sparsity, e.g., in [SBB06] and [DWB08], to express approximately sparse signals, but the structures considered on the large coefficients were limited to simple sparsity and tree sparsity.

[1] Signals constrained to an ℓ_p-ball, with $p \leq 1$, satisfy this rapid decay.

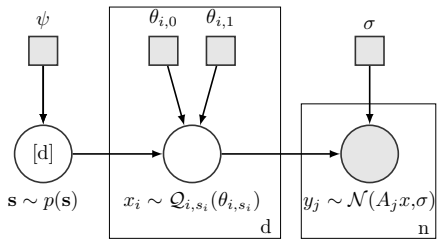


Figure 6.1: Graphical model

6.1.2 Contributions

We introduce a Bayesian mixture model with combinatorial priors, which allows to incorporate both continuous and discrete prior knowledge on the signal. We extend the majorization-minimization algorithm presented in Chapter 5 to approximate the corresponding *maximum a posteriori* estimate. The resulting algorithm is again guaranteed to converge, and efficient for several classes of structures, including submodular, supermodular and ILP penalties. Our numerical results show that the proposed algorithm can take full advantage of all available prior information on the signal, while for non-truly sparse signals, state-of-the-art methods are capable of leveraging only a part of it. For sparse signals, our algorithm can be used to further improve on existing methods.

This chapter is based on the joint work with Luca Baldassarre and Volkan Cevher [EHBC14].

6.2 Mixture model with combinatorial priors

In a Bayesian framework, prior knowledge is encoded by placing a prior distribution on the signal $\boldsymbol{x} \in \mathbb{R}^d$ that favors the desired structure. We differentiate between two kinds of prior knowledge: a discrete structure on the state (small/large) of the coefficients (e.g., the large coefficients are clustered together), and a continuous structure on the values of the coefficients of the signal (e.g., the coefficients are sampled from a Gaussian distribution with fixed variance). In this paper, we investigate a model that leverage both types of structure.

We consider a Bayesian mixture model where the signal is generated by a mixture of probability distributions. Mixture models provide flexibility to model real-world signals, and are often used as priors in practice. In particular, we assume each component x_i is independently drawn from one of two possible distributions $\mathcal{Q}_0$ and $\mathcal{Q}_1$, which corresponds to two possible states of x_i. To this end, we introduce for each x_i a latent binary random state variable $s_i \in \{0, 1\}$ which indicates the distribution x_i was drawn from, i.e., $x_i \sim \mathcal{Q}_{s_i}(\theta_{i,s_i})$ where θ_{i,s_i} are the parameters of $\mathcal{Q}_{s_i}$ (see Section 6.4.2 for examples of continuous priors). For example, if $\mathcal{Q}_0$ is a Dirac delta distribution centered at zero, i.e., all small coefficients are exactly zero, $\boldsymbol{s}$ would correspond to the support of $\boldsymbol{x}$, i.e., $\boldsymbol{s} = \operatorname{supp}(\boldsymbol{x})$.

The discrete structure is encoded by a prior distribution over the state vector $\boldsymbol{s}$ that ensures that certain state configurations are more likely than others. In particular, we assume that the discrete structure can be captured by a prior $p(\boldsymbol{s}) = \exp(-F(\mathrm{supp}(\boldsymbol{s})))$, where F is a set function with parameters ψ (see Section 6.4.3 for examples of discrete priors). We can treat set functions F as functions on the boolean hypercube $\{0,1\}^d$, i.e., we use $F(\boldsymbol{s}) = F(\mathrm{supp}(\boldsymbol{s}))$.

For simplicity, we assume all hyperparameters in our model $(\theta_{i,1}, \theta_{i,0}, \psi)$ are known. Learning the hyperparameters is deferred to future work. A graphical summary of the considered model is depicted in Figure 6.1, for the case where the noise $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$ (see Section 6.4.1 for examples of priors on the noise).

We propose to estimate $\boldsymbol{x}$ by computing its maximum a posteriori (MAP) estimate $\hat{\boldsymbol{x}}$. However, the presence of the discrete component F in our model renders the optimization difficult. We present an extension of the Majorization-Minimization algorithm introduced in Chapter 5, that iteratively maximizes the log-posterior $\log p(\boldsymbol{x}, \boldsymbol{s}|\boldsymbol{y})$, with guaranteed convergence.

6.3 Majorization-minimization algorithm

In what follows, we denote the likelihood distribution by $p(\boldsymbol{y}|\boldsymbol{x}) = \exp(-\mathcal{L}_y(\boldsymbol{x}))$, where $\mathcal{L}_y(\boldsymbol{x})$ is some suitable data fidelity term. In our model, we make the following assumptions:

A1. The loss function $\mathcal{L}_y(\boldsymbol{x}) = -\log p(\boldsymbol{y}|\boldsymbol{x})$ is *smooth* with L-Lipschitz continuous gradient, i.e., $\forall\, \boldsymbol{x}, \boldsymbol{x}' \in \mathrm{dom}(\mathcal{L}_y)$, $\|\nabla \mathcal{L}_y(\boldsymbol{x}) - \nabla \mathcal{L}_y(\boldsymbol{x}')\|_2 \leq L\|\boldsymbol{x} - \boldsymbol{x}'\|_2$.

A2. The variables x_i are independent given s_i, and accordingly the continuous prior $G(\boldsymbol{x}|\boldsymbol{s}) = -\log p(\boldsymbol{x}|\boldsymbol{s})$ is *separable*, i.e. $G(\boldsymbol{x}|\boldsymbol{s}) = -\sum_{i=1}^d \log p(x_i|s_i)$.

A3. The proximal operator of G, i.e., $\mathrm{prox}_{G(\cdot|\boldsymbol{s})}(\boldsymbol{z}) := \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} \frac{1}{2}\|\boldsymbol{z} - \boldsymbol{x}\|_2^2 + G(\boldsymbol{x}|\boldsymbol{s})$ has a closed form solution.

A4. The discrete prior $F(\boldsymbol{s}) = -\log p(\boldsymbol{s})$ admits an efficient minimizer for $\min_{S\subseteq V} M(S) + F(S)$, for any modular function M.

We provide examples of interesting priors that satisfy these assumptions in Section 6.4. Given these assumptions, we want to compute the MAP estimate of $[\boldsymbol{x}, \boldsymbol{s}]$.

$$
\begin{aligned}
[\boldsymbol{x}^\star, \boldsymbol{s}^\star] &= \underset{\boldsymbol{x}\in\mathbb{R}^d, \boldsymbol{s}\in\{0,1\}^d}{\arg\max}\; p(\boldsymbol{x}, \boldsymbol{s}|\boldsymbol{y}) \\
&= \underset{\boldsymbol{x}\in\mathbb{R}^d, \boldsymbol{s}\in\{0,1\}^d}{\arg\min}\; -\log p(\boldsymbol{y}|\boldsymbol{x}) - \log p(\boldsymbol{x}|\boldsymbol{s}) - \log p(\boldsymbol{s}) \\
&= \underset{\boldsymbol{x}\in\mathbb{R}^d, \boldsymbol{s}\in\{0,1\}^d}{\arg\min}\; \mathcal{L}_y(\boldsymbol{x}) + G(\boldsymbol{x}|\boldsymbol{s}) + F(\boldsymbol{s}) \qquad (6.1)
\end{aligned}
$$

Unfortunately, computing the MAP estimate (6.1) is NP-Hard[2]. Here, we aim to efficiently

[2] If $G(\boldsymbol{x}|\boldsymbol{s}) = \iota_{\mathrm{supp}(\boldsymbol{x})=\boldsymbol{s}}(\boldsymbol{x})$, we recover the original structured sparse learning problem (1.1)

Algorithm 8 MAP-MM algorithm

1: **Input:** $\boldsymbol{x}^0 \in \mathbb{R}^d$
2: **while** *not converged* **do**
3: $\quad \boldsymbol{u} = \boldsymbol{x}^t - \frac{1}{L}\nabla\mathcal{L}_y(\boldsymbol{x}^t)$
4: $\quad \widehat{\boldsymbol{x}}(\boldsymbol{s}) = \mathrm{prox}_{G(\cdot|\boldsymbol{s})/L}(\boldsymbol{u})$
5: $\quad \boldsymbol{s}^{t+1} = \arg\min_{\boldsymbol{s}\in\{0,1\}^d} \frac{L}{2}\|\widehat{\boldsymbol{x}}(\boldsymbol{s}) - \boldsymbol{u}\|_2^2 + G(\widehat{\boldsymbol{x}}(\boldsymbol{s})|\boldsymbol{s}) + F(\boldsymbol{s})$
6: $\quad \boldsymbol{x}^{t+1} = \widehat{\boldsymbol{x}}(\boldsymbol{s}^{t+1})$
7: $\quad \boldsymbol{x}^{t+1} = \arg\min_{\boldsymbol{x}} \mathcal{L}_y(\boldsymbol{x}) + G(\boldsymbol{x}|\boldsymbol{s}^{t+1})$ (optional full correction step)
8: **end while**

compute numerically good approximations to the MAP estimator.

Given our assumptions, the objective function in (6.1) can be iteratively minimized by the majorization-minimization scheme of Algorithm 8. By assumption A1, the loss function admits the following quadratic upper bound:

$$\mathcal{L}_y(\boldsymbol{x}) \leq \mathcal{L}_y(\boldsymbol{x}') + \langle \nabla\mathcal{L}_y(\boldsymbol{x}'), \boldsymbol{x} - \boldsymbol{x}'\rangle + \frac{L}{2}\|\boldsymbol{x} - \boldsymbol{x}'\|_2^2 := Q(\boldsymbol{x}, \boldsymbol{x}') \tag{6.2}$$

for all $\boldsymbol{x}, \boldsymbol{x}' \in \mathbb{R}^d$. Therefore, the objective function in (6.1) is upper bounded by $Q(\boldsymbol{x}, \boldsymbol{x}') + G(\boldsymbol{x}|\boldsymbol{s}) + F(\boldsymbol{s})$. At each iteration $t+1$, Algorithm 8 proceeds by minimizing this majorizer with $\boldsymbol{x}' = \boldsymbol{x}^t$, the estimate obtained at the previous iteration,

$$\begin{aligned} &\min_{\boldsymbol{x},\boldsymbol{s}} Q(\boldsymbol{x}, \boldsymbol{x}^t) + G(\boldsymbol{x}|\boldsymbol{s}) + F(\boldsymbol{s}) = \\ &\min_{\boldsymbol{s}} \min_{\boldsymbol{x}} \frac{L}{2}\|\boldsymbol{x} - (\boldsymbol{x}^t - \frac{1}{L}\nabla\mathcal{L}_y(\boldsymbol{x}^t))\|_2^2 + G(\boldsymbol{x}|\boldsymbol{s}) + F(\boldsymbol{s}) \end{aligned} \tag{6.3}$$

For a fixed $\boldsymbol{s}$, the inner minimization with respect to $\boldsymbol{x}$ reduces to the proximal operator of $G(\cdot|\boldsymbol{s})$, which by assumption A3 has a closed form solution. Let $\widehat{\boldsymbol{x}}(\boldsymbol{s}) = \mathrm{prox}_{G(\cdot|\boldsymbol{s})/L}(\boldsymbol{x}^t - \frac{1}{L}\nabla\mathcal{L}_y(\boldsymbol{x}^t))$, then the minimization in (6.3) with respect to $\boldsymbol{s}$ is equivalent to:

$$\min_{\boldsymbol{s}\in\{0,1\}^d} \frac{L}{2}\|\widehat{\boldsymbol{x}}(\boldsymbol{s}) - (\boldsymbol{x}^t - \frac{1}{L}\nabla\mathcal{L}_y(\boldsymbol{x}^t))\|_2^2 + G(\widehat{\boldsymbol{x}}(\boldsymbol{s})|\boldsymbol{s}) + F(\boldsymbol{s}) \tag{6.4}$$

Since G is separable, then the function $M(\boldsymbol{s}) = \frac{L}{2}\|\widehat{\boldsymbol{x}}(\boldsymbol{s}) - (\boldsymbol{x}^k - \frac{1}{L}\nabla\mathcal{L}_y(\boldsymbol{x}^k))\|_2^2 + G(\widehat{\boldsymbol{x}}(\boldsymbol{s})|\boldsymbol{s})$ is *modular*. Hence by assumption A4, problem (6.4) can be efficiently solved.

Given the current estimate of the state vector $\boldsymbol{s}^{t+1}$, we update our estimate $\boldsymbol{x}^{t+1}$ with a full correction step (line 7), where we minimize the original objective function with $\boldsymbol{s} = \boldsymbol{s}^{t+1}$, if it can be done efficiently, otherwise we use $\boldsymbol{x}^{t+1} = \widehat{\boldsymbol{x}}(\boldsymbol{s}^{t+1})$.

Proposition 21. *Algorithm 8 produces a sequence $\boldsymbol{x}^{t+1}$ that satisfies $p(\boldsymbol{x}^{t+1}, \boldsymbol{s}^{t+1}|\boldsymbol{y}) \geq p(\boldsymbol{x}^t, \boldsymbol{s}^t|\boldsymbol{y})$ which implies convergence.*

The proof of Proposition 21 follows from similar arguments as in Proposition 20.

6.4 Examples

In this section, we present some examples of signal priors that fit in our framework.

6.4.1 Priors on the noise

Classical priors on the noise do indeed satisfy assumption A1. For example when $\boldsymbol{\varepsilon}$ is a zero-mean Gaussian noise with covariance $\sigma^2\boldsymbol{I}$, i.e., $p(\boldsymbol{y}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{A}\boldsymbol{x}, \sigma^2\boldsymbol{I})$, the data fidelity term is the usual least squares loss function $\mathcal{L}_y(\boldsymbol{x}) = \frac{1}{2\sigma^2}\|\boldsymbol{y}-\boldsymbol{A}\boldsymbol{x}\|_2^2 + n\log(\sqrt{2\pi}\sigma)$. Another example is the logistic loss function, commonly used in classification, $\mathcal{L}_y(\boldsymbol{x}) = \sum_{i=1}^n \log(1+\exp(-y_i(\mathbf{a}_i^T\boldsymbol{x})))$, where $\mathbf{a}_i$ is the i-th row of $\boldsymbol{A}$, which corresponds to the prior $p(\boldsymbol{y}|\boldsymbol{x}) = \prod_{i=1}^n \frac{1}{1+\exp(-y_i(\mathbf{a}_i^T\boldsymbol{x}))}$.

6.4.2 Priors on the continuous structure of the signal

The Gaussian and Laplacian distributions are examples of distributions with closed form proximal operators. Then sampling each x_i independently from a Gaussian or Laplacian distribution with parameters depending on s_i, leads to a continuous prior satisfying assumptions A2 and A3. For $p(x_i|s_i) = \mathcal{N}(\mu_{i,s_i}, \sigma_{i,s_i}^2)$, the proximal operator $\widehat{\boldsymbol{x}}(\boldsymbol{s}) = \mathrm{prox}_{G(\cdot|\boldsymbol{s})/L}(\boldsymbol{u})$ used in Algorithm 8 is given by:

$$\widehat{x}_i(s_i) = \frac{Lu_i + \mu_{i,s_i}/\sigma_{i,s_i}^2}{L + 1/\sigma_{i,si}^2},$$

and for $p(x_i|s_i) = \text{Laplace}(\mu_{i,s_i}, \sigma_{i,s_i})$, it is given by:

$$\widehat{x}_i(s_i) = \mu_{i,s_i} + S(u_i, 1/(L\sigma_{i,s_i})),$$

where $S(x,\lambda) = \mathrm{sign}(x)\max\{|x|-\lambda, 0\}$ is the standard soft-thresholding operator.
The mixture of two (or more) Gaussians, i.e $\mathcal{Q}_{s_i}(\theta_{i,s_i}) = \mathcal{N}(\mu_{i,s_i}, \sigma_{i,s_i}^2)$ such that $\sigma_{i,1} > \sigma_{i,0}$, is ubiquitous in literature (see, e.g., [DWB08, BND12, JXC08]), due to their simplicity and effectiveness in modeling real-world signals. One can also use a Gaussian-Laplacian mixture, i.e $\mathcal{Q}_1(\theta_{i,1}) = \mathcal{N}(\mu_{i,1}, \sigma_{i,1}^2)$, and $\mathcal{Q}_0(\theta_{i,0}) = \text{Laplace}(\mu_{i,0}, \sigma_{i,0})$ where the Laplacian distribution is used as a sparsity promoting prior [See08]. Another example is the Laplacian mixture model, an analogue to the Gaussian mixture model, that is better suited to model signals with "peaky" distributions, see, e.g., [GO10, AZG07, MS]. To enforce exact sparsity, the Dirac delta distribution can be used. The mixture of a Gaussian and delta distribution is known as the spike and slab model, see e.g., [IR$^+$05].

6.4.3 Priors on the discrete structure of the signal

Examples of set functions that satisfy assumption A4 include submodular, supermodular, ILP penalties and minimal set cover functions (see Section 5.3). We highlight below three examples of interesting discrete structures expressed by submodular functions. In what follows, we refer to coefficients with $s_i = 1$ as "large", and $s_i = 0$ as "small". Accordingly, large coefficients are drawn from the distribution of larger variance $\sigma_{i,1}$, and small ones from the distribution of smaller variance $\sigma_{i,0}$.

Approximately sparse model

The simplest discrete prior on $\boldsymbol{x}$ is the expected number k of large coefficients, which is equivalent to sparsity for signals whose small coefficients are exactly zero. In this model, each binary variable s_i is drawn independently from the same Bernoulli distribution with known parameter k/d. We then have $p(\boldsymbol{s}) = \prod_{i=1}^{d} (\frac{k}{d})^{s_i} (1 - \frac{k}{d})^{1-s_i}$ and

$$-\log p(\mathbf{s}) = \sum_{i=1}^{d} \left(s_i \log \left(\frac{d-k}{k} \right) - \log \left(1 - \frac{k}{d} \right) \right),$$

which is a modular function of $\boldsymbol{s}$. When $k \ll d$, this discrete prior used in conjunction with the mixture model captures well the structure of approximately sparse signals, where small values are not small enough to be ignored. Note that for $\sigma_0 = 0$, we recover the standard sparsity model.

The special case of Gaussian mixture with the above sparsity prior was considered in [SBB06] to recover approximately sparse signals. However, the proposed recovery algorithm relies on a specific measurement scheme, and is not guaranteed to converge.

Markov tree model

Moving beyond simple sparsity priors, one can consider priors where each binary variable s_i is drawn from a Bernoulli distribution with parameters that depends on the index i. In particular, we consider the Markov Tree Gaussian mixture model described in [DWB08] which assumes that the variables x_i are organized over a given tree $\mathcal{T}$, and their values tend to decay from root to leaves. This model provides a good description of wavelet coefficients encountered in many classes of signals [DWB08].

Formally, we have $p(\mathbf{s}) = \prod_{i=1}^{d} \mathcal{B}(1, p_i)$, where p_i depend on the level of the variable x_i in the tree $\mathcal{T}$, so that

$$-\log p(\mathbf{s}) = \sum_{i=1}^{d} \left(s_i \log \left(\frac{1-p_i}{p_i} \right) - \log (1 - p_i) \right),$$

which is again a modular function of $\boldsymbol{s}$.

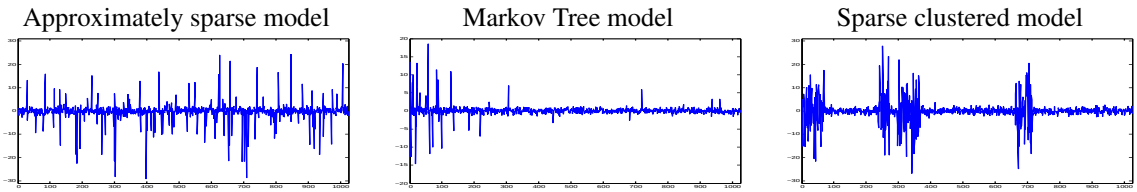


Figure 6.2: Signals sampled from each model, for $\sigma_1/\sigma_0 = 10$, and other parameters as described in the text.

Sparse clustered model

We revisit the clustering model discussed in Section 5.2. Recall that clustering can be naturally enforced via cut functions defined over an undirected graph $\mathcal{G} = (V, E)$, where the vertices are the d indices, and the edges connect "neighboring" indices (e.g., adjacent pixels in an image). We recall the definition of the cut function, $F_{\text{cut}}(S) = |\delta(S)|$, where $\delta(S) = \{(i, j) \in E : i \in S, j \notin S\}$ is the cut-set induced by S in $\mathcal{G}$.

In the context of our model, a signal whose large coefficients are sparse and clustered can be modeled by the following prior:

$$p(\boldsymbol{s}) \propto \exp(-\lambda F_{\text{cut}}(\boldsymbol{s}) - \rho|\boldsymbol{s}|),$$

where the parameters $\lambda, \rho \geq 0$ control the level of sparsity and "clusteredness". Note that $-\log p(\boldsymbol{s})$ is then a submodular function.

Remark 5. *Note that, since the approximately sparse model and Markov tree model yield modular regularizers, our algorithm can easily handle more than two states with these priors.*

6.5 Experiments

We demonstrate our approach on the two state Gaussian mixture model with the three discrete priors described in Section 6.4.3. An example of a signal generated by each model is shown in Figure 6.2. We compare against state-of-the-art convex methods adapted to each model.

We consider a linear model, $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}^\natural + \boldsymbol{\varepsilon}$, with $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \sigma^2\boldsymbol{I})$, and $\boldsymbol{A}$ a random normalized Gaussian matrix. We measure the relative recovery error with $E = \|\hat{\boldsymbol{x}} - \boldsymbol{x}^\natural\|_2/\|\boldsymbol{x}\|_2$, and the state recovery quality with $Q = \|\hat{\boldsymbol{s}} - \boldsymbol{s}^\natural\|_0$, where $\boldsymbol{s}^\natural$ is the true state vector. We fix $\sigma = 0.01$, $\sigma_0 = 1$ and $\sigma_1 = r$, with $r = 10$ and $r = 100$. The value of r controls the sparsity of the signal; a small r leads to signals not truly sparse (see Figure6.2), while a large r leads to sparser signals. We adopt two initializations for our proposed algorithm, MAP-MM starts with $\mathbf{x}^0 = \mathbf{0}$, while MAP-MM-I starts with the estimate of the best convex competing method. Figures 6.3, 6.4, and 6.5 (left) illustrate the importance of a correct initialization of MAP-MM in the sparse case ($r = 100$), where convex approaches capture well the structure of the signal. Starting from their estimate allows MAP-MM to achieve further improvement. For the non-sparse case ($r = 10$), even MAP-MM obtains excellent performance, as shown in Figures 6.3, 6.4, and 6.5 (middle).

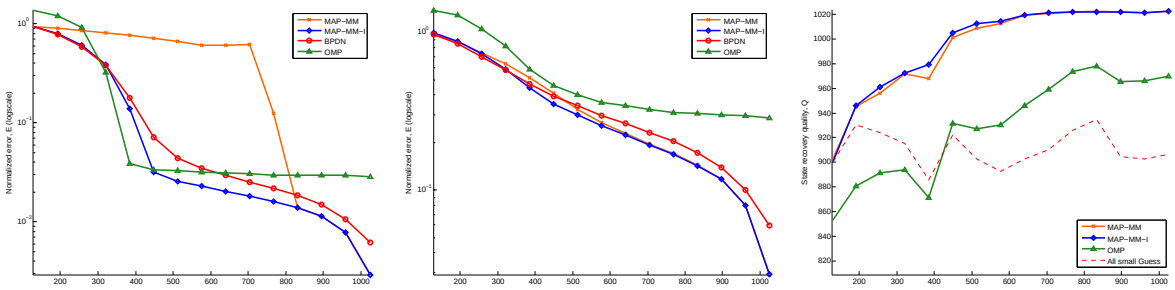


Figure 6.3: Performance of MAP-MM compared to other state-of-the arts algorithms for the approximately sparse Gaussian mixture model, in terms of signal recovery error E (in logscale) for $\sigma = 0.01$, $r = 100$ (right) and $r = 10$ (middle), and in terms of state recovery quality Q for $\sigma = 0.01, r = 10$ (left). The average number of large coefficients over the 50 simulations is 129.

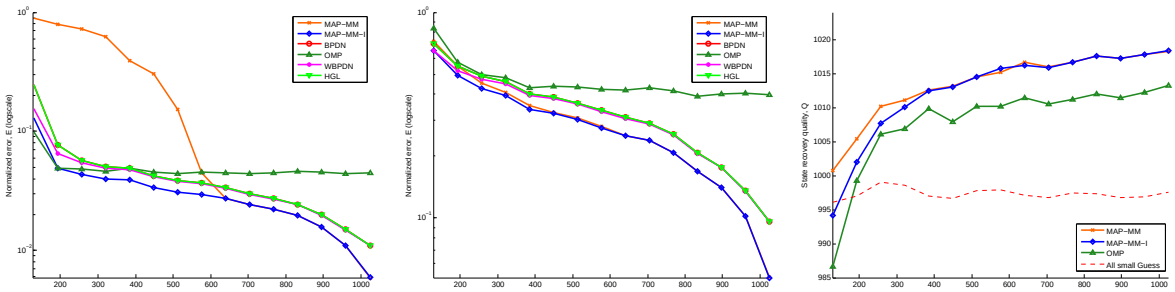


Figure 6.4: Performance of MAP-MM compared to other state-of-the arts algorithms for the Hidden Markov Tree Gaussian mixture model, in terms of signal recovery error E (in logscale) for $\sigma = 0.01$, $r = 100$ (right) and $r = 10$ (middle), and in terms of state recovery quality Q for $\sigma = 0.01, r = 10$ (left). The average number of large coefficients over the 50 simulations is 27.

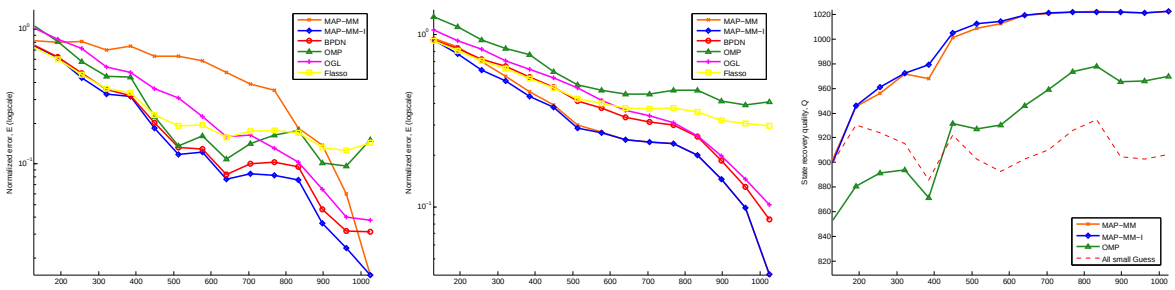


Figure 6.5: Performance of MAP-MM compared to other state-of-the arts algorithms for the sparse clustered Gaussian mixture model, in terms of signal recovery error E (in logscale) for $\sigma = 0.01, r = 100$ (right) and $r = 10$ (middle), and in terms of state recovery quality Q for $\sigma = 0.01, r = 10$ (left). The average number of large coefficients over the 50 simulations is 113.

We fix the dimension $d = 1024$, and vary n from 128 to 1024 measurements. For each n we perform 50 simulations using different randomly generated signals and measurement matrices.

For recovery of the state variables, we only consider MAP-MM and Orthogonal Matching Pursuit (OMP) [TG07], since all the other algorithms considered cannot recover the state variables. As a baseline, we compare against the recovery quality Q achieved by always guessing small (red dashed line). MAP-MM (or MAP-MM-I for $r = 100$) always outperform OMP in terms of recovering the correct states (Figures 6.3, 6.4, and 6.5 (right))

6.5.1 Approximately sparse Gaussian mixture model

We consider signals where each s_i is sampled from $\mathcal{B}(0, \frac{k}{d})$ with $k = 128$, then each x_i is independently drawn from $\mathcal{N}(0, \sigma_{s_i}^2)$ with the same large/small variances for all $i \in V := \{1, \cdots, d\}$. Figure 6.3 shows the performance of MAP-MM, OMP and Basis pursuit denoising (BPDN) [CDS98], as we vary n. OMP only exploits the true number of large coefficients in $\boldsymbol{x}$ which is clearly not enough for signals with non-negligible small coefficients (middle). BPDN uses the true variance of the noise and also accounts for sparsity by way of the ℓ_1-norm, yielding better estimates than OMP, but still worse than MAP-MM.

6.5.2 Hidden Markov tree Gaussian mixture model

We consider the Markov Tree Model proposed in [DWB08] using a binary tree. We assume that the root is always picked as a "large" coefficient with variance σ_1^2, while its child is either large with probability $p_i = 0.9$ and variance σ_1^2, or small with probability 0.1 and variance σ_0^2. The large and small variances decay according to the level and p_i depends both on the level and the state of its parent. For details, we refer to [DWB08]. We use the following parameters: $\alpha_0 = 0.2$, $\alpha_1 = 0.1$, $C_{11} = 0.5$, $C_{00} = 2$, $\gamma_0 = 5$ and $\gamma_1 = 0.5$. This choice implies that the coefficients' states are persistent across levels and the variances decay slowly, so that small coefficients at deep levels are still not negligible.

Figure 6.4 shows the performance of MAP-MM, OMP, BPDN, hierarchical group Lasso (HGL) [ZRY09] and the weighted BP algorithm with weights defined as the probability of being large (WBPDN) [DWB08], as we vary n. MAP-MM-I outperforms all the other methods for both $r = 10$ and $r = 100$. Even though WBPDN and HGL leverage the tree structure, they do not take into account the coefficient variances and hence produce poorer results.

6.5.3 Sparse clustered Gaussian mixture model

We consider the one dimensional clustering model over a chain. We sample $\boldsymbol{s}$ from $p(\boldsymbol{s}) \propto \exp(-\lambda F_{\text{cut}}(\boldsymbol{s}) - \rho|\boldsymbol{s}|)$ with the parameters $\lambda = 5$ and $\rho = 1.0986$ that yield an average sparsity of 113. Each $\boldsymbol{x}_i$ is then independently drawn from $\mathcal{N}(0, \sigma_{s_i}^2)$, with the same large/small variances

for all $i \in V$.

Figure 6.5 shows the performance of MAP-MM, OMP, BPDN, overlapping group Lasso (OGL) [JAB11] with sequential groups of length 2 and overlap 1 and fused Lasso (FLasso) [TSR+05], as we vary n. MAP-MM-I again outperforms all the other algorithms. Both OGL and FLasso promote clustering and sparsification of the coefficients, but do not exploit the continuous prior, yielding suboptimal performance.

6.6 Discussion

We proposed a mixture model with combinatorial priors, which provides a more realistic model for structured signals, which are not exactly sparse. In contrast to conventional structured sparsity models, our model explicitly account for small coefficients, instead of assuming they are exactly zero. We adapted the discrete proximal gradient descent method from Chapter 5, to directly approximate the corresponding non-convex MAP estimate, without relying on convex relaxations. In addition to the incentives considered in Chapter 5, the non-convex approach is further motivated here by the fact that the obtained MAP criterion is not easily amenable to tractable convex relaxations. Indeed, it is not clear if the function $g(\boldsymbol{x}) = \min_{\boldsymbol{s}\in\{0,1\}^d} G(\boldsymbol{x}|\boldsymbol{s}) + F(\boldsymbol{s})$ admits a tractable convex envelope, even in cases where $F(\mathrm{supp}(\boldsymbol{x}))$ does (when $\sigma_0 \neq 0$). Our numerical results demonstrate that our proposed approach can be used to improve upon solutions returned by convex methods, when a statistical characterization of the signal is available.

As in Chapter 5, the missing piece in our discussion is again the theoretical characterization of the quality of the solutions returned by the proposed majorization-minimization algorithm. As the setting in this chapter is more general, resolving Open question 8 is a necessary prerequisite for answering the following ones.

Open question 9. The MAP majorization-minimization (MAP-MM) algorithm proposed in this chapter reduces to the discrete proximal gradient descent method in the case where $G(\boldsymbol{x}|\boldsymbol{s}) = \iota_{\mathrm{supp}(\boldsymbol{x})=\boldsymbol{s}}(\boldsymbol{x})$. Assuming theoretical recovery guarantees can be obtained in the special case of sparse or compressible signals, *can such results be further extended to general signals satisfying our model? How does the sample complexity relate to the ratio σ_1/σ_0?*

Open question 10. Our proposed algorithm also assumed that true hyper-parameter values, e.g., of σ_1 and σ_0 are known, which is an unrealistic assumption. *Can these hyper-parameters be learned instead from the data, or chosen according to some theoretical recommendation?* It would be interesting to study the performance of MAP-MM, when the guessed and true values of these parameters are mismatched.

7 Conclusions

7.1 Summary

In this thesis, we addressed some computational and statistical concerns arising in structured sparsity learning problems, and extended the applicability of convex and non-convex approaches, used to solve these problems, to a wider range of structures.

By borrowing tools from integer programming, we introduced a new structured sparsity framework, which allows us to naturally describe a large range of structures, and to automatically obtain corresponding tight convex relaxations, amenable to efficient optimization. This enabled us to recover several popular structured sparsity-inducing norms in the literature, as well as define new interesting ones, which are not necessarily norms. This framework thus expands the class of practical structures that can be efficiently expressed via convex penalties.

We also presented a theoretical characterization of which combinatorial structures can be truly expressed via convex penalties. In particular, we demonstrated how the common practice of imposing homogeneity on convex regularizers leads to an unnecessary loss of structure. We further showed that non-homogeneous convex penalties can better capture structure in general, both in a geometric and a statistical sense. Such theoretical insights can help guide the design of convex structured sparsity-inducing penalties in the future.

Furthermore, we established the tractability of a non-Euclidean proximal gradient method for solving a broad class of non-smooth convex minimization problems, including ones that arise as convex relaxations of structured sparsity learning problems. In particular, we proposed efficient algorithms to compute proximal operators based on non-Euclidean norms, in various settings. We also provided an accelerated variant of this method, with a small extra computational cost. The interest in the non-Euclidean setting stems from the benefit it can entail on the computational complexity. We further illustrated numerically this benefit on some structured sparsity examples.

To handle structures where the convex approach is not applicable, we proposed to use a discrete proximal gradient descent method, which directly addresses structured sparse learning problems,

without relying on convex relaxations. This method is efficient for several classes of structures, for which existing non-convex methods are inapplicable, and is competitive with alternative heuristic convex methods.

Finally, we proposed a probabilistic model as a prior for structured signals which are only approximately sparse. We extended the discrete proximal gradient descent method to approximate the corresponding maximum a posteriori estimate. We illustrated numerically that this approach improves on state-of-the-art methods, for signals which satisfy this prior.

7.2 Future directions

The results presented in this thesis lead to several interesting directions for future research. We have presented some of them at the end of each chapter, we now discuss two additional questions.

Existing approaches (including the ones presented in this thesis) to solve structured sparsity learning problems fall into two main classes; convex and non-convex ones, each with its own advantages and disadvantages. These two approaches have so far been studied separately. An important avenue of research is then the development of methods that combine the benefits of both approaches. The following two questions propose research problems along that direction.

Open question 11. Computing and optimizing the convex envelope of structures outside the class of submodular and ILP penalties (introduced in Chapter 2) is in general intractable. *Is it possible to still design approximate efficient convex methods to handle some of these "intractable" structures in a principled way?*

In particular, we are interested in efficient algorithms to approximately solve the relaxed convex structured sparsity learning problem (1.2), where the convex surrogate is chosen, as in Chapter 3, to be the convex envelope an ℓ_p-regularized combinatorial penalty $F_p(\boldsymbol{x}) = \frac{1}{q}F(\mathrm{supp}(\boldsymbol{x})) + \frac{1}{p}\|\boldsymbol{x}\|_p^p$, even when such surrogate is intractable to evaluate.

One way to address this question is to develop a convex method able to exploit discrete approximation algorithms. Such algorithmic tools have proved to be key for the development of efficient non-convex methods, for a number of intractable structures, while convex methods have so far failed to exploit them.

A promising approach is to reformulate the convex relaxed problem (1.2) as the following convex-concave saddle point problem:

$$\min_{\boldsymbol{x}\in\mathbb{R}^d} \max_{\boldsymbol{y}\in\mathbb{R}^d} f(\boldsymbol{x}) + \boldsymbol{x}^\top \boldsymbol{y} - F_p^*(\boldsymbol{y}), \tag{7.1}$$

and to leverage the following insight, based on the results presented in Chapter 3: A subgradient of the Fenchel conjugate F_p^* can be obtained by solving a discrete proximal/projection operator. Such operations are at the heart of most non-convex algorithms, and though NP-Hard in general,

Structure class	Discrete Proximal operator	Discrete Projection operator	Example applications
Minimum weighted group cover [OJV11, HZM11]	Submodular maximization [BFSS15]	Weighted maximum cover [NWF78]	Genomics [OJV11, STM+05] Image processing [PF11]
Supermodular functions [DDK12]	Submodular maximization [BFSS15]	✗	Diverse and noise-robust feature selection [DDK12] Representative feature selection [DP05]
Weighted graph models [HIS15a]	✗	Prize-collecting Steiner tree [HIS15a]	Computer vision [HIS15a] Seismic exploration, astronomical sensing [HIS15a, SHI13].

Table 7.1: Examples of structures, whose convex envelope is intractable, but that admit efficient approximate discrete projection/proximal operators.

they can be solved exactly or approximately in several cases, via discrete optimization algorithms (see e.g., Sections 1.5 and 5.3, and Table 7.1).

Problem (7.1) is a special case of variational inequality problems (VIP), and thus can be solved via subgradient schemes such as [Nes07], when the domain of $\boldsymbol{y}$ is bounded, and the subgradient of F_p^* can be computed exactly. An interesting research problem is then to develop algorithms that builds on such subgradient schemes, and to analyse their performance when the domain of $\boldsymbol{y}$ is unbounded as in problem (7.1), and we only have access to *inexact first-order* information about F_p^*, obtained from solving the discrete proximal/projection operator approximately.

Developing such methods is challenging, but if successful would greatly enhance the class of structures that can be handled via principled convex approaches (e.g., structures in Table 7.1), and can further inspire new applications of structured sparsity in various fields.

Open question 12. In Chapter 5, we considered the discrete proximal gradient method, which is a generalization of the discrete projected gradient method, on which most existing non-convex methods for solving the structured sparsity learning problem (1.1) are based (see Section 1.5). The iterates in this algorithm are $\boldsymbol{x}^{k+1} \in \mathrm{prox}_F(\boldsymbol{x}^k - \frac{\nabla f(\boldsymbol{x}^k)}{L})$, where L is the Lipschitz constant of f, and $\mathrm{prox}_F(\boldsymbol{y}) \in \arg\min_{\boldsymbol{x}\in\mathbb{R}^d} \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|_2^2 + \frac{1}{2}F(\mathrm{supp}(\boldsymbol{x}))$ is the discrete proximal operator of F. Based on the insight discussed in Open question 11, we can equivalently view these iterates as $\boldsymbol{x}^{k+1} \in \partial F_2^*(\boldsymbol{x}^k - \frac{\nabla f(\boldsymbol{x}^k)}{L})$.

This observation raises the following question: *Can the discrete proximal gradient method be viewed as a convex algorithm solving the relaxed convex problem* (1.2)*, with F_2^{**} as a regularizer?* Such connection might seem far-fetched, as these updates do not correspond to any known convex algorithm, but if true it would explain for example the similarity between statistical results obtained by convex and non-convex approaches.

It is then worth investigating the performance of such an algorithm, through the lens of convexity, in particular with respect to solving problem (7.1), for $p = 2$. This perspective can also easily be extended for any p, by considering a general variant where $\boldsymbol{x}^{k+1} \in \partial F_p^*(\boldsymbol{x}^k - \frac{\nabla f(\boldsymbol{x}^k)}{L})$.

Such a perspective would lead to a unifying view of convex and non-convex approaches to structured sparsity. This would enable the exchange of insights between the two approaches, and would potentially lead to the design of better structured sparsity methods. Such analysis would also provide an alternative way to theoretically characterize the quality of the solutions returned

by the discrete proximal gradient method, i.e., it would answer our open question 8.

Furthermore, it is also interesting to analyze the performance of this algorithm, with *inexact* proximal/projection operators. This was studied from the discrete perspective for example in [HIS15a] and [JRD16]. Studying this from the convex perspective would provide an alternative approach to address the above open question 11.

A Submodular Analysis

In this appendix, we briefly review some concepts from submodular analysis, which we use in the thesis. For more exhaustive reviews, we refer the reader to [Fuj05, Bac13].

A.1 Submodular functions and their Lovász extensions

In this section, we consider finite-valued set functions $F : 2^V \to \mathbb{R}$ such that $F(\emptyset) = 0$. Submodular functions admit several equivalent definitions. We present here the two most commonly used ones. For other equivalent definitions, see [NWF78, Prop. 2.1] and [Bac13, Section 2.1].

Definition 16 (Submodular functions)**.** *A set function $F : 2^V \to \mathbb{R}$ is submodular iff*

$$F(A) + F(B) \geq F(A \cup B) + F(A \cap B), \qquad \forall A, B \subseteq V.$$
$$F(A \cup \{i\}) - F(A) \geq F(B \cup \{i\}) - F(B), \qquad \forall A \subseteq B \subseteq V, \forall i \in B^c.$$

If F is submodular, then $-F$ is called supermodular. If F is both submodular and supermodular, it is called modular, and it can be written as $F(A) = \sum_{i \in A} w_i$, for some $w \in \mathbb{R}^d$.

Lovász extension We can define the Lovász extension for any set-function F, but in the case of submodular functions it satisfies several nice properties, some of which we present below.

Definition 17 (Lovász extension)**.** *Given any set-function F such that $F(\emptyset) = 0$, its Lovász*

extension $f_L : \mathbb{R}^d \to \mathbb{R}$ is given by:

$$\begin{aligned} f_L(\boldsymbol{x}) &= \sum_{k=1}^{d-1} x_{j_k}[F(\{j_1, \dots, j_k\}) - F(\{j_1, \dots, j_{k-1}\})] \\ &= \sum_{k=1}^{d-1} F(\{j_1, \dots, j_k\})(x_{j_k} - x_{j_{k+1}}) + F(V)x_{j_d}. \end{aligned}$$

where $x_{j_1} \geq \cdots \geq x_{j_d}$ are the component of $\boldsymbol{x} \in \mathbb{R}^d$ sorted in decreasing order.

The Lovász extension admits other equivalent definitions too, see [Bac13, Def. 3.1]. It is easy to see from Definition 17 that f_L is an extension of F from $\{0,1\}^d$ to $\mathbb{R}^d$, i.e, $f_L(\mathbb{1}_S) = F(S), \forall S \subseteq V$.

Submodular polyhedra Two classical polyhedra associated with submodular functions are the submodular polyhedron $P(F) = \{\boldsymbol{s} \in \mathbb{R}^d : \boldsymbol{s}(A) \leq F(A), \forall A \subseteq V\}$ and the base polytope $B(F) = \{\boldsymbol{s} \in \mathbb{R}^d : s(A) \leq F(A), \forall A \subset V, s(V) = F(V)\}$.

Both polyhedra $P(F)$ and $B(F)$ are solvable; i.e., we can efficiently optimize linear functions over them (despite having exponentially many constraints). In fact, this can be done in $O(d \log d)$ via a *greedy algorithm*.

Proposition 22 (Greedy algorithm [Edm03])**.** *Given a submodular function F, and its Lovász extension f_L, let $\boldsymbol{x} \in \mathbb{R}^d$ such that $x_{j_1} \geq \cdots \geq x_{j_d}$, we define $\hat{\boldsymbol{s}} \in \mathbb{R}^d$ such that $\hat{s}_{j_k} = F(\{j_1, \cdots, j_k\}) - F(\{j_1, \cdots, j_{k-1}\})$, then $\hat{\boldsymbol{s}} \in B(F)$ and is a maximizer in the following two problems:*

- $\forall \boldsymbol{x} \in \mathbb{R}^d_+, \max_{\boldsymbol{s} \in P(F)} \boldsymbol{x}^T \boldsymbol{s} = f_L(\boldsymbol{x})$,
- $\forall \boldsymbol{x} \in \mathbb{R}^d, \max_{\boldsymbol{s} \in B(F)} \boldsymbol{x}^T \boldsymbol{s} = f_L(\boldsymbol{x})$.

Proof. First let's show that $\hat{\boldsymbol{s}} \in B(F)$. For all $A \subseteq V$, we have:

$$\begin{aligned} \sum_{j_k \in A} \hat{s}_{j_k} &= \sum_{j_k \in S} F(\{j_1, \cdots, j_k\}) - F(\{j_1, \cdots, j_{k-1}\}) \\ &\leq \sum_{j_k \in S} F(A \cap \{j_1, \cdots, j_k\}) - F(A \cap \{j_1, \cdots, j_{k-1}\}) && \text{(by submodularity)} \\ &= \sum_{k=1}^{n} F(A \cap \{j_1, \cdots, j_k\}) - F(A \cap \{j_1, \cdots, j_{k-1}\}) \\ &= F(A) && \text{(by telescoping the sums)} \end{aligned}$$

It remains to show that $\hat{\boldsymbol{s}}$ is the optimal point in both $B(F)$ and $P(F)$ (when $\boldsymbol{x} \in \mathbb{R}_+^d$). For all $\forall \boldsymbol{s} \in P$, we have:

$$\begin{aligned}
\boldsymbol{x}^T(\hat{\boldsymbol{s}} - \boldsymbol{s}) &= \sum_{k=1}^{d} x_{j_k}(\hat{s}_{j_k} - s_{j_k}) \\
&= \sum_{k=1}^{d} x_{j_k}(F(\{j_1, \cdots, j_k\}) - F(\{j_1, \cdots, j_{k-1}\}) - s_{j_k}) \\
&= \sum_{k=1}^{d} x_{j_k}[(F(\{j_1, \cdots, j_k\}) - \boldsymbol{s}(\{j_1, \cdots, j_k\})) - (F(\{j_1, \cdots, j_{k-1}\}) - \boldsymbol{s}(\{j_1, \cdots, j_{k-1}\}))] \\
&= \sum_{k=1}^{d} x_{j_k}[F(\{j_1, \cdots, j_k\}) - \boldsymbol{s}(\{j_1, \cdots, j_k\})] - \sum_{k=1}^{d} x_{j_k}(F(\{j_1, \cdots, j_{k-1}\}) - \boldsymbol{s}(\{j_1, \cdots, j_{k-1}\})) \\
&= \sum_{k=1}^{n} x_{j_k}(F(\{j_1, \cdots, j_k\}) - \boldsymbol{s}(\{j_1, \cdots, j_k\})) - \sum_{k=0}^{d-1} x_{j_{k+1}}(F(\{j_1, \cdots, j_k\}) - \boldsymbol{s}(\{j_1, \cdots, j_k\})) \\
&= \sum_{k=1}^{d-1} (x_{j_k} - x_{j_{k+1}})(F(\{j_1, \cdots, j_k\}) - \boldsymbol{s}(\{j_1, \cdots, j_k\})) + x_d(F(V) - \boldsymbol{s}(V)) \\
&\geq 0
\end{aligned}$$

□

Submodular function minimization A variety of methods were developed to minimize general submodular functions, in polynomial time, either exactly or approximately up to some accuracy $F(A) - \min_{S \subseteq V} F(S) \leq \epsilon$. These methods include combinatorial algorithms such as [IFF01, Sch00], and convex algorithms which utilize the following connections between submodularity and convexity.

Proposition 23 ([Lov83])**.** *The Lovász extension f_L is convex iff F is submodular.*

Proposition 24 (Submodular function minimization, see, e.g., [Bac13, Proposition 3.7])**.** *Given a submodular function F and its Lovász extension f_L, we have the following equivalence:*

$$\min_{S \subseteq V} F(S) = \min_{\boldsymbol{s} \in [0,1]^d} f_L(\boldsymbol{s})$$

Moreover, if $S^\star$ is a minimizer of F, then $\mathbb{1}_{S^\star}$ is a minimizer of f_L, and if $\boldsymbol{s}^\star$ is a minimizer of f_L, then any set $\{i : s_i^\star \geq \theta\}$, obtained by thresholding $\boldsymbol{s}^\star$ with any $\theta \in (0, 1)$, is a minimizer of F.

Proposition 25 (Lemma 7.4 in [Fuj05])**.** *Consider the following quadratic program over the base polytope:*

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x} \in B(f)} \frac{1}{2}\|x\|_2^2$$

We define the sets $A_- = \{i \in V | x_i^* < 0\}$ and $A_+ = \{i \in V | x_i^* \leq 0\}$. Then, A_- is the unique minimal minimizer of F, and A_+ is the unique maximal minimizer of F.

For a review of submodular minimization algorithms see [Bac13, Chapter 10], and [CJK14, CLSW17] for more recent updates. The best known time-complexity for general submodular minimization problems is $\tilde{O}(d^{5/3} \cdot \mathrm{EO} \cdot \mathrm{M}/\epsilon^2)$ time, given by [CLSW17], where EO is the time to evaluate F on any set $S \subseteq V$ and $M = \max_{S \subseteq V} F(S)$.

A.2 Convex closure of set functions

In this section, we consider a general set function $F : 2^V \to \overline{\mathbb{R}}$, allowed to take infinite value, and which is not necessarily submodular. We review the concept of the *convex closure* of F, which characterizes its *tightest* convex extension on $[0,1]^d$. The results presented here are mostly based on the survey [Dug09] and the lecture notes [Von10], which we have adjusted to allow for infinite values.

Formally, the convex closure is defined as follows.

Definition 18 (Convex Closure, see, e.g., [Dug09, Def. 3.1])**.** *Given a set function $F : 2^V \to \overline{\mathbb{R}}$, the convex closure $f^- : [0,1]^d \to \overline{\mathbb{R}}$ is the point-wise largest convex function from $[0,1]^d$ to $\overline{\mathbb{R}}$ that always lower bounds F.*

The convex closure admit also two variational forms, which we present below.

Definition 19 (Equivalent definition of Convex Closure, see, e.g., [Von10, Def. 1] and [Dug09, Def. 3.2])**.** *Given a set function $F : 2^V \to \overline{\mathbb{R}}$, the convex closure of F can equivalently be defined $\forall \boldsymbol{x} \in [0,1]^n$ as:*

$$f^-(\boldsymbol{x}) = \inf\{\sum_{S \subseteq V} \alpha_S F(S) : \boldsymbol{x} = \sum_{S \subseteq V} \alpha_S \mathbb{1}_S, \sum_{S \subseteq V} \alpha_S = 1, \alpha_S \geq 0\}$$

It is easy to see that f^- is a convex extension of F from the above definition.

Proof. To see that the two definitions are equivalent, let f_1^- denote the function as defined in Def. 18 and f_2^- as defined in Def. 19. Note first that if $f_2^-(\boldsymbol{x}) = \infty$ then $f_1^-(\boldsymbol{x}) = \infty$, since $f_2^- \leq f_1^-$. Otherwise if $f_2^-(\boldsymbol{x}) \neq \infty$

$$\begin{aligned}
f_1^-(\boldsymbol{x}) &= f_1^-(\sum \alpha_S^\star(\boldsymbol{x}) \mathbb{1}_S) && \text{(where } \boldsymbol{\alpha}^\star \text{ is the minimizer in the Def. of } f_2^-\text{)} \\
&\leq \sum \alpha_S^\star(\boldsymbol{x}) f_1^-(\mathbb{1}_S) && \text{(by convexity of } f_1^-\text{)} \\
&= \sum \alpha_S^\star(\boldsymbol{x}) F(S) && (f_1^- \text{ is a lower bound on } F) \\
&= f_2^-(\boldsymbol{x})
\end{aligned}$$

On the other hand, since f_2^- is a convex extension of F and f_1^- is the largest convex lower bound on F, we must have $f_2^- \leq f_1^-$, and hence $f_2^- = f_1^-$. □

Definition 20 (Another equivalent definition of Convex Closure)**.** *Given a set function $F : 2^V \to \overline{\mathbb{R}}$, the convex closure of F can be also written $\forall \boldsymbol{x} \in \operatorname{dom} f^-$ as:*

$$f^-(\boldsymbol{x}) = \max_{\boldsymbol{\kappa}\in\mathbb{R}^d, \rho\in\mathbb{R}} \{\boldsymbol{\kappa}^T\boldsymbol{x} + \rho : \boldsymbol{\kappa}(S) + \rho \leq F(S), \forall S \subseteq V\}$$

Proof.

$$\begin{aligned}
f^-(\boldsymbol{x}) &= \min_{\boldsymbol{\alpha}} \{\sum_{S\subseteq V} \alpha_S F(S) : \boldsymbol{x} = \sum_{S\subseteq V} \alpha_S \mathbb{1}_S, \sum_{S\subseteq V} \alpha_S = 1, \alpha_S \geq 0\} \\
&= \min_{\boldsymbol{\alpha}\geq 0} \max_{\boldsymbol{\kappa},\rho} \{\sum_{S\subseteq V} \alpha_S F(S) + \boldsymbol{\kappa}^T(\boldsymbol{x} - \sum_{S\subseteq V} \alpha_S \mathbb{1}_S) + \rho(\sum_{S\subseteq V} \alpha_S - 1)\} \\
&= \max_{\boldsymbol{\kappa},\rho} \min_{\boldsymbol{\alpha}\geq 0} \{\boldsymbol{\kappa}^T\boldsymbol{x} + \sum_{S\subseteq V} \alpha_S (F(S) - \boldsymbol{\kappa}^T \mathbb{1}_S - \rho) + \rho\} && \text{(by Slater's condition)} \\
&= \max_{\boldsymbol{\kappa},\rho} \{\boldsymbol{\kappa}^T\boldsymbol{x} + \rho : \boldsymbol{\kappa}^T \mathbb{1}_S + \rho \leq F(S))\}
\end{aligned}$$

□

Next, we present some useful properties of the convex closure.

Proposition 26. *The convex closure f^- of a set function $F : 2^V \to \overline{\mathbb{R}}$ is lower semi-continuous.*

Proof. This follows from Definition 20 of $f^-(\boldsymbol{x})$. We present an elementary proof here.

It's enough to show that the sublevel sets of f^- are closed. First note that the domain of f^- is closed. Since $\operatorname{dom} f^- = \operatorname{conv}\{S : F(S) \neq \infty\}$ and the convex hull of a finite set is closed. Then given any $\alpha \in \mathbb{R}$, let $f(\boldsymbol{x}) > \alpha$, we will show that $\exists \epsilon > 0$, s.t., $\forall \boldsymbol{x}' \in B_\epsilon(\boldsymbol{x}), f(\boldsymbol{x}') > \alpha$. If $\boldsymbol{x} \notin \operatorname{dom} f$, this holds since $\operatorname{dom} f$ is a closed set. Otherwise, let $\boldsymbol{\kappa}^\star, \rho^\star$ be the maximizers in Definition 20 of $f^-(\boldsymbol{x})$. We choose $\epsilon < \frac{f^-(\boldsymbol{x})-\alpha}{\|\kappa^\star\|_2}$. Then $\forall \boldsymbol{x}' \in B_\epsilon(\boldsymbol{x})$ we have,

$$\begin{aligned}
f^-(\boldsymbol{x}') &\geq (\boldsymbol{\kappa}^\star)^T\boldsymbol{x}' + \rho^\star = (\boldsymbol{\kappa}^\star)^T(\boldsymbol{x}' - \boldsymbol{x}) + (\boldsymbol{\kappa}^\star)^T\boldsymbol{x} + \rho^\star \\
&\geq -\|\boldsymbol{\kappa}^\star\|_2 \|\boldsymbol{x}' - \boldsymbol{x}\|_2 + f^-(\boldsymbol{x}) \\
&> \alpha
\end{aligned}$$

Then the sublevel set $\{f(\boldsymbol{x}) \leq \alpha\}$ is closed. □

Proposition 27 (see, e.g., [Dug09, Prop. 3.23])**.** *The minimum values of a proper set function F and its convex closure f^- are equal, i.e.,*

$$\min_{\boldsymbol{s}\in[0,1]^d} f^-(\boldsymbol{s}) = \min_{S\subseteq V} F(S)$$

Moreover, if $S^\star$ is a minimizer of F, then $\mathbb{1}_{S^\star}$ is a minimizer of f^-, and if $\boldsymbol{s}^\star$ is a minimizer of f^-, then every set in the support of $\boldsymbol{\alpha}^\star$, where $\boldsymbol{\alpha}^\star$ is the corresponding minimizer in Definition

19 of $f^-(\boldsymbol{x}^\star)$, is a minimizer of F.

Proof. First note that, $\{0,1\}^d \subseteq [0,1]^d$ implies that $f^-(\boldsymbol{s}^\star) \leq F(S^\star)$. On the other hand, $f^-(\boldsymbol{s}^\star) = \sum_{S\subseteq V} \alpha_S^\star F(S) \geq \sum_{S\subseteq V} \alpha_S^\star F(S^\star) = F(S^\star)$. The rest of the proposition follows directly. □

Corollary 12. *The convex closure f^- of a set function F satisfies*

$$\max_{S\subseteq V} m(S) - F(S) = \max_{\boldsymbol{s}\in[0,1]^d} \boldsymbol{m}^T\boldsymbol{s} - f^-(\boldsymbol{s})$$

for any modular function $m : 2^V \to \mathbb{R}$ and the corresponding vector representation $\boldsymbol{m} \in \mathbb{R}^d$.

Proof. This follows directly from proposition 27 by noting that the convex closure of $m(S) - F(S)$ is given by $\boldsymbol{m}^T\boldsymbol{s} - f^-(\boldsymbol{s})$. □

Proposition 28 (see, e.g., [Von10, Lemma 4])**.** *The convex closure f^- and the Lovász extension f_L of a set function $F : 2^V \to \mathbb{R}$ are identical, i.e., $f^-(\boldsymbol{s}) = f_L(\boldsymbol{s}), \forall \boldsymbol{s} \in [0,1]^d$, iff F is a submodular function.*

Note that Proposition 24 can be seen then as a corollary of Proposition 27 and 28.

Bibliography

[Aho16] M. Ahookhosh. Accelerated first-order methods for large-scale convex minimization. *arXiv preprint arXiv:1604.08846*, 2016.

[AZG07] T. Amin, M. Zeytinoglu, and L. Guan. Application of laplacian mixture model to image and video retrieval. *Multimedia, IEEE Transactions on*, 9(7), 2007.

[AZO14] Z. Allen-Zhu and L. Orecchia. Linear coupling: An ultimate unification of gradient and mirror descent. *arXiv preprint arXiv:1407.1537*, 2014.

[Bac08] F. Bach. Consistency of the group lasso and multiple kernel learning. *Journal of Machine Learning Research*, 9(Jun):1179–1225, 2008.

[Bac10a] F. Bach. Structured sparsity-inducing norms through submodular functions. In *NIPS*, pages 118–126, 2010.

[Bac10b] F. Bach. Convex analysis and optimization with submodular functions: a tutorial. *arXiv preprint arXiv:1010.4207*, 2010.

[Bac11] F. Bach. Shaping level sets with submodular functions. In *Advances in Neural Information Processing Systems*, pages 10–18, 2011.

[Bac13] F. Bach. Learning with submodular functions: A convex optimization perspective. *Foundations and Trends® in Machine Learning*, 6(2-3):145–373, 2013.

[Bae09] M. Baes. Estimate sequence methods: extensions and approximations. *Institute for Operations Research, ETH, Zürich, Switzerland*, 2009.

[BBC+16] L. Baldassarre, N. Bhan, V. Cevher, A. Kyrillidis, and S. Satpathi. Group-sparse model selection: Hardness and relaxations. *IEEE Transactions on Information Theory*, 62(11):6508–6534, 2016.

[BBCK13] L. Baldassarre, N. Bhan, V. Cevher, and A. Kyrillidis. Group-sparse model selection: Hardness and relaxations. *arXiv preprint arXiv:1303.3207*, 2013.

[BCDH10] R. Baraniuk, V. Cevher, M. Duarte, and C. Hegde. Model-based compressive sensing. *Information Theory, IEEE Transactions on*, 56(4):1982–2001, 2010.

[BD86] J. P. Boyle and R. L. Dykstra. A method for finding projections onto the intersection of convex sets in hilbert spaces. In *Advances in order restricted statistical inference*, pages 28–47. Springer, 1986.

[BD09] T. Blumensath and M. Davies. Sampling theorems for signals from the union of finite-dimensional linear subspaces. *Information Theory, IEEE Transactions on*, 55(4):1872–1882, 2009.

[Ben09] Y. Bengio. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

[Ber82] D. P. Bertsekas. Projected newton methods for optimization problems with simple constraints. *SIAM Journal on control and Optimization*, 20(2):221–246, 1982.

[BFSS15] N. Buchbinder, M. Feldman, J. Seffi, and R. Schwartz. A tight linear time (1/2)-approximation for unconstrained submodular maximization. *SIAM Journal on Computing*, 44(5):1384–1402, 2015.

[BH18] A. Beck and N. Hallak. Optimization problems involving group sparsity terms. *Mathematical Programming*, Apr 2018.

[BJM+12] F. Bach, R. Jenatton, J. Mairal, G. Obozinski, et al. Optimization with sparsity-inducing penalties. *Foundations and Trends® in Machine Learning*, 4(1):1–106, 2012.

[BL10] J. Borwein and A. S. Lewis. *Convex analysis and nonlinear optimization: theory and examples*. Springer Science & Business Media, 2010.

[BLM09] K. Bredies, D. A. Lorenz, and P. Maass. A generalized conditional gradient method and its connection to an iterative shrinkage method. *Computational Optimization and Applications*, 42(2):173–193, 2009.

[BND12] S. Babacan, S. Nakajima, and M. Do. Bayesian group-sparse modeling and variational inference. *Submitted to IEEE Transactions on Signal Processing*, 2012.

[BS17] A. Beck and S. Shtern. Linearly convergent away-step conditional gradient for non-strongly convex functions. *Mathematical Programming*, 164(1-2):1–27, 2017.

[BT04] A. Beck and M. Teboulle. A conditional gradient method with linear rate of convergence for solving convex linear systems. *Mathematical Methods of Operations Research*, 59(2):235–247, 2004.

[BT09a] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[BT09b] A. Beck and M. Teboulle. Gradient-based algorithms with applications to signal recovery. *Convex optimization in signal processing and communications*, pages 42–88, 2009.

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

[BWB14] C. Boyer, P. Weiss, and J. Bigot. An algorithm for variable density sampling with block-constrained acquisition. *SIAM Journal on Imaging Sciences*, 7(2):1080–1107, 2014.

[CC68] M. D. Canon and C. D. Cullum. A tight upper bound on the rate of convergence of frank-wolfe algorithm. *SIAM Journal on Control*, 6(4):509–516, 1968.

[CDS98] S. Chen, D. Donoho, and M. Saunders. Atomic decomposition by basis pursuit. *SIAM journal on scientific computing*, 20(1):33–61, 1998.

[Cha00] C.-C. Chang. Libsvm: Introduction and benchmarks. *http://www. csie. ntn. edu. tw/˜ cjlin/libsvm*, 2000.

[CHDB09] V. Cevher, C. Hegde, M. Duarte, and R. Baraniuk. Sparse signal recovery using markov random fields. In *NIPS*, 2009.

[Cio90] I. Cioranescu. *Geometry of Banach spaces, duality mappings and nonlinear problems*, volume 62 of *Mathematics and its Applications*. Kluwer Academic Publishers Group, Dordrecht, 1990.

[CJK14] D. Chakrabarty, P. Jain, and P. Kothari. Provable submodular minimization via fujishige-wolfe's algorithm. *Adv. in Neu. Inf. Proc. Sys.(NIPS)*, 2014.

[Cla10] K. L. Clarkson. Coresets, sparse greedy approximation, and the frank-wolfe algorithm. *ACM Transactions on Algorithms (TALG)*, 6(4):63, 2010.

[CLS84] W. Cook, L. Lovász, and A. Schrijver. A polynomial-time test for total dual integrality in fixed dimension. In *Mathematical programming at Oberwolfach II*, pages 64–69. Springer, 1984.

[CLSW17] D. Chakrabarty, Y. T. Lee, A. Sidford, and S. C.-w. Wong. Subquadratic submodular function minimization. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, pages 1220–1231, New York, NY, USA, 2017. ACM.

[CP11] P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In *Fixed-point algorithms for inverse problems in science and engineering*, pages 185–212. Springer, 2011.

[CR02] S. F. Cotter and B. D. Rao. Sparse channel estimation via matching pursuit with application to equalization. *IEEE Transactions on Communications*, 50(3):374–377, 2002.

[CRPW12] V. Chandrasekaran, B. Recht, P. Parrilo, and A. Willsky. The convex geometry of linear inverse problems. *Foundations of Computational Mathematics*, 12:805–849, 2012.

[CT05] E. J. Candes and T. Tao. Decoding by linear programming. *IEEE transactions on information theory*, 51(12):4203–4215, 2005.

[CT13] C. Cartis and A. Thompson. An exact tree projection algorithm for wavelets. *arXiv preprint arXiv:1304.4570*, 2013.

[CW05] P. Combettes and V. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling & Simulation*, 4(4):1168–1200, 2005.

[CWB08] E. J. Candes, M. B. Wakin, and S. P. Boyd. Enhancing sparsity by reweighted minimization. *Journal of Fourier analysis and applications*, 14(5):877–905, 2008.

[DDFG10] I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk. Iteratively reweighted least squares minimization for sparse recovery. *Communications on Pure and Applied Mathematics*, 63(1):1–38, 2010.

[DDK12] A. Das, A. Dasgupta, and R. Kumar. Selecting diverse features via spectral regularization. In *NIPS*, pages 1592–1600, 2012.

[dGJ13] A. d'Aspremont, C. Guzmán, and M. Jaggi. An optimal affine invariant smooth minimization algorithm. *arXiv preprint arXiv:1301.0465*, 2013.

[DHCB09] M. F. Duarte, C. Hegde, V. Cevher, and R. G. Baraniuk. Recovery of compressible signals in unions of subspaces. In *Information Sciences and Systems, 2009. CISS 2009. 43rd Annual Conference on*, pages 175–180. IEEE, 2009.

[DJ95] D. L. Donoho and I. M. Johnstone. Adapting to unknown smoothness via wavelet shrinkage. *Journal of the american statistical association*, 90(432):1200–1224, 1995.

[DK11] A. Das and D. Kempe. Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. *arXiv preprint arXiv:1102.3975*, 2011.

[DNW13] M. A. Davenport, D. Needell, and M. B. Wakin. Signal space cosamp for sparse recovery with redundant dictionaries. *IEEE Transactions on Information Theory*, 59(10):6820–6829, 2013.

[Don06] D. Donoho. Compressed sensing. *Information Theory, IEEE Transactions on*, 52(4):1289–1306, 2006.

[DP05] C. Ding and H. Peng. Minimum redundancy feature selection from microarray gene expression data. *Journal of bioinformatics and computational biology*, 3(02):185–205, 2005.

[DSSSC08] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the l 1-ball for learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, pages 272–279. ACM, 2008.

[Dug09] S. Dughmi. Submodular functions: Extensions, distributions, and algorithms. a survey. *arXiv preprint arXiv:0912.0322*, 2009.

[DWB08] M. F. Duarte, M. B. Wakin, and R. G. Baraniuk. Wavelet-domain compressive signal reconstruction using a hidden markov tree model. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 5137–5140. IEEE, 2008.

[EB09] Y. C. Eldar and H. Bolcskei. Block-sparsity: Coherence and efficient recovery. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 2885–2888. IEEE, 2009.

[Edm03] J. Edmonds. Submodular functions, matroids, and certain polyhedra. In *Combinatorial Optimization—Eureka, You Shrink!*, pages 11–26. Springer, 2003.

[EHBC13] M. El Halabi, L. Baldassarre, and V. Cevher. To convexify or not? Regression with clustering penalties on graphs. In *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2013 IEEE 5th International Workshop on*, pages 21–24. IEEE, 2013.

[EHBC14] M. El Halabi, L. Baldassarre, and V. Cevher. Map estimation for bayesian mixture models with submodular priors. In *Machine Learning for Signal Processing (MLSP), 2014 IEEE International Workshop on*, pages 1–6. IEEE, 2014.

[EHBC18] M. El Halabi, F. Bach, and V. Cevher. Combinatorial penalties: Structure preserved by convex relaxations. *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics*, 2018.

[EHC15] M. El Halabi and V. Cevher. A totally unimodular view of structured sparsity. *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, pp. 223–231*, 2015.

[EHHV+17] M. El Halabi, Y.-P. Hsieh, B. Vu, Q. Nguyen, and V. Cevher. General proximal gradient method: A case for non-Euclidean norms. (EPFL-CONF-230391), 2017.

[EKDN16] E. R. Elenberg, R. Khanna, A. G. Dimakis, and S. Negahban. Restricted strong convexity implies weak submodularity. *arXiv preprint arXiv:1612.00804*, 2016.

[EM09] Y. Eldar and M. Mishali. Robust recovery of signals from a structured union of subspaces. *Information Theory, IEEE Transactions on*, 55(11):5302–5316, 2009.

[FBDN07] M. A. Figueiredo, J. M. Bioucas-Dias, and R. D. Nowak. Majorization–minimization algorithms for wavelet-based image restoration. *IEEE Transactions on Image processing*, 16(12):2980–2991, 2007.

[Fei98] U. Feige. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998.

[FF93] L. E. Frank and J. H. Friedman. A statistical view of some chemometrics regression tools. *Technometrics*, 35(2):109–135, 1993.

[FL01] J. Fan and R. Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360, 2001.

[Fuj05] S. Fujishige. *Submodular functions and optimization*, volume 58. Elsevier Science, 2005.

[FW56] M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics (NRL)*, 3(1-2):95–110, 1956.

[GB14] M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. http://cvxr.com/cvx, March 2014.

[GE13] R. Giryes and M. Elad. Iterative hard thresholding with near optimal projection for signal recovery. In *10th Intl. Conf. on Sampling Theory and Appl.(SAMPTA)*, 2013.

[GH13] D. Garber and E. Hazan. A linearly convergent conditional gradient algorithm with applications to online and stochastic optimization. *arXiv preprint arXiv:1301.4666*, 2013.

[GH15] D. Garber and E. Hazan. Faster rates for the frank-wolfe method over strongly-convex sets. In *International Conference on Machine Learning*, pages 541–549, 2015.

[GH16] D. Garber and E. Hazan. A linearly convergent variant of the conditional gradient algorithm under strong convexity, with applications to online and stochastic optimization. *SIAM Journal on Optimization*, 26(3):1493–1528, 2016.

[GK02] W. Gerstner and W. Kistler. *Spiking neuron models: Single neurons, populations, plasticity*. Cambridge university press, 2002.

[GM86] J. Guélat and P. Marcotte. Some comments on wolfe's ‚Äòaway step,Äô. *Mathematical Programming*, 35(1):110–119, 1986.

[GO10] P. Garrigues and B. Olshausen. Group sparse coding with a laplacian scale mixture prior. In *NIPS*, 2010.

[GP79] F. Giles and W. R. Pulleyblank. Total dual integrality and integer polyhedra. *Linear algebra and its applications*, 25:191–196, 1979.

[GR98] A. Goldberg and S. Rao. Beyond the flow decomposition barrier. *J. ACM*, 45(5):783–797, September 1998.

[GRC09] G. Gasso, A. Rakotomamonjy, and S. Canu. Recovering sparse signals with a certain family of nonconvex penalties and dc programming. *IEEE Transactions on Signal Processing*, 57(12):4686–4698, 2009.

[Gül92] O. Güler. New proximal point algorithms for convex minimization. *SIAM J. Optim.*, 2(4):649–664, 1992.

[Gur16] Gurobi Optimization, Inc. Gurobi optimizer reference manual, 2016.

[HDC09] C. Hegde, M. Duarte, and V. Cevher. Compressive sensing recovery of spike trains using a structured sparsity model. In *SPARS'09-Signal Processing with Adaptive Sparse Structured Representations*, 2009.

[HGT06] K. K. Herrity, A. C. Gilbert, and J. A. Tropp. Sparse approximation via iterative thresholding. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 3, pages III–III. IEEE, 2006.

[HIS15a] C. Hegde, P. Indyk, and L. Schmidt. Approximation algorithms for model-based compressive sensing. *IEEE Transactions on Information Theory*, 61(9):5129–5147, 2015.

[HIS15b] C. Hegde, P. Indyk, and L. Schmidt. A nearly-linear time framework for graph-structured sparsity. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 928–937, 2015.

[HJN15] Z. Harchaoui, A. Juditsky, and A. Nemirovski. Conditional gradient algorithms for norm-regularized smooth convex optimization. *Mathematical Programming*, 152(1-2):75–112, 2015.

[HSK17] H. Hassani, M. Soltanolkotabi, and A. Karbasi. Gradient methods for submodular maximization. In *Advances in Neural Information Processing Systems*, pages 5843–5853, 2017.

[HZ10] J. Huang and T. Zhang. The benefit of group sparsity. *The Annals of Statistics*, 38(4):1978–2004, 2010.

[HZM11] J. Huang, T. Zhang, and D. Metaxas. Learning with structured sparsity. *The Journal of Machine Learning Research*, 12:3371–3412, 2011.

[IFF01] S. Iwata, L. Fleischer, and S. Fujishige. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *Journal of the ACM (JACM)*, 48(4):761–777, 2001.

[IR+05] H. Ishwaran, J. S. Rao, et al. Spike and slab variable selection: frequentist and bayesian strategies. *The Annals of Statistics*, 33(2):730–773, 2005.

[JAB11] R. Jenatton, J.-Y. Audibert, and F. Bach. Structured variable selection with sparsity-inducing norms. *Journal of Machine Learning Research*, 12:2777–2824, 2011.

[Jag13] M. Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pages 427–435, 2013.

[JMOB11] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach. Proximal methods for hierarchical sparse coding. *Journal of Machine Learning Reasearch*, 12:2297–2334, 2011.

[JOB10] R. Jenatton, G. Obozinski, and F. Bach. Structured sparse principal component analysis. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 366–373, 2010.

[JOV09] L. Jacob, G. Obozinski, and J. Vert. Group lasso with overlap and graph lasso. In *International Conference on Machine Learning*, 2009.

[JRD16] P. Jain, N. Rao, and I. S. Dhillon. Structured sparse regression via greedy hard thresholding. In *Advances in Neural Information Processing Systems*, pages 1516–1524, 2016.

[JSK11] V. Jojic, S. Saria, and D. Koller. Convex envelopes of complexity controlling penalties: the case against premature envelopment. In *International Conference on Artificial Intelligence and Statistics*, pages 399–406, 2011.

[JXC08] S. Ji, Y. Xue, and L. Carin. Bayesian compressive sensing. *Signal Processing, IEEE Transactions on*, 56(6), 2008.

[KBEH+15] A. Kyrillidis, L. Baldassarre, M. El Halabi, Q. Tran-Dinh, and V. Cevher. Structured sparsity: Discrete and convex approaches. In *Compressed Sensing and its Applications*, pages 341–387. Springer, 2015.

[KF00] K. Knight and W. Fu. Asymptotics for lasso-type estimators. *Annals of statistics*, pages 1356–1378, 2000.

[KG12] A. Krause and C. E. Guestrin. Near-optimal nonmyopic value of information in graphical models. *arXiv preprint arXiv:1207.1394*, 2012.

[KLOS14] J. A. Kelner, Y. T. Lee, L. Orecchia, and A. Sidford. An almost-linear-time algorithm for approximate max flow in undirected graphs, and its multicommodity generalizations. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 217–226. SIAM, 2014.

[KX10] S. Kim and E. Xing. Tree-guided group lasso for multi-task regression with structured sparsity. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 543–550, 2010.

[KZ04] V. Kolmogorov and R. Zabin. What energy functions can be minimized via graph cuts? *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(2):147–159, 2004.

[Lan12] G. Lan. An optimal method for stochastic composite optimization. *Mathematical Programming*, 133(1-2):365–397, 2012.

[LDP07] M. Lustig, D. Donoho, and J. Pauly. Sparse mri: The application of compressed sensing for rapid mr imaging. *Magnetic resonance in medicine*, 58(6):1182–1195, 2007.

[Liu10] H. Liu. *Nonparametric Learning in High Dimensions*. Carnegie Mellon University, 2010.

[LJJ15] S. Lacoste-Julien and M. Jaggi. On the global linear convergence of frank-wolfe optimization variants. In *Advances in Neural Information Processing Systems*, pages 496–504, 2015.

[LLN06] B. Lehmann, D. Lehmann, and N. Nisan. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior*, 55(2):270–296, 2006.

[LMH15] H. Lin, J. Mairal, and Z. Harchaoui. A universal catalyst for first-order optimization. In *Advances in Neural Information Processing Systems*, pages 3384–3392, 2015.

[Lov83] L. Lovász. Submodular functions and convexity. In *Mathematical Programming The State of the Art*, pages 235–257. Springer, 1983.

[LST13] J. D. Lee, Y. Sun, and J. E. Taylor. On model selection consistency of penalized m-estimators: a geometric theory. In *Advances in Neural Information Processing Systems*, pages 342–350, 2013.

[LZ16] G. Lan and Y. Zhou. Conditional gradient sliding for convex optimization. *SIAM Journal on Optimization*, 26(2):1379–1409, 2016.

[Mar70] B. Martinet. Régularisation d'inéquations variationnelles par approximations successives. *Revue française d'informatique et de recherche opérationnelle. Série rouge*, 4(R3):154–158, 1970.

[Mil02] A. Miller. *Subset selection in regression*. CRC Press, 2002.

[MJBO10] J. Mairal, R. Jenatton, F. R. Bach, and G. R. Obozinski. Network flow algorithms for structured sparsity. In *Advances in Neural Information Processing Systems*, pages 1558–1566, 2010.

[MJOB11] J. Mairal, R. Jenatton, G. Obozinski, and F. Bach. Convex and network flow optimization for structured sparsity. *Journal of Machine Learning Research*, 12(Sep):2681–2720, 2011.

[MMP13] C. A. Micchelli, J. M. Morales, and M. Pontil. Regularizers for structured sparsity. *Advances in Computational Mathematics*, pages 1–35, 2013.

[Mor62] J.-J. Moreau. Fonctions convexes duales et points proximaux dans un espace hilbertien. *CR Acad. Sci. Paris Ser. A Math.*, 255:2897–2899, 1962.

[MRS+10] S. Mosci, L. Rosasco, M. Santoro, A. Verri, and S. Villa. Solving structured sparsity regularization with proximal methods. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 418–433. Springer, 2010.

[MS] N. Mitianoudis and T. Stathaki. Overcomplete source separation using laplacian mixture models. *IEEE Signal Processing Letters*, 12(4).

[MZ93] S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *Signal Processing, IEEE Transactions on*, 41(12):3397–3415, 1993.

[Nat95] B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM journal on computing*, 24(2):227–234, 1995.

[Nes04] Y. Nesterov. *Introductory lectures on convex optimization*, volume 87 of *Applied Optimization*. Kluwer Academic Publishers, Boston, MA, 2004. A basic course.

[Nes05] Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1):127–152, 2005.

[Nes07] Y. Nesterov. Dual extrapolation and its applications to solving variational inequalities and related problems. *Mathematical Programming*, 109(2-3):319–344, 2007.

[Nes13] Y. Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.

[Nes15] Y. Nesterov. Complexity bounds for primal-dual methods minimizing the model of objective function. *Mathematical Programming*, pages 1–20, 2015.

[NRW+12] S. N. Negahban, P. Ravikumar, M. J. Wainwright, B. Yu, et al. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. *Statistical Science*, 27(4):538–557, 2012.

[NRWY11] S. Negahban, P. Ravikumar, M. J. Wainwright, and B. Yu. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. In *Adv. Neural Inf. Proc. Sys.(NIPS)*, 2011.

[NT09] D. Needell and J. Tropp. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26(3):301–321, 2009.

[NW99] G. L. Nemhauser and L. A. Wolsey. *Integer and combinatorial optimization*, volume 18. Wiley New York, 1999.

[NWF78] G. Nemhauser, L. Wolsey, and M. Fisher. An analysis of approximations for maximizing submodular set functions — I. *Mathematical Programming*, 14(1):265–294, 1978.

[NYD83] A. Nemirovskii, D. B. Yudin, and E. R. Dawson. Problem complexity and method efficiency in optimization. 1983.

[OB12] G. Obozinski and F. Bach. Convex relaxation for combinatorial penalties. *arXiv preprint arXiv:1205.1240*, 2012.

[OB16] G. Obozinski and F. Bach. A unified perspective on convex structured sparsity: Hierarchical, symmetric, submodular norms and beyond. 2016.

[OJV11] G. Obozinski, L. Jacob, and J. Vert. Group lasso with overlaps: The latent group lasso approach. *arXiv preprint arXiv:1110.0413*, 2011.

[OTJ10] G. Obozinski, B. Taskar, and M. I. Jordan. Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing*, 20(2):231–252, 2010.

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

[PF11] G. Peyré and J. Fadili. Group sparsity with overlapping partition functions. In *Signal Processing Conference, 2011 19th European*, pages 303–307. IEEE, 2011.

[PVMH08] F. Parvaresh, H. Vikalo, S. Misra, and B. Hassibi. Recovering sparse signals using sparse measurement matrices in compressed dna microarrays. *IEEE Journal of Selected Topics in Signal Processing*, 2(3):275–285, 2008.

[RBV08] F. Rapaport, E. Barillot, and J. Vert. Classification of arraycgh data using fused svm. *Bioinformatics*, 24(13):i375–i382, 2008.

[RNWK11] N. S. Rao, R. D. Nowak, S. J. Wright, and N. G. Kingsbury. Convex approaches to model wavelet sparsity patterns. In *Image Processing (ICIP), 2011 18th IEEE International Conference on*, pages 1917–1920. IEEE, 2011.

[RRN12] N. Rao, B. Recht, and R. Nowak. Signal recovery in unions of subspaces with applications to compressive imaging. *arXiv preprint arXiv:1209.3079*, 2012.

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

[S+58] M. Sion et al. On general minimax theorems. *Pacific J. Math*, 8(1):171–176, 1958.

[SBB06] S. Sarvotham, D. Baron, and R. Baraniuk. Compressed sensing reconstruction via belief propagation. *preprint*, 2006.

[Sch00] A. Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory, Series B*, 80(2):346–355, 2000.

[SCJX17] C. Song, S. Cui, Y. Jiang, and S.-T. Xia. Accelerated stochastic greedy coordinate descent by soft thresholding projection onto simplex. In *Advances in Neural Information Processing Systems*, pages 4841–4850, 2017.

[See08] M. Seeger. Bayesian inference and optimal design for the sparse linear model. *The Journal of Machine Learning Research*, 9, 2008.

[SFHT13] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani. A sparse-group lasso. *Journal of Computational and Graphical Statistics*, 22(2):231–245, 2013.

[Sha49] C. E. Shannon. Communication in the presence of noise. *Proceedings of the IRE*, 37(1):10–21, 1949.

[She17] J. Sherman. Area-convexity, l &infty; regularization, and undirected multicommodity flow. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 452–460. ACM, 2017.

[SHI13] L. Schmidt, C. Hegde, and P. Indyk. The constrained earth mover distance model, with applications to compressive sensing. In *10th Intl. Conf. on Sampling Theory and Appl.(SAMPTA)*, 2013.

[SPH09] M. Stojnic, F. Parvaresh, and B. Hassibi. On the reconstruction of block-sparse signals with an optimal number of measurements. *Signal Processing, IEEE Transactions on*, 57(8):3075–3085, 2009.

[SRB11] M. Schmidt, N. Roux, and F. Bach. Convergence rates of inexact proximal-gradient methods for convex optimization. *NIPS, Granada, Spain*, 2011.

[SSS07] S. Shalev-Shwartz and Y. Singer. Online learning: Theory, algorithms, and applications. Technical report, Hebrew University, 2007.

[STM+05] A. Subramanian, P. Tamayo, V. Mootha, S. Mukherjee, B. Ebert, M. Gillette, A. Paulovich, S. Pomeroy, T. Golub, E. Lander, et al. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences of the United States of America*, 102(43):15545–15550, 2005.

[TG07] J. Tropp and A. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *Information Theory, IEEE Transactions on*, 53(12):4655–4666, 2007.

[Tib96] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[Tro04] J. A. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on Information theory*, 50(10):2231–2242, 2004.

[Tru90] K. Truemper. A decomposition theory for matroids. v. testing of matrix total unimodularity. *Journal of Combinatorial Theory, Series B*, 49(2):241–281, 1990.

[Tse08] P. Tseng. On accelerated proximal gradient methods for convex-concave optimization. submitted to siam j. *J. Optim*, 2008.

[TSR+05] 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 (Statistical Methodology)*, 67(1):91–108, 2005.

[VDGB09] S. A. Van De Geer and P. Bühlmann. On the conditions used to prove oracle results for the lasso. *Electronic Journal of Statistics*, 3:1360–1392, 2009.

[Von07] J. Vondrák. Submodularity in combinatorial optimization. 2007.

[Von10] J. Vondrák. Continuous extensions of submodular functions. CS 369P: Polyhedral techniques in combinatorial optimization, https://theory.stanford.edu/~jvondrak/CS369P/lec17.pdf, November 2010.

[VRMV14] S. Villa, L. Rosasco, S. Mosci, and A. Verri. Proximal methods for the latent group lasso penalty. *Computational Optimization and Applications*, 58(2):381–407, 2014.

[VSBV13] S. Villa, S. Salzo, L. Baldassarre, and A. Verri. Accelerated and inexact forward-backward algorithms. *SIAM Journal on Optimization*, 23(3):1607–1633, 2013.

[Wei05] S. Weisberg. *Applied linear regression*, volume 528. John Wiley & Sons, 2005.

[WNF09] S. Wright, R. Nowak, and M. Figueiredo. Sparse reconstruction by separable approximation. *Signal Processing, IEEE Transactions on*, 57(7):2479–2493, 2009.

[Wol70] P. Wolfe. Convergence theory in nonlinear programming. *Integer and nonlinear programming*, pages 1–36, 1970.

[WT13] M. Walter and K. Truemper. Implementation of a unimodularity test. *Mathematical Programming Computation*, 5(1):57–73, 2013.

[WWJ16] A. Wibisono, A. C. Wilson, and M. I. Jordan. A variational perspective on accelerated methods in optimization. *arXiv preprint arXiv:1603.04245*, 2016.

[WWW+16] W. Wen, C. Wu, Y. Wang, Y. Chen, and H. Li. Learning structured sparsity in deep neural networks. In *Advances in Neural Information Processing Systems*, pages 2074–2082, 2016.

[YB+17] X. Yan, J. Bien, et al. Hierarchical sparse modeling: A choice of two group lasso formulations. *Statistical Science*, 32(4):531–560, 2017.

[YL06] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006.

[YWW10] F. C. Yang, Z. Wei, and D. Wang. Subdifferential representation of homogeneous functions and extension of smoothness in banach spaces. *Acta Mathematica Sinica, English Series*, 26(8):1535–1544, 2010.

[YZS17] Y. Yu, X. Zhang, and D. Schuurmans. Generalized conditional gradient for sparse estimation. *Journal of Machine Learning Research*, 18(144):1–46, 2017.

[Zal02] C. Zalinescu. *Convex analysis in general vector spaces*. World scientific, 2002.

[ZJH10] Y. Zhou, R. Jin, and S. Hoi. Exclusive lasso for multi-task feature selection. In *International Conference on Artificial Intelligence and Statistics*, pages 988–995, 2010.

[ZL08] H. Zou and R. Li. One-step sparse estimates in nonconcave penalized likelihood models. *Annals of statistics*, 36(4):1509, 2008.

[Zou06] H. Zou. The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476):1418–1429, 2006.

[ZRY09] P. Zhao, G. Rocha, and B. Yu. The composite absolute penalties family for grouped and hierarchical variable selection. *The Annals of Statistics*, 37(6A):3468–3497, 2009.

Marwa El Halabi

Allée du Tilleul 3
1022 Chavannes-près-Renens
Switzerland
+41 78 930 85 74
marwa.elhalabi@epfl.ch
Google Scholar ID:
Vd6RW7cAAAAJ

Research Interests

Machine Learning, Discrete & Continuous Optimization, Submodularity, Algorithms.

Education

2012–present **PhD in Computer and Communication Sciences**, *École Polytechnique Fédérale de Lausanne (EPFL)*, Switzerland. Advisor: Prof. Volkan Cevher.
Thesis title: "Learning with Structured Sparsity: From Discrete to Convex and Back."
Date of private defense: May 18, 2018.

2008–2012 **B.Sc. in Computer and Communications Engineering, Minor in Mathematics**, *American University of Beirut (AUB)*, Lebanon.
Graduated with distinction, GPA: 3.98/4.

Research Experience

2012–present **Doctoral Assistant**, *Laboratory for Information and Inference Systems, LIONS*, EPFL. Advisor: Prof. Volkan Cevher.
Developed novel efficient methods for structured sparse recovery, convex and submodular optimization.

Jan - Apr, 2017 **Visiting Researcher**, *Machine learning research laboratory, SIERRA*, INRIA. Host: Prof. Francis Bach.
Characterized geometrically and statistically combinatorial structures preserved by convex continuous relaxations.

Nov, 2016 **Visiting Researcher**, *Learning & Adaptive Systems Group, LAS*, ETH. Host: Prof. Andreas Krause.
Investigated variational inference in structured sparsity models.

2011–2012 **Final Year Project**, *Electrical and Computer Engineering Dept.*, AUB. Advisor: Prof. Fadi Karameh.
Studied the effect of different electrode configurations on spatial targeting in electroconvulsive therapy (ECT), using a modeling and numerical solver software (OpenMEEG).

Jul–Aug, 2011 **Research Intern**, *Algorithmics Laboratory, ALGO*, EPFL. Advisor: Prof. Amin Shokrollahi.
Implemented in VHDL encoding/decoding methods for a parallelized construction of generalized Reed-Solomon codes to study speed-up compared to a regular construction.

2010-2011 **Research Assistant**, *Electrical and Computer Engineering Dept.*, AUB. Advisor: Prof. Wassim Masri.
Developed a multidimensional visualization tool, in Java, to visualize and analyze test cases based on their execution profiles and enable user-aided software fault localization.

Teaching and Supervision Experience

2012–2016 **Teaching Assistant**, *EPFL*.

- Mathematics of Data: From Theory to Computation, Fall'15, Fall'16 and Fall'17. Masters course ($\sim$ 60 Students).
- Advanced Topics in Data Sciences, Spring'16. PhD course (13 Students).
- Circuits and Systems I, Fall'12 and Fall'13. Undergraduate course ($\sim$150 Students).

May-Jul. 2015 **Student Project Co-Supervisor**, *EPFL*.
Siddhartha Satpathi, "Totally unimodular structure in phase retrieval of sparse signals and in nuclear magnetic resonance (NMR) spectroscopy", internship.

Awards and Honors

2015 Nominated for SPARS Best Student Paper award.

2013 Nominated for CAMSAP Best Student Paper award.

2013 CAMSAP travel grant, supported by the U.S. Army Research Office.

2008–2012 Placed on the Dean's Honor List at AUB (awarded for students ranked in the top 10% of their class).

Publications

Book Chapters

2015 Kyrillidis, A., Baldassarre, L., El Halabi, M., Tran-Dinh, Q. and Cevher, V. "Structured sparsity: Discrete and convex approaches". In *Compressed Sensing and its Applications*, pp. 341–387. Springer. Available at: `https://arxiv.org/abs/1507.05367`

Preprints

2017 El Halabi, M., Hsieh, Y.-P., Vu, B., Nguyen, Q. and Cevher, V. "General proximal gradient method: A case for non-Euclidean norms". Available at: `https://infoscience.epfl.ch/record/230391`.

Refereed Conference & Workshop Publications

2017 El Halabi, M., Bach. F. and Cevher, V. "Combinatorial penalties: Which structures are preserved by convex relaxations?". In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS)*. Available at: `https://arxiv.org/abs/1710.06273`.

2016 Norouzi, A., Bazzi, A., El Halabi, M., Bogunovic, I., Hsieh, Y.-P. and Cevher, V. "An efficient streaming algorithm for the submodular cover problem". In *Advances in Neural Information Processing Systems (NIPS)*. Available at: `https://arxiv.org/abs/1611.08574`.

2016 Odor, G., Li, Y.-H., Yurtsever, A., Hsieh, Y.-P., Tran-Dinh, Q., El Halabi, M. and Cevher, V. " Frank-Wolfe works for non-Lipschitz continuous gradient objectives: Scalable poisson phase retrieval". In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Available at: `https://arxiv.org/abs/1602.00724`.

2015 El Halabi, M. and Cevher, V. "A totally unimodular view of structured sparsity". In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*. Available at: `https://arxiv.org/abs/1411.1990`.**Extended abstract nominated for SPARS Best Student Paper award.**

2014 El Halabi, M., Baldassarre, L. and Cevher, V. "Map estimation for Bayesian mixture models with submodular priors". In *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*. Available at: `https://infoscience.epfl.ch/record/201800/files/PID3330271.pdf`.

2013 El Halabi, M., Baldassarre, L. and Cevher, V. "To convexify or not? Regression with clustering penalties on graphs". In *IEEE 5th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*. Available at: `https://infoscience.epfl.ch/record/189976`. **Nominated for Best Student Paper award.**

Skills

Programming C/C++, Matlab. Prior experience in: Python, Java, and VHDL.

Languages Arabic (native), English (fluent), French (fluent).

Invited and Conference Talks

- **Combinatorial penalties: Which structures are preserved by convex relaxations?**
  - International Conference on Artificial Intelligence and Statistics (AISTATS), Lanzarote, Canary Islands, April 2018 (Poster).

- **General proximal gradient method: A case for non-Euclidean norms.**
  - The Summer Research Institute (SuRI), Data Science track poster session, EPFL, Switzerland, June 2017 (Poster).
  - SIAM Conference on Optimization (OP17), Vancouver, Canada, May 2017 (Oral).

- **An efficient streaming algorithm for the submodular cover problem.**
  - Advances in Neural Information Processing Systems (NIPS), Barcelona, Spain, December 2016 (Poster).
  - EPFL-Google Research Day, EPFL, Switzerland, February 2018 (Poster).

- **A totally unimodular view of structured sparsity.**
  - Signal Processing with Adaptive Sparse Structured Representations (SPARS), Cambridge, UK, July 2015 (Oral).
  - International Conference on Artificial Intelligence and Statistics (AISTATS), San Diego, California, USA, May 2015 (Poster).
  - EPFL-Idiap-ETH Sparsity Workshop, EPFL, Switzerland, March 2015 (Oral).

- **Map estimation for Bayesian mixture models with submodular priors.**
  - IEEE International Workshop on Machine Learning for Signal Processing (MLSP), Reims, France, September 2014 (Oral).

- **To convexify or not? Regression with clustering penalties on graphs.**
  - IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing

(CAMSAP), Saint Martin, French West Indies, December 2013 (Oral & poster).

Professional Service

Journal & Conference Reviews:

- International Conference on Machine Learning (ICML), 2018.
- Advances in Neural Information Processing Systems (NIPS), 2017.
- IEEE Transactions on Signal Processing (TSP), 2016.
- IEEE International Symposium on Information Theory (ISIT), 2014.

Conference Volunteer:

- Advances in Neural Information Processing Systems (NIPS), 2017.
- IEEE 5th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2013.