Learning with Submodular Functions:
A Convex Optimization Perspective

Francis Bach[1]

[1] INRIA - SIERRA Project-Team, Laboratoire d'Informatique de l'Ecole Normale Supérieure, 23, avenue d'Italie, Paris, 75013, France, francis.bach@inria.fr

Abstract

Submodular functions are relevant to machine learning for mainly two reasons: (1) some problems may be expressed directly as the optimization of submodular functions and (2) the Lovász extension of submodular functions provides a useful set of regularization functions for supervised and unsupervised learning. In this paper, we present the theory of submodular functions from a convex analysis perspective, presenting tight links between certain polyhedra, combinatorial optimization and convex optimization problems. In particular, we show how submodular function minimization is equivalent to solving a wide variety of convex optimization problems. This allows the derivation of new efficient algorithms for approximate submodular function minimization with theoretical guarantees and good practical performance. By listing many examples of submodular functions, we review various applications to machine learning, such as clustering or subset selection, as well as a family of structured sparsity-inducing norms that can be derived and used from submodular functions.

Contents

i

Introduction

Many combinatorial optimization problems may be cast as the minimization of a *set-function*, that is a function defined on the set of subsets of a given base set V. Equivalently, they may be defined as functions on the vertices of the hyper-cube, i.e, $\{0,1\}^p$ where p is the cardinality of the base set V—they are then often referred to as pseudo-boolean functions [15]. Among these set-functions, submodular functions play an important role, similar to convex functions on vector spaces, as many functions that occur in practical problems turn out to be submodular functions or slight modifications thereof, with applications in many areas areas of computer science and applied mathematics, such as machine learning [86, 105, 80, 85], computer vision [18, 62], operations research [63, 118] or electrical networks [110]. Since submodular functions may be minimized exactly, and maximized approximately with some guarantees, in polynomial time, they readily lead to efficient algorithms for all the numerous problems they apply to.

However, the interest for submodular functions is not limited to discrete optimization problems. Indeed, the rich structure of submodular functions and their link with convex analysis through the Lovász extension [92] and the various associated polytopes makes them particularly

adapted to problems beyond combinatorial optimization, namely as regularizers in signal processing and machine learning problems [21, 6]. Indeed, many continuous optimization problems exhibit an underlying discrete structure, and submodular functions provide an efficient and versatile tool to capture such combinatorial structures.

In this paper, the theory of submodular functions is presented, in a self-contained way, with all results proved from first principles of convex analysis common in machine learning, rather than relying on combinatorial optimization and traditional theoretical computer science concepts such as matroids. A good knowledge of convex analysis is assumed (see, e.g., [17, 16]) and a short review of important concepts is presented in Appendix A.

Paper outline.   The paper is organized in several sections, which are summarized below:

(1) **Definitions**: In Section 1, we give the different definitions of submodular functions and of the associated polyhedra.

(2) **Lovász extension**: In Section 2, we define the Lovász extension and give its main properties. In particular we present the key result in submodular analysis, namely, the link between the Lovász extension and the submodular polyhedra through the so-called "greedy algorithm". We also present the link between sparsity-inducing norms and the Lovász extensions of non-decreasing submodular functions.

(3) **Examples**: In Section 3, we present classical examples of submodular functions, together with the main applications in machine learning.

(4) **Polyhedra**: Associated polyhedra are further studied in Section 4, where support functions and the associated maximizers are computed. We also detail the facial structure of such polyhedra, and show how it relates to the sparsity-inducing properties of the Lovász extension.

(5) **Separable optimization - Analysis**: In Section 5, we consider separable optimization problems regularized by the Lovász extension, and show how this is equivalent to a se-

quence of submodular function minimization problems. This is the key theoretical link between combinatorial and convex optimization problems related to submodular functions.

(6) **Separable optimization - Algorithms**: In Section 6, we present two sets of algorithms for separable optimization problems. The first algorithm is a an exact algorithm which relies on the availability of a submodular function minimization algorithm, while the second set of algorithms are based on existing iterative algorithms for convex optimization, some of which come with online and offline theoretical guarantees.

(7) **Submodular function minimization**: In Section 7, we present various approaches to submodular function minimization. We present briefly the combinatorial algorithms for exact submodular function minimization, and focus in more depth on the use of specific *convex* separable optimization problems, which can be solved iteratively to obtain approximate solutions for submodular function minimization, with theoretical guarantees and approximate optimality certificates.

(8) **Submodular optimization problems**: in Section 8, we present other combinatorial optimization problems which can be partially solved using submodular analysis, such as submodular function maximization and the optimization of differences of submodular functions, and relate these to non-convex optimization problems on the submodular polyhedra.

(9) **Experiments**: in Section 9, we provide illustrations of the optimization algorithms described earlier, for submodular function minimization, as well as for convex optimization problems (separable or not). The Matlab code for all these experiments may be found at `http://www.di.ens.fr/~fbach/submodular/`.

In Appendix A, we review relevant notions from convex analysis and convex optimization, while in Appendix B, we present several results related to submodular functions, such as operations that preserve

submodularity.

Several books and paper articles already exist on the same topic and the material presented in this paper rely mostly on those [49, 110, 133, 87]. However, in order to present the material in the simplest way, ideas from related research papers have also been used.

Notation.   We consider the set $V = \{1, \dots, p\}$, and its power set 2^V, composed of the 2^p subsets of V. Given a vector $s \in \mathbb{R}^p$, s also denotes the modular set-function defined as $s(A) = \sum_{k \in A} s_k$. Moreover, $A \subset B$ means that A is a subset of B, potentially equal to B. For $q \in [1, +\infty]$, we denote by $\|w\|_q$ the ℓ_q-norm of w, by $|A|$ the cardinality of the set A, and, for $A \subset V = \{1, \dots, p\}$, 1_A denotes the indicator vector of the set A. If $w \in \mathbb{R}^p$, and $\alpha \in \mathbb{R}$, then $\{w \geqslant \alpha\}$ (resp. $\{w > \alpha\}$) denotes the subset of $V = \{1, \dots, p\}$ defined as $\{k \in V, \ w_k \geqslant \alpha\}$ (resp. $\{k \in V, \ w_k > \alpha\}$), which we refer to as the weak (resp. strong) α-sup-level sets of w. Similarly if $v \in \mathbb{R}^p$, we denote $\{w \geqslant v\} = \{k \in V, \ w_k \geqslant v_k\}$.

1

Definitions

Throughout this paper, we consider $V = \{1, \ldots, p\}$, $p > 0$ and its power set (i.e., set of all subsets) 2^V, which is of cardinality 2^p. We also consider a real-valued set-function $F : 2^V \to \mathbb{R}$ such that $F(\varnothing) = 0$. As opposed to the common convention with convex functions (see Appendix A), we do not allow infinite values for the function F.

1.1 Equivalent definitions of submodularity

Submodular functions may be defined through several equivalent properties, which we now present.

Definition 1.1 (Submodular function). A set-function $F : 2^V \to \mathbb{R}$ is submodular if and only if, for all subsets $A, B \subset V$, we have: $F(A) + F(B) \geqslant F(A \cup B) + F(A \cap B)$.

The simplest example of submodular function is the cardinality (i.e., $F(A) = |A|$ where $|A|$ is the number of elements of A), which is both submodular and supermodular (i.e., its opposite is submodular), which we refer to as *modular*.

From Def. 1.1, it is clear that the set of submodular functions is closed under linear combination and multiplication by a positive scalar. Checking the condition in Def. 1.1 is not always easy in practice; it turns out that it can be restricted to only certain sets A and B, which we now present.

The following proposition shows that a submodular has the "diminishing return" property, and that this is sufficient to be submodular. Thus, submodular functions may be seen as a discrete analog to *concave* functions. However, as shown in Section 2, in terms of optimization they behave more like *convex* functions (e.g., efficient minimization, duality theory, links with convex Lovász extension).

Proposition 1.1. (Definition with first order differences) The set-function F is submodular if and only if for all $A, B \subset V$ and $k \in V$, such that $A \subset B$ and $k \notin B$, we have $F(A \cup \{k\}) - F(A) \geqslant F(B \cup \{k\}) - F(B)$.

Proof. Let $A \subset B$, and $k \notin B$, $F(A \cup \{k\}) - F(A) - F(B \cup \{k\}) + F(B) = F(C) + F(D) - F(C \cup D) - F(C \cap D)$ with $C = A \cup \{k\}$ and $D = B$, which shows that the condition is necessary. To prove the opposite, we assume that the condition is satisfied; one can first show that if $A \subset B$ and $C \cap B = \varnothing$, then $F(A \cup C) - F(A) \geqslant F(B \cup C) - F(B)$ (this can be obtained by summing the m inequalities $F(A \cup \{c_1, \ldots, c_k\}) - F(A \cup \{c_1, \ldots, c_{k-1}\}) \geqslant F(B \cup \{c_1, \ldots, c_k\}) - F(B \cup \{c_1, \ldots, c_{k-1}\})$ where $C = \{c_1, \ldots, c_m\}$).

Then, for any $X, Y \subset V$, take $A = X \cap Y$, $C = X \backslash Y$ and $B = Y$ (which implies $A \cup C = X$ and $B \cup C = X \cup Y$) to obtain $F(X) + F(Y) \geqslant F(X \cup Y) + F(X \cap Y)$, which shows that the condition is sufficient. $\square$

The following proposition gives the tightest condition for submodularity (easiest to show in practice).

Proposition 1.2. (Definition with second order differences) The set-function F is submodular if and only if for all $A \subset V$ and $j, k \in V \backslash A$, we have $F(A \cup \{k\}) - F(A) \geqslant F(A \cup \{j, k\}) - F(A \cup \{j\})$.

Proof. This condition is weaker than the one from previous proposition (as it corresponds to taking $B = A \cup \{j\}$). To prove that it is still sufficient, simply apply it to subsets $A \cup \{b_1, \ldots, b_{s-1}\}$, $j = b_s$ for $B = A \cup \{b_1, \ldots, b_m\} \supset A$ with $k \notin B$, and sum the m inequalities $F(A\cup\{b_1,\ldots,b_{s-1}\}\cup\{k\})-F(A\cup\{b_1,\ldots,b_{s-1}\}) \geqslant F(A\cup\{b_1,\ldots,b_s\}\cup \{k\}) - F(A \cup \{b_1, \ldots, b_s\})$, to obtain the condition in Prop. 1.1. □

In order to show that a given set-function is submodular, there are several possibilities: (a) using Prop. 1.2 directly, (b) use the Lovász extension (see Section 2) and show that it is convex, (c) cast it as a special case from Section 3 (typically a cut or a flow), or (d) use known operations on submodular functions presented in Appendix B.2.

1.2 Associated polyhedra

A vector $s \in \mathbb{R}^p$ naturally leads to a modular set-function defined as $s(A) = \sum_{k \in A} s_k = s^\top 1_A$, where $1_A \in \mathbb{R}^p$ is the indicator vector of the set A. We now define specific polyhedra in $\mathbb{R}^p$. These play a crucial role in submodular analysis, as most results may be interpreted or proved using such polyhedra.

Definition 1.2 (Submodular and base polyhedra). Let F be a submodular function such that $F(\varnothing) = 0$. The submodular polyhedron $P(F)$ and the base polyhedron $B(F)$ are defined as:

$$\begin{aligned} P(F) &= \{s \in \mathbb{R}^p, \ \forall A \subset V, s(A) \leqslant F(A)\} \\ B(F) &= \{s \in \mathbb{R}^p, \ s(V) = F(V), \ \forall A \subset V, s(A) \leqslant F(A)\} \\ &= P(F) \cap \{s(V) = F(V)\}. \end{aligned}$$

As shown in the following proposition, the submodular polyhedron $P(F)$ has non-empty interior and is unbounded. Note that the other polyhedron (the base polyhedron) will be shown to be non-empty and bounded as a consequence of Prop. 2.2. It has empty interior since it is included in the subspace $s(V) = F(V)$. See Figure 1.1 for examples with $p = 2$ and $p = 3$.
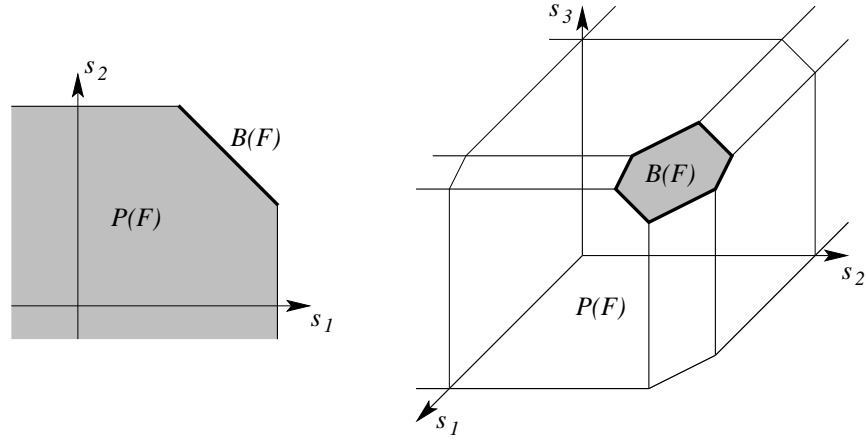
Fig. 1.1: Submodular polyhedron $P(F)$ and base polyhedron $B(F)$ for $p = 2$ (left) and $p = 3$ (right), for a non-decreasing submodular function (for which $B(F) \subset \mathbb{R}^p_+$, see Prop. 1.4).

Proposition 1.3. (Properties of submodular polyhedron) Let F be a submodular function such that $F(\varnothing) = 0$. If $s \in P(F)$, then for all $t \in \mathbb{R}^p$, such that $t \leqslant s$, we have $t \in P(F)$. Moreover, $P(F)$ has non-empty interior.

Proof. The first part is trivial, since $t \leqslant s$ implies that for all $A \subset V$, $t(A) \leqslant s(A)$. For the second part, we only need to show that $P(F)$ is non-empty, which is true since the constant vector equal to $\min_{A \subset V, A \neq \varnothing} \frac{F(A)}{|A|}$ belongs to $P(F)$. $\square$

1.3 Polymatroids (non-increasing submodular functions)

When the submodular function F is also *non-decreasing*, i.e., when for $A, B \subset V$, $A \subset B \Rightarrow F(A) \leqslant F(B)$, then the function is often referred to as a *polymatroid rank function* (see related matroid rank functions in Section 3.8). For theses functions, the base polyhedron is included in the positive orthant, and this is in fact a characteristic property.

Proposition 1.4. (Base polyhedron and polymatroids) Let F be a submodular function such that $F(\varnothing) = 0$. The function F is non-decreasing, if and only if the base polyhedron is included in the positive orthant $\mathbb{R}^p_+$.

Proof. The simplest proof uses the greedy algorithm from Section 2.2. We have from Prop. 2.2, $\min_{s \in B(F)} s_k = -\max_{s \in B(F)} (-1_{\{k\}})^\top s = -f(-1_{\{k\}}) = F(V) - F(V \backslash \{k\})$. Thus, $B(F) \subset \mathbb{R}^p_+$ if and only if for all $k \in V$, $F(V) - F(V \backslash \{k\}) \geqslant 0$. Since, by submodularity, for all $A \subset V$ and $k \notin A$, $F(A \cup \{k\}) - F(A) \geqslant F(V) - F(V \backslash \{k\})$, $B(F) \subset \mathbb{R}^p_+$ if and only if F is non-decreasing. $\qquad\square$

For polymatroids, another polyhedron is often considered, the symmetric independence polyhedron, which we now define. This polyhedron will turn out to be the unit ball of the dual norm of the norm defined in Section 2.3 (see more details and figures in Section 2.3).

Definition 1.3 (Symmetric independence polyhedron). Let F be a non-decreasing submodular function such that $F(\varnothing) = 0$. The submodular polyhedron $|P|(F)$ is defined as:

$$|P|(F) \quad = \quad \{ s \in \mathbb{R}^p, \ \forall A \subset V, |s|(A) \leqslant F(A) \} = \{ s \in \mathbb{R}^p, \ |s| \in P(F) \}$$

2

Lovász extension

We first consider a set-function F such that $F(\varnothing) = 0$, *which is not necessary submodular.* We can define its Lovász extension [92], which is often referred to as its Choquet integral [26]. The Lovász extension allows to draw links between submodular set-functions and regular convex functions, and transfer known results from convex analysis, such as duality. In particular, we prove in this section, the two key results of submodular analysis, namely that (a) a set-function is submodular if and only if its Lovász extension is convex, and (b) that the Lovász extension is the support function of the base polyhedron, with a direct relationship through the "greedy algorithm". We then present in Section 2.3 how for non-decreasing submodular functions, the Lovász extension may be used to define a structured sparsity-inducing norm.

2.1 Definition

We now define the Lovász extension of any set-function (not necessary submodular).

Definition 2.1 (Lovász extension). Given a set-function F such that $F(\varnothing) = 0$, the Lovász extension $f : \mathbb{R}^p \to \mathbb{R}$ is defined as follows;

for $w \in \mathbb{R}^p$, order the components in decreasing order $w_{j_1} \geqslant \cdots \geqslant w_{j_p}$, and define $f(w)$ through any of the following equivalent equations:

$$f(w) = \sum_{k=1}^{p} w_{j_k} \left[F(\{j_1, \ldots, j_k\}) - F(\{j_1, \ldots, j_{k-1}\}) \right], \qquad (2.1)$$

$$f(w) = \sum_{k=1}^{p-1} F(\{j_1, \ldots, j_k\})(w_{j_k} - w_{j_{k+1}}) + F(V)w_{j_p}, \qquad (2.2)$$

$$f(w) = \int_{\min\{w_1,\ldots,w_p\}}^{+\infty} F(\{w \geqslant z\})dz + F(V)\min\{w_1,\ldots,w_p\}, \quad (2.3)$$

$$f(w) = \int_0^{+\infty} F(\{w \geqslant z\})dz + \int_{-\infty}^0 [F(\{w \geqslant z\}) - F(V)]dz. \quad (2.4)$$

Proof. To prove that we actually define a function, one needs to prove that the definitions are independent of the potentially non unique ordering $w_{j_1} \geqslant \cdots \geqslant w_{j_p}$, which is trivial from the last formulation in Eq. (2.4). The first and second formulations in Eq. (2.1) and Eq. (2.2) are equivalent (by integration by parts, or Abel summation formula). To show equivalence with Eq. (2.3), one may notice that $z \mapsto F(\{w \geqslant z\})$ is piecewise constant, with value zero for $z > w_{j_1} = \max\{w_1, \ldots, w_p\}$, and equal to $F(\{j_1, \ldots, j_k\})$ for $z \in (w_{j_{k+1}}, w_{j_k})$, $k = \{1, \ldots, p-1\}$, and equal to $F(V)$ for $z < w_{j_p} = \min\{w_1, \ldots, w_p\}$. What happens at break points is irrelevant for integration.

To prove Eq. (2.4) from Eq. (2.3), notice that for $\alpha \leqslant \min\{0, w_1, \ldots, w_p\}$, Eq. (2.3) leads to

$$
\begin{aligned}
f(w) &= \int_{\alpha}^{+\infty} F(\{w \geqslant z\})dz - \int_{\alpha}^{\min\{w_1,\ldots,w_p\}} F(\{w \geqslant z\})dz \\
&\qquad\qquad\qquad\qquad\qquad\qquad + F(V)\min\{w_1,\ldots,w_p\} \\
&= \int_{\alpha}^{+\infty} F(\{w \geqslant z\})dz - \int_{\alpha}^{\min\{w_1,\ldots,w_p\}} F(V)dz \\
&\qquad\qquad\qquad\qquad\qquad\qquad + \int_0^{\min\{w_1,\ldots,w_p\}} F(V)dz \\
&= \int_{\alpha}^{+\infty} F(\{w \geqslant z\})dz - \int_{\alpha}^0 F(V)dz,
\end{aligned}
$$

and we get the result by letting α tend to $-\infty$. $\qquad\square$

Note that for modular functions $A \mapsto s(A)$, with $s \in \mathbb{R}^p$, then the Lovász extension is the linear function $w \mapsto w^\top s$. Moreover, for $p = 2$, we have

$$
\begin{aligned}
f(w) &= \frac{1}{2}[F(\{1\}) + F(\{2\}) - F(\{1,2\})] \cdot |w_1 - w_2| \\
&\quad + \frac{1}{2}[F(\{1\}) - F(\{2\}) + F(\{1,2\})] \cdot w_1 \\
&\quad + \frac{1}{2}[-F(\{1\}) + F(\{2\}) + F(\{1,2\})] \cdot w_2 \\
&= -[F(\{1\}) + F(\{2\}) - F(\{1,2\})] \min\{w_1, w_2\} \\
&\quad + F(\{1\})w_1 + F(\{2\})w_2,
\end{aligned}
$$

which allows an illustration of various propositions in this section (in particular Prop. 2.1).

The following proposition details classical properties of the Choquet integral/Lovász extension. In particular, property (e) below implies that the Lovász extension is equal to the original set-function on $\{0, 1\}^p$ (which can canonically be identified to 2^V), and hence is indeed an *extension* of F. See an illustration in Figure 2.1 for $p = 2$.

Proposition 2.1. (Properties of Lovász extension) Let F be any set-function such that $F(\varnothing) = 0$. We have:

(a) if F and G are set-functions with Lovász extensions f and g, then $f + g$ is the Lovász extension of $F + G$, and for all $\lambda \in \mathbb{R}$, λf is the Lovász extension of λF,

(b) for $w \in \mathbb{R}_+^p$, $f(w) = \int_0^{+\infty} F(\{w \geqslant z\})dz$,

(c) if $F(V) = 0$, for all $w \in \mathbb{R}^p$, $f(w) = \int_{-\infty}^{+\infty} F(\{w \geqslant z\})dz$,

(d) for all $w \in \mathbb{R}^p$ and $\alpha \in \mathbb{R}$, $f(w + \alpha 1_V) = f(w) + \alpha F(V)$,

(e) the Lovász extension f is positively homogeneous,

(f) for all $A \subset V$, $F(A) = f(1_A)$,

(g) if F is symmetric (i.e., $\forall A \subset V,\ F(A) = F(V \backslash A)$), then f is even,

(h) if $V = A_1 \cup \cdots \cup A_m$ is a partition of V, and $w = \sum_{i=1}^m v_i 1_{A_i}$ (i.e., w is constant on each set A_i), with $v_1 \geqslant \cdots \geqslant v_m$, then $f(w) = \sum_{i=1}^{m-1}(v_i - v_{i+1})F(A_1 \cup \cdots \cup A_i) + v_{m+1}F(V)$.

Proof. Properties (a), (b) and (c) are immediate from Eq. (2.4) and Eq. (2.2). Properties (d), (e) and (f) are straightforward from Eq. (2.2).
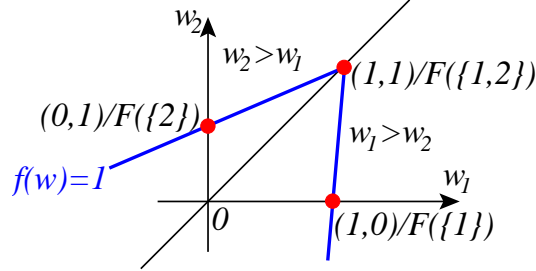
Fig. 2.1: Lovász extension for $V = \{1, 2\}$: the function is piecewise affine, with different slopes for $w_1 \geqslant w_2$, with values $F(\{1\})w_1 + [F(\{1,2\}) - F(\{1\})]w_2$, and for $w_1 \leqslant w_2$, with values $F(\{2\})w_2 + [F(\{1,2\}) - F(\{2\})]w_1$. The level set $\{w \in \mathbb{R}^2, f(w) = 1\}$ is displayed in blue, together with points of the form $\frac{1}{F(A)}1_A$.

If F is symmetric, then $F(V) = F(\varnothing) = 0$, and thus $f(-w) = \int_{-\infty}^{+\infty} F(\{-w \geqslant z\})dz = \int_{-\infty}^{+\infty} F(\{w \leqslant -z\})dz = \int_{-\infty}^{+\infty} F(\{w \leqslant z\})dz = \int_{-\infty}^{+\infty} F(\{w > z\})dz = f(w)$ (because we may replace strict inequalities by regular inequalities), i.e., f is even. Finally, property (h) is a direct consequence of Eq. (2.3). $\qquad\square$

Note that when the function is a cut function (see Section 3.2), then the Lovász extension is related to the total variation and property (c) is often referred to as the co-area formula (see [21] and references therein, as well as Section 3.2).

Decomposition into modular plus non-negative function. Given any submodular function G and an element t of the base polyhedron $B(G)$ defined in Def. 1.2, then the function $F = G - t$ is also submodular, and is such that F is always non-negative and $F(V) = 0$. Thus G may be (non uniquely) decomposed as the sum of a modular function t and a submodular function F which is always non-negative and such that $F(V) = 0$. Such functions F have interesting Lovász extensions. Indeed, for all $w \in \mathbb{R}^p$, $f(w) \geqslant 0$ and $f(w + \alpha 1_V) = f(w)$. Thus in order to represent the level set $\{f(w) = 1\}$, we only need to project onto a subspace orthogonal to 1_V. In Figure 2.2, we consider a
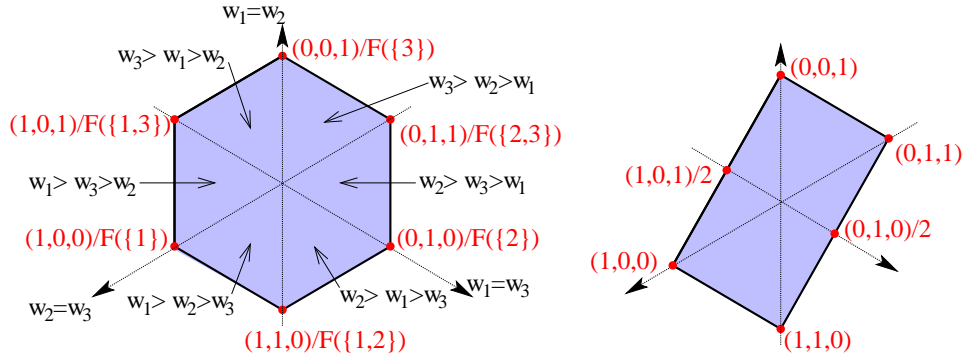
Fig. 2.2: Top: Polyhedral level set of f (projected on the set $w^\top 1_V = 0$), for 2 different submodular symmetric functions of three variables, with different inseparable sets leading to different sets of extreme points; changing values of F may make some of the extreme points disappear (see Section 4.2 for a discussion of inseparable sets and faces of this polytope). The various extreme points cut the space into polygons where the ordering of the components is fixed. Left: $F(A) = 1_{|A| \in \{1,2\}}$, leading to $f(w) = \max_{k \in \{1,2,3\}} w_k - \min_{k \in \{1,2,3\}} w_k$ (all possible extreme points); note that the polygon need not be symmetric in general. Right: one-dimensional total variation on three nodes, i.e., $F(A) = |1_{1 \in A} - 1_{2 \in A}| + |1_{2 \in A} - 1_{3 \in A}|$, leading to $f(w) = |w_1 - w_2| + |w_2 - w_3|$, for which the extreme points corresponding to the separable set $\{1, 3\}$ and its complement disappear.

function F which is symmetric (which implies that $F(V) = 0$ and F is non-negative, see more details in Section 7.4).

2.2   Greedy algorithm

The next result relates the Lovász extension with the support function[1] of the submodular polyhedron $P(F)$ which is defined in Def. 1.2. This is the basis for many of the theoretical results and algorithms related to submodular functions. It shows that maximizing a linear function with non-negative coefficients on the submodular polyhedron may be obtained in closed form, by the so-called "greedy algorithm" (see [92, 42]

---

[1] The support function is obtained by maximizing linear functions; see definition in Appendix A.

and Section 3.8 for an intuitive explanation of this denomination in the context of matroids), and the optimal value is equal to the value $f(w)$ of the Lovász extension. Note that otherwise, solving a linear programming problem with $2^p - 1$ constraints would then be required. This applies to the submodular polyhedron $P(F)$ and to the base polyhedron $B(F)$; note the different assumption regarding the positivity of the components of w.

Proposition 2.2. (Greedy algorithm for submodular and base polyhedra) Let F be a submodular function such that $F(\varnothing) = 0$. Let $w \in \mathbb{R}^p$, with components ordered in decreasing order, i.e., $w_{j_1} \geqslant \cdots \geqslant w_{j_p}$ and define $s_{j_k} = F(\{j_1, \ldots, j_k\}) - F(\{j_1, \ldots, j_{k-1}\})$. Then $s \in B(F)$ and,
(a) if $w \in \mathbb{R}_+^p$, s is a maximizer of $\max_{s \in P(F)} w^\top s$, and $\max_{s \in P(F)} w^\top s = f(w)$,
(b) s is a maximizer of $\max_{s \in B(F)} w^\top s$, and $\max_{s \in B(F)} w^\top s = f(w)$.

Proof. By convex duality (which applies because $P(F)$ has non empty interior from Prop. 1.3), we have, by introducing Lagrange multipliers $\lambda_A \in \mathbb{R}_+$ for the constraints $s(A) \leqslant F(A)$, $A \subset V$, the following pair of convex optimization problems dual to each other:

$$\max_{s \in P(F)} w^\top s = \min_{\lambda_A \geqslant 0, A \subset V} \max_{s \in \mathbb{R}^p} \left\{ w^\top s - \sum_{A \subset V} \lambda_A [s(A) - F(A)] \right\} \quad (2.5)$$

$$= \min_{\lambda_A \geqslant 0, A \subset V} \max_{s \in \mathbb{R}^p} \left\{ \sum_{A \subset V} \lambda_A F(A) + \sum_{k=1}^{p} s_k \left(w_k - \sum_{A \ni k} \lambda_A \right) \right\}$$

$$= \min_{\lambda_A \geqslant 0, A \subset V} \sum_{A \subset V} \lambda_A F(A) \text{ such that } \forall k \in V, \ w_k = \sum_{A \ni k} \lambda_A.$$

If we take the (primal) candidate solution s obtained from the greedy algorithm, we have $f(w) = w^\top s$ from Eq. (2.1). We now show that s is feasible (i.e., in $P(F)$), as a consequence of the submodularity of F. Indeed, without loss of generality, we assume that $j_k = k$ for all $k \in \{1, \ldots, p\}$. We can decompose any subset of $\{1, \ldots, p\}$ as $A =$

$A_1 \cup \cdots \cup A_m$, where $A_k = (u_k, v_k]$ are *integer* intervals. We then have:

$$s(A) = \sum_{k=1}^{m} s(A_k)\} \text{ by modularity}$$

$$= \sum_{k=1}^{m} \big\{ F((0, v_k]) - F((0, u_k]) \big\}$$

$$\leqslant \sum_{k=1}^{m} \big\{ F((u_1, v_k]) - F((u_1, u_k]) \big\} \text{ by submodularity}$$

$$= F((u_1, v_1]) + \sum_{k=2}^{m} \big\{ F((u_1, v_k]) - F((u_1, u_k]) \big\}$$

$$\leqslant F((u_1, v_1]) + \sum_{k=2}^{m} \big\{ F((u_1, v_1] \cup (u_2, v_k]) - F((u_1, v_1] \cup (u_2, u_k]) \big\}$$

$$\text{by submodularity}$$

$$= F((u_1, v_1] \cup (u_2, v_2])$$
$$+ \sum_{k=3}^{m} \big\{ F((u_1, v_1] \cup (u_2, v_k]) - F((u_1, v_1] \cup (u_2, u_k]) \big\}.$$

By pursuing applying submodularity, we finally obtain that $s(A) \leqslant F((u_1, v_1] \cup \cdots \cup (u_m, v_m]) = F(A)$, i.e., $s \in P(F)$.

Moreover, we can define dual variables $\lambda_{\{j_1, \ldots, j_k\}} = w_{j_k} - w_{j_{k+1}}$ for $k \in \{1, \ldots, p-1\}$ and $\lambda_V = w_{j_p}$ with all other λ_A equal to zero. Then they are all non negative (notably because $w \geqslant 0$), and satisfy the constraint $\forall k \in V$, $w_k = \sum_{A \ni k} \lambda_A$. Finally, the dual cost function has also value $f(w)$ (from Eq. (2.2)). Thus by duality (which holds, because $P(F)$ has a non-empty interior), s is an optimal solution. Note that it is not unique (see Prop. 4.2 for a description of the set of solutions).

In order to show (b), we may first assume that $w \geqslant 0$, we may replace $P(F)$ by $B(F)$, by simply dropping the constraint $\lambda_V \geqslant 0$ in Eq. (2.5). Since the solution obtained by the greedy algorithm satisfies $s(V) = F(V)$, we get a pair of primal-dual solutions, hence the optimality.

The result generalizes to all possible w, because we may add a large constant vector to w, which does not change the maximization with respect to $B(F)$ (since it includes the constraint $s(V) = F(V)$). $\square$

The next proposition draws precise links between convexity and submodularity, by showing that a set-function F is submodular if and only if its Lovász extension f is convex [92]. This is further developed in Prop. 2.4 where it is shown that, when F is submodular, minimizing F on 2^V (which is equivalent to minimizing f on $\{0,1\}^p$ since f is an extension of F) and minimizing f on $[0,1]^p$ are equivalent.

Proposition 2.3. (Convexity and submodularity) A set-function F is submodular if and only if its Lovász extension f is convex.

Proof. Let $A, B \subset V$. The vector $1_{A \cup B} + 1_{A \cap B} = 1_A + 1_B$ has components equal to 0 (on $V \backslash (A \cup B)$), 2 (on $A \cap B$) and 1 (on $A \Delta B = (A \backslash B) \cup (B \backslash A)$). Therefore, $f(1_{A \cup B} + 1_{A \cap B}) = \int_0^2 F(1_{\{w \geqslant z\}}) dz = \int_0^1 F(A \cup B) dz + \int_1^2 F(A \cap B) dz = F(A \cup B) + F(A \cap B)$.

If f is convex, then by homogeneity, $f(1_A + 1_B) \leqslant f(1_A) + f(1_B)$, which is equal to $F(A) + F(B)$, and thus F is submodular.

If F is submodular, then by Prop. 2.2, for all $w \in \mathbb{R}_+^p$, $f(w)$ is a maximum of linear functions, thus, it is convex on $\mathbb{R}_+^p$. Moreover, because $f(w + \alpha 1_V) = f(w) + \alpha F(V)$, it is convex on $\mathbb{R}^p$. $\qquad\square$

The next proposition completes Prop. 2.3 by showing that minimizing the Lovász extension on $[0,1]^p$ is equivalent to minimizing it on $\{0,1\}^p$, and hence to minimizing the set-function F on 2^V (when F is submodular).

Proposition 2.4. (Minimization of submodular functions) Let F be a submodular function and f its Lovász extension; then $\min_{A \subset V} F(A) = \min_{w \in \{0,1\}^p} f(w) = \min_{w \in [0,1]^p} f(w)$.

Proof. Because f is an extension from $\{0,1\}^p$ to $[0,1]^p$ (property (d) from Prop. 2.1), we must have $\min_{A \subset V} F(A) = \min_{w \in \{0,1\}^p} f(w) \geqslant \min_{w \in [0,1]^p} f(w)$. For the other inequality, any $w \in [0,1]^p$ may be decomposed as $w = \sum_{i=1}^p \lambda_i 1_{B_i}$ where $B_1 \subset \cdots \subset B_p = V$, where λ is nonnegative and has a sum smaller than or equal to one (this can be obtained by considering B_i the set of indices of the i largest values of

w). We then have $f(w) = \sum_{i=1}^{p} \int_{\sum_{k=1}^{i-1} \lambda_k}^{\sum_{k=1}^{i} \lambda_k} F(B_i) dz = \sum_{i=1}^{p} \lambda_i F(B_i) \geqslant \sum_{i=1}^{p} \lambda_i \min_{A \subset V} F(A) \geqslant \min_{A \subset V} F(A)$ (because $\min_{A \subset V} F(A) \leqslant 0$). This leads to the desired result.

Note that the last equality shows that the minimizers of $f(w)$ on $w \in [0,1]^p$ must have sup-level sets (i.e., the sets B_i defined above) which are minimizers of F (i.e., w is a convex hull of the indicator vectors of all minimizers of F). $\qquad\square$

We end this section, by simply stating the greedy algorithm for the symmetric independence polyhedron, whose proof is similar to the proof of Prop. 2.2 (we define the sign of a as $+1$ if $a > 0$, and -1 if $a < 0$, and zero otherwise; $|w|$ denotes the vector composed of the absolute values of the components of w).

Proposition 2.5. (Greedy algorithm for symmetric independence polyhedron) Let F be a submodular function such that $F(\varnothing) = 0$ and F is non-decreasing. Let $w \in \mathbb{R}^p$. A maximizer of $\max_{s \in |P|(F)} w^\top s$ may be obtained by the following algorithm: order the components of $|w|$, as $|w_{j_1}| \geqslant \cdots \geqslant |w_{j_p}|$ and define $s_{j_k} = \text{sign}(w_{j_k})[F(\{j_1, \ldots, j_k\}) - F(\{j_1, \ldots, j_{k-1}\})]$. Moreover, for all $w \in \mathbb{R}^p$, $\max_{s \in |P|(F)} w^\top s = f(|w|)$.

2.3 Structured sparsity and convex relaxations

Structured sparsity. The concept of parsimony is central in many scientific domains. In the context of statistics, signal processing or machine learning, it takes the form of variable or feature selection problems.

In a supervised learning problem, we aim to predict n responses $y_i \in \mathbb{R}$, from n observations $x_i \in \mathbb{R}^p$, for $i \in \{1, \ldots, n\}$. In this paper, we focus on linear predictors of the form $f(x) = w^\top x$, where $w \in \mathbb{R}^p$ (for extensions to non-linear predictions, see [4, 5] and references therein). We consider estimators obtained by the following regularized empirical

risk minimization formulation:

$$\min_{w\in\mathbb{R}^p} \frac{1}{n}\sum_{i=1}^{n}\ell(y_i, w^\top x_i) + \lambda\Omega(w), \tag{2.6}$$

where $\ell(y,\hat{y})$ is a loss between a prediction $\hat{y}$ and the true response y, and Ω is a norm. Typically, the quadratic loss $\ell(y,\hat{y}) = \frac{1}{2}(y-\hat{y})^2$ is used for regression problems and the logistic loss $\ell(y,\hat{y}) = \log(1+\exp(-y\hat{y}))$ is used for binary classification problems where $y \in \{-1,1\}$ (see, e.g., [128] and [58] for more complete descriptions of loss functions).

In order to promote sparsity, the ℓ_1-norm is commonly used and, in a least-squares regression framework is referred to as the Lasso [131] in statistics and as basis pursuit [24] in signal processing. Sparse models are commonly used in two situations: First, to make the model or the prediction more interpretable or cheaper to use, i.e., even if the underlying problem does not admit sparse solutions, one looks for the best sparse approximation. Second, sparsity can also be used given prior knowledge that the model should be sparse. In these two situations, reducing parsimony to finding models with low cardinality turns out to be limiting, and structured parsimony has emerged as a fruitful practical extension, with applications to image processing, text processing, bioinformatics or audio processing (see, e.g., [140, 74, 68, 71, 82, 76, 95, 90], a review in [8, 9] and Section 3 for various examples, and in particular Section 3.3 for relationships with grouped ℓ_1-norm with overlapping groups).

Convex relaxation of combinatorial penalty. Most of the work based on convex optimization and the design of dedicated sparsity-inducing norms has focused mainly on the specific allowed set of sparsity patterns [140, 74, 71, 76]: if $w \in \mathbb{R}^p$ denotes the predictor we aim to estimate, and $\text{Supp}(w)$ denotes its support, then these norms are designed so that penalizing with these norms only leads to supports from a given family of allowed patterns. We can instead follow the approach of [59, 68] and consider specific penalty functions $F(\text{Supp}(w))$ of the support set $\text{Supp}(w) = \{j \in V, \ w_j \neq 0\}$, which go beyond the cardinality function, but are not limited or designed to only forbid certain sparsity patterns. As first shown in [6], for *non-decreasing* submodular

functions, these may also lead to restricted sets of supports but their interpretation in terms of an *explicit* penalty on the support leads to additional insights into the behavior of structured sparsity-inducing norms.

While direct greedy approaches (i.e., forward selection) to the problem are considered in [59, 68], submodular analysis may be brought to bear to provide convex relaxations to the function $w \mapsto F(\text{Supp}(w))$, which extend the traditional link between the ℓ_1-norm and the cardinality function.

Proposition 2.6. (Convex relaxation of functions defined through supports) Let F be a non-decreasing submodular function. The function $w \mapsto f(|w|)$ is the convex envelope (tightest convex lower bound) of the function $w \mapsto F(\text{Supp}(w))$ on the unit ℓ_∞-ball $[-1,1]^p$.

Proof. We use the notation $|w|$ to denote the p-dimensional vector composed of the absolute values of the components of w. We denote by g^* the Fenchel conjugate (see definition in Appendix A) of $g : w \mapsto F(\text{Supp}(w))$ on the domain $\{w \in \mathbb{R}^p, \ \|w\|_\infty \leqslant 1\} = [-1,1]^p$, and g^{**} its bidual [17]. We only need to show that the Fenchel bidual is equal to the function $w \mapsto f(|w|)$. By definition of the Fenchel conjugate, we have:

$$
\begin{aligned}
g^*(s) &= \max_{\|w\|_\infty \leqslant 1} w^\top s - g(w) \\
&= \max_{\delta \in \{0,1\}^p} \max_{\|w\|_\infty \leqslant 1} (\delta \circ w)^\top s - f(\delta) \text{ by definition of } g, \\
&= \max_{\delta \in \{0,1\}^p} \delta^\top |s| - f(\delta) \text{ by maximizing out } w, \\
&= \max_{\delta \in [0,1]^p} \delta^\top |s| - f(\delta) \text{ because } F - |s| \text{ is submodular.}
\end{aligned}
$$

Thus, for all w such that $\|w\|_\infty \leqslant 1$,

$$
\begin{aligned}
g^{**}(w) &= \max_{s \in \mathbb{R}^p} s^\top w - g^*(s) \\
&= \max_{s \in \mathbb{R}^p} \min_{\delta \in [0,1]^p} s^\top w - \delta^\top |s| + f(\delta)
\end{aligned}
$$

By strong convex duality (which applies because Slater's condition [17]

is satisfied), we get:

$$g^{**}(w) \quad = \quad \min_{\delta \in [0,1]^p} \max_{s \in \mathbb{R}^p} \; s^\top w - \delta^\top |s| + f(\delta)$$

by strong duality and

$$= \quad \min_{\delta \in [0,1]^p, \delta \geqslant |w|} f(\delta) = f(|w|) \text{ because } F \text{ is nonincreasing,}$$

which leads to the desired result. Note that F non-increasing implies that f is non-increasing with respect to all of its components. $\qquad\square$

The previous proposition provides a relationship between combinatorial optimization problems (involving functions of the form $w \mapsto F(\mathrm{Supp}(w))$) and convex optimization problems involving the Lovász extension. A desirable behavior of a convex relaxation is that some of the properties of the original problem are preserved. In this paper, we will focus mostly on the allowed set of sparsity patterns (see below and Section 4.3). For more details about theroretical guarantees and applications of submodular functions to structured sparsity, see [6, 7]. In Section 3, we consider several example of submodular functions and present when appropriate how they translate to sparsity-inducing norms.

Optimization for regularized risk minimization.   Given the representation of Ω as the maximum of linear functions (Prop. 2.5), we can easily obtain a subgradient of Ω, thus allowing the use of subgradient descent techniques (see a description in Appendix A.2). However, these methods typically require many iterations, and given the structure of our norms, more efficient methods are available: we describe in Section 5.1 *proximal methods*, which generalizes soft-thresholding algorithms for the ℓ_1-norm and grouped ℓ_1-norm, and can use efficiently the combinatorial structure of the norms.

Structured sparsity-inducing norms and dual balls.   We assume in this paragraph that F is submodular and non-decreasing, and such that the values on all singletons is strictly positive. The function $\Omega : w \mapsto f(|w|)$ is then a norm [6]. Through the representation $\Omega(w) = \max_{s \in P(F)} |w|^\top s = \max_{|s| \in P(F)} w^\top s = \max_{s \in |P|(F)} w^\top s,$

the dual norm is equal to $\Omega^*(s) = \max_{A \subset V} \frac{|s|(A)}{F(A)} = \max_{A \subset V} \frac{\|s_A\|_1}{F(A)}$, and the unit dual ball is the symmetric independence polyhedron $|P|(F) = \{s \in \mathbb{R}^p, \ |s| \in P(F)\} = \{s \in \mathbb{R}^p, \ \forall A \subset V, \|s_A\|_1 \leqslant A\}$ (see Appendix A for more details on polar sets and dual norms).

The dual ball $|P|(F) = \{s \in \mathbb{R}^p, \ \Omega^*(s) \leqslant 1\}$ is naturally characterized by half planes of the form $\frac{w^\top s}{F(\mathrm{Supp}(w))} \leqslant$ for $w \in \{-1, 0, 1\}^p$. Thus, the unit ball of Ω is the convex hull of the vectors $\frac{1}{F(\mathrm{Supp}(w))}w$ for the same vectors $w \in \{-1, 0, 1\}^p$. See Figure 2.3 for examples for $p = 2$ and Figure 2.4 for examples with $p = 3$.

A particular feature of the unit ball of Ω is that it has faces which are composed of vectors with many zeros, leading to structured sparsity (see Section 3.3 for examples and Section 4.2, for more details about the facial structure of the symmetric independence polyhedron). However, as can be seen in Figures 2.3 and 2.4, there are additional extreme points and faces where many of the components of $|w|$ are equal (e.g., the corners of the ℓ_∞-ball). In the context of sparsity-inducing norms, this has the sometimes undesirable effect of inducing vectors with many components of equal magnitude. As shown in [115], this effect due to the ℓ_∞-norm in Prop. 2.6 may be corrected by the appropriate use of ℓ_q-norms $q \in (1, \infty)$, which we now present for the ℓ_2-norm.

ℓ_2-relaxations of submodular penalties.   Given a non-decreasing submodular function such that $F(\{k\}) > 0$ for all $k \in V$, we may define a norm Θ as follows:

$$\Theta(w) = \frac{1}{2} \min_{\eta \in \mathbb{R}^p_+} \left\{ \frac{w_i^2}{\eta_i} + f(\eta) \right\},$$

using the usual convention that that $\frac{w_i^2}{\eta_i}$ is equal to zero as soon as $w_i = 0$, and equal to $+\infty$ if $w_i \neq 0$ and $\eta_i = 0$ (for more details on variational representations of any norms through squared ℓ_2-norm, see [8]). As shown in [115], this defines a norm, which shares the same sparsity-inducing effects as Ω, without the extra singular points. Moreover, the optimization results presented in this paper can be used as well to derive efficient algorithms for optimization problems regularized by this norm. Moreover, Prop. 2.6 may be extended, and Θ is the convex envelope of the function $w \mapsto F(\mathrm{Supp}(w))\|w\|_2$, or the homogeneous convex
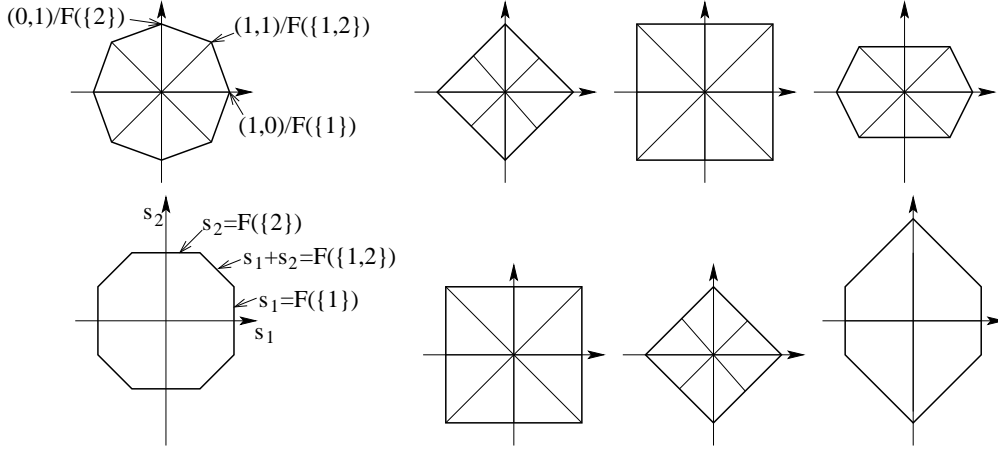
Fig. 2.3: Polyhedral unit ball of Ω (top) with the associated dual unit ball (bottom), for 4 different submodular functions (two variables), with different sets of extreme points; changing values of F may make some of the extreme points disappear (see the notion of stable sets in Section 4.3). From left to right: $F(A) = |A|^{1/2}$ (all possible extreme points), $F(A) = |A|$ (leading to the ℓ_1-norm), $F(A) = \min\{|A|, 1\}$ (leading to the ℓ_∞-norm), $F(A) = \frac{1}{2}1_{\{A \cap \{2\} \neq \varnothing\}} + 1_{\{A \neq \varnothing\}}$ (leading to the structured norm $\Omega(w) = \frac{1}{2}|w_2| + \|w\|_\infty$). Extreme points of the primal balls correspond to full-dimensional faces of the dual ball, and vice-versa.

envelope (the tightest homogeneous convex lower bound) of the function $w \mapsto \frac{1}{2}F(\mathrm{Supp}(w)) + \frac{1}{2}\|w\|_2^2$, thus replacing the ℓ_∞-constraint by an ℓ_2-penalty.

Shaping level sets through symmetric submodular functions. For a non-decreasing submodular function F, we have defined a norm $\Omega(w) = f(|w|)$, that essentially allows the definition of a prior knowledge on supports of predictors w. When using the Lovász extension directly for symmetric submodular functions, then it turns out that the effect is on all sub-level sets $\{w \leqslant \alpha\}$ and not only on the support $\{w \neq 0\}$. Indeed, as shown in [7], the Lovász extension is the convex envelope of the function $w \mapsto \max_{\alpha \in \mathbb{R}} F(\{w \leqslant \alpha\})$ on the set

$$F(A) = |A|$$
$$\Omega(w) = \|w\|_1$$

$$F(A) = \min\{|A|, 1\}$$
$$\Omega(w) = \|w\|_\infty$$

$$F(A) = |A|^{1/2}$$
all possible extreme points

$$F(A) = 1_{\{A \cap \{1\} \neq \varnothing\}} + 1_{\{A \cap \{2,3\} \neq \varnothing\}}$$
$$\Omega(w) = |w_1| + \|w_{\{2,3\}}\|_\infty$$

$$F(A) = 1_{\{A \cap \{1,2,3\} \neq \varnothing\}}$$
$$+ 1_{\{A \cap \{2,3\} \neq \varnothing\}} + 1_{\{A \cap \{3\} \neq \varnothing\}}$$
$$\Omega(w) = \|w\|_\infty + \|w_{\{2,3\}}\|_\infty + |w_3|$$
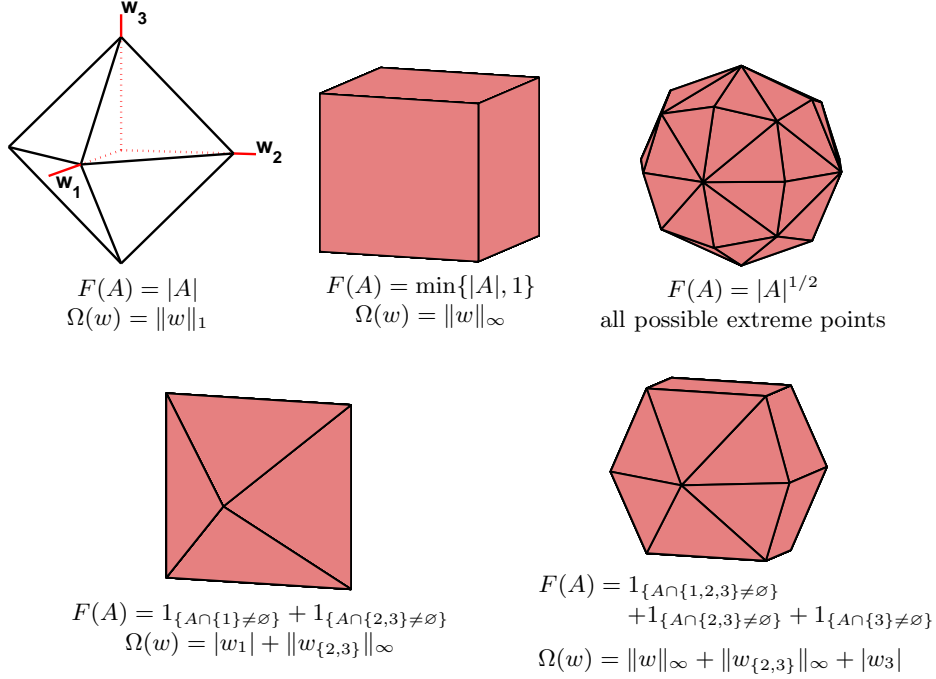
Fig. 2.4: Unit balls for structured sparsity-inducing norms, with the corresponding submodular functions and the associated norm.

$[0,1]^p + \mathbb{R}1_V = \{w \in \mathbb{R}^p, \ \max_{k \in V} w_k - \min_{k \in V} w_k \leqslant 1\}$.

The main examples of such symmetric functions are cuts in undirected graphs, which we describe in Section 3.2, leading to the total variation, but other examples are interesting as well for machine learning (see [7]). Finally, while the facial structure of the symmetric independence polyhedron $|P|(F)$ was key to analysing the regularization properties for shaping supports, the base polyhedron $B(F)$ is the proper polyhedron (see Section 4.2 for more details).

3

Examples and applications of submodular functions

We now present classical examples of submodular functions. For each of these, we also describe the corresponding Lovász extensions, and, when appropriate, the associated submodular polyhedra. We also present applications to machine learning, either through formulations as combinatorial optimization problems of through the regularization properties of the Lovász extension. We are by no means exhaustive and other applications may be found in facility location [31, 30, 1], game theory [45], document summarization [91], social networks [81], or clustering [107].

Note that in Appendix B.2, we present several operations that preserve submodularity (such as symmetrization and partial minimization), which can be applied to any of the functions presented in this section, thus defining new functions.

3.1 Cardinality-based functions

We consider functions that depend only on $s(A)$ for a certain $s \in \mathbb{R}^p_+$. If $s = 1_V$, these are functions of the cardinality. The next proposition shows that only concave functions lead to submodular functions, which is coherent with the diminishing return property from

Section 1 (Prop. 1.1).

Proposition 3.1. (Submodularity of cardinality-based set-functions) If $s \in \mathbb{R}_+^p$ and $g : \mathbb{R}_+ \to \mathbb{R}$ is a concave function, then $F : A \mapsto g(s(A))$ is submodular. If $F : A \mapsto g(s(A))$ is submodular for all $s \in \mathbb{R}_+^p$, then g is concave.

Proof. The function $F : A \mapsto g(s(A))$ is submodular if and only if for all $A \subset V$ and $j, k \in V \backslash A$: $g(s(A) + s_k) - g(s(A)) \geqslant g(s(A) + s_k + s_j) - g(s(A) + s_j)$. If g is concave and $a \geqslant 0$, $t \mapsto g(a + t) - g(t)$ is non-increasing, hence the first result. Moreover, if $t \mapsto g(a+t) - g(t)$ is non-increasing for all $a \geqslant 0$, then g is concave, hence the second result. $\square$

Proposition 3.2. (Lovász extension of cardinality-based set-functions) Let $s \in \mathbb{R}_+^p$ and $g : \mathbb{R}_+ \to \mathbb{R}$ be a concave function such that $g(0) = 0$, the Lovász extension of the submodular function $F : A \mapsto g(s(A))$ is equal to

$$f(w) = \sum_{k=1}^{p} w_{j_k}[g(s_{j_1} + \cdots + s_{j_k}) - g(s_{j_1} + \cdots + s_{j_{k-1}})].$$

If $s = 1_V$, i.e., $F(A) = g(|A|)$, then $f(w) = \sum_{k=1}^{p} w_{j_k}[g(k) - g(k-1)]$.

Thus, for functions of the cardinality (for which $s = 1_V$), the Lovász extension is thus a linear combination of order statistics (i.e., r-th largest component of w, for $r \in \{1, \ldots, p\}$).

Application to machine learning.   In terms of set functions, considering $g(s(A))$ instead of $s(A)$ does not make a significant difference. However, it does in terms of the Lovász extension. Indeed, as shown in [7], using the Lovász extension for regularization encourages components of w to be equal (see also Section 2.3), and hence provides a convex prior for clustering or outlier detection, depending on the choice of the concave function g (see more details in [7, 64]). This is a situation where this effect has positive desired consequences.

Some special cases of non-decreasing functions are of interest, such as $F(A) = |A|$, for which $f(w) = w^\top 1_V$ and Ω is the ℓ_1-norm, and $F(A) = 1_{|A|>0}$ for which $f(w) = \max_{k \in V} w_k$ and Ω is the ℓ_∞-norm. When restricted to subsets of V and then linearly combined, we obtain set covers defined in Section 3.3. Other interesting examples of combinations of functions of restricted weighted cardinality functions may be found in [130, 83].

3.2 Cut functions

Given a set of (non necessarily symmetric) weights $d : V \times V \to \mathbb{R}_+$, define the cut as

$$F(A) = \sum_{k \in A, \ j \in V \backslash A} d(k, j),$$

which we denote $d(A, V \backslash A)$. Note that for a cut function and disjoint subsets A, B, C, we always have (see [35] for more details):

$$
\begin{aligned}
F(A \cup B \cup C) \ =\ & F(A \cup B) + F(A \cup C) + F(B \cup C) \\
& \qquad\qquad -F(A) - F(B) - F(C) + F(\varnothing) \\
F(A \cup B) \ =\ & d(A \cup B, (A \cup B)) = d(A, A^{\mathsf{c}} \cap B^{\mathsf{c}}) + d(B, A^{\mathsf{c}} \cap B^{\mathsf{c}}) \\
\leqslant\ & d(A, A^{\mathsf{c}}) + d(B, B^{\mathsf{c}}) = F(A) + F(B),
\end{aligned}
$$

where we denote $A^{\mathsf{c}} = V \backslash A$. This implies that F is sub-additive. We then have, for any sets $A, B \subset V$:

$$
\begin{aligned}
& F(A \cup B) \\
=\ & F([A \cap B] \cup [A \backslash B] \cup [B \backslash A]) \\
=\ & F([A \cap B] \cup [A \backslash B]) + F([A \cap B] \cup [B \backslash A]) + F([A \backslash B] \cup [B \backslash A]) \\
& -F(A \cap B) - F(A \backslash B) - F(B \backslash A) + F(\varnothing) \\
=\ & F(A) + F(B) + F(A \Delta B) - F(A \cap B) - F(A \backslash B) - F(B \backslash A) \\
=\ & F(A) + F(B) - F(A \cap B) + [F(A \Delta B) - F(A \backslash B) - F(B \backslash A)] \\
\leqslant\ & F(A) + F(B) - F(A \cap B), \text{ by sub-additivity,}
\end{aligned}
$$

which shows submodularity. Moreover, the Lovász extension is equal to

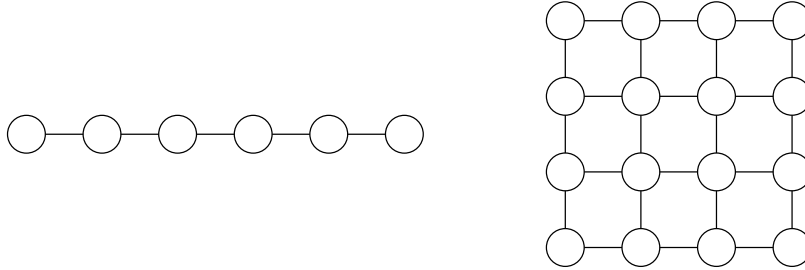$$f(w) = \sum_{k,j \in V} d(k, j)(w_k - w_j)_+$$

Fig. 3.1: Two-dimensional grid with 4-connectivity. The cut in these undirected graphs lead to Lovász extensions which are certain versions of total variations, which enforce level sets of w to be connected with respect to the graph.

(which provides an alternative proof of submodularity owing to Prop. 2.3). Thus, if the weight function d is symmetric, then the submodular function is also symmetric and the Lovász extension is even (from Prop. 2.1). Examples of graphs related to such cuts (i.e., graphs defined on V for which there is an edge from k to j if and only if $d(k, j) > 0$) are shown in Figures 3.1 and 3.2. An interesting instance of these Lovász extensions plays a crucial role in signal and image processing; indeed, for a graph composed of a two-dimensional grid with 4-connectivity (see Figure 3.1), we obtain a certain version of the total variation, which is a common prior to induce piecewise-constant signals (see applications to machine learning below). In fact, some of the results presented in this paper were first shown on this particular case (see, e.g., [21] and references therein).

Note that these functions can be extended to cuts in hypergraphs, which may have interesting applications in computer vision [18]. Moreover, directed cuts (i.e., when $d(k, j)$ and $d(j, k)$ may be different) may be interesting to favor increasing or decreasing jumps along the edges of the graph. Finally, there is another interesting link between directed cuts and isotonic regression (see, e.g., [93] and references therein), which corresponds to solving a separable optimization problem regularized by a large constant times the associated Lovász extension. See another link with isotonic regression in Section 5.4.

Interpretation in terms of quadratic functions of indicator variables. For undirected graphs (i.e., for which the function d is symmetric), we may rewrite the cut as follows:

$$F(A) = \frac{1}{2} \sum_{k=1}^{p} \sum_{j=1}^{p} d(k,j)|(1_A)_k - (1_A)_j|$$

$$= \frac{1}{2} \sum_{k=1}^{p} \sum_{j=1}^{p} d(k,j)|(1_A)_k - (1_A)_j|^2$$

because $|(1_A)_k - (1_A)_j|^2 \in \{0,1\}$. This leads to

$$F(A) = \frac{1}{2} \sum_{k=1}^{p} \sum_{j=1}^{p} (1_A)_k (1_A)_j \Big[1_{j=k} \sum_{i=1}^{p} d(i,k) - d(j,k) \Big]$$

$$= \frac{1}{2} 1_A^\top Q 1_A,$$

with $Q = \text{Diag}(D1) - D$ where D is the square weighted affinity matrix obtained from d, which has non-positive diagonal elements (Q is the Laplacian of the graph [27]). It turns out that a sum of linear and quadratic functions of 1_A is submodular only in this situation.

Proposition 3.3. (Submodularity of quadratic functions) Let $Q \in \mathbb{R}^{p \times p}$ and $q \in \mathbb{R}^p$. Then the function $F : A \mapsto q^\top 1_A + \frac{1}{2} 1_A^\top Q 1_A$ is submodular if and only if all off-diagonal elements of Q are non-positive.

Proof. Since cuts are submodular, the previous developments show that the condition is sufficient. It is necessary by simply considering the inequality $0 \leqslant F(\{i\}) + F(\{j\}) - F(\{i,j\}) = q_i + \frac{1}{2}Q_{ii} + q_j + \frac{1}{2}Q_{jj} - [q_i + q_j + \frac{1}{2}Q_{ii} + \frac{1}{2}Q_{jj} + Q_{ij}] = -Q_{ij}$. $\qquad\square$

Regular functions and robust total variation. By partial minimization, we obtain so-called *regular functions* [18, 21]. One application is "noisy cut functions": for a given weight function $d : W \times W \to \mathbb{R}_+$, where each node in W is uniquely associated in a node in V, we consider the submodular function obtained as the minimum cut

adapted to A in the augmented graph (see top-right plot of Figure 3.2): $F(A) = \min_{B \subset W} \sum_{k \in B, \, j \in W \setminus B} d(k, j) + \lambda |A \Delta B|$, where $A \Delta B = (A \setminus B) \cup (B \setminus A)$ is the symmetric difference between sets A and B. This allows for robust versions of cuts, where some gaps may be tolerated; indeed, compared to having directly a small cut for A, B needs to have a small cut and be close to A, thus allowing some elements to be removed or added to A in order to lower the cut (see more details in [7]).

The class of regular functions is particularly interesting, because it leads to a family of submodular functions for which dedicated fast algorithms exist. Indeed, minimizing the cut functions or the partially minimized cut, plus a modular function defined by $z \in \mathbb{R}^p$, may be done with a min-cut/max-flow algorithm (see, e.g., [29]). Indeed, following [18, 21], we add two nodes to the graph, a source s and a sink t. All original edges have non-negative capacities $d(k, j)$, while, the edge that links the source s to the node $k \in V$ has capacity $(z_k)_+$ and the edge that links the node $k \in V$ to the sink t has weight $-(z_k)_-$ (see bottom line of Figure 3.2). Finding a minimum cut or maximum flow in this graph leads to a minimizer of $F - z$. For a detailed study of the expressive power of functions expressible in terms of graph cuts, see, e.g., [141, 22].

For proximal methods, such as defined in Eq. (5.5) (Section 5), we have $z = \psi(\alpha)$ and we need to solve an instance of a *parametric max-flow* problem, which may be done using efficient dedicated algorithms [51, 62, 21]. See also Section 7.3 for generic algorithms based on a sequence of singular function minimizations.

Applications to machine learning. Finding minimum cuts in undirected graphs such as two-dimensional grids or extensions thereof in more than two dimesions has become an important tool in computer vision for image segmentation, where it is commonly referred to as *graph cut* techniques (see, e.g., [84] and references therein). In this context, several extensions have been considered, such as multi-way cuts, where exact optimization is not possible anymore, and a sequence of binary graph cuts is used to find an approximate minimum (see also [108] for a
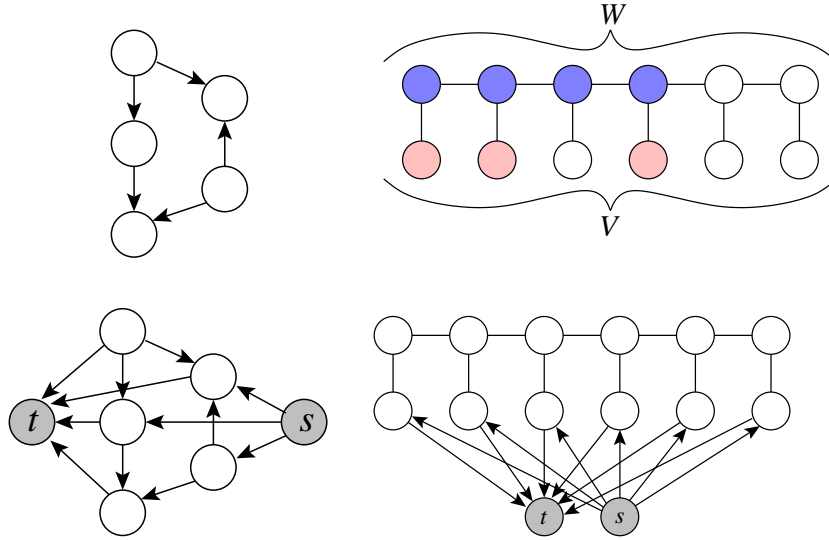
Fig. 3.2: Top: directed graph (left) and undirected corresponding to regular functions (which can be obtained from cuts by partial minimization; a set $A \subset V$ is displayed in red, with a set $B \subset W$ with small cut but one more element than A, see text in Section 3.2 for details). Bottom: graphs corresponding to the $s - t$ min-cut formulation for minimizing the submodular function above plus a modular function (see text for details).

specific multi-way extension based on different submodular functions).

The Lovász extension of cuts in an undirected graph, often referred to as the total variation, has now become a classical regularizer in signal processing and machine learning: given a graph, it will encourages solutions to be piecewise-constant according to the graph (as opposed to the graph Laplacian, which will impose smoothness along the edges of the graph) [65, 64]. See Section 4.2 for a formal description of the sparsity-inducing properties of the Lovász extension; for chain graphs, we obtain usual piecewise constant vectors, and the have many applications in sequential problems (see, e.g., [57, 132, 94, 21] and references therein). Note that in this context, separable optimization problems

considered in Section 5 are heavily used and that algorithms presented in Section 6 provide unified and efficient algorithms for all these situations.

3.3   Set covers

Given a *non-negative* set-function $D : 2^V \to \mathbb{R}_+$, then we can define a set-function F through

$$F(A) \quad = \sum_{G \subset V, \ G \cap A \neq \varnothing} D(G),$$

with Lovász extension $f(w) = \sum_{G \subset V} D(G) \max_{k \in G} w_k$.

   The submodularity and the Lovász extension can be obtained using linearity and the fact that the Lovász extension of $A \mapsto 1_{G \cap A \neq \varnothing}$ is $w \mapsto \max_{k \in G} w_k$. In the context of structured sparsity-inducing norms (see Section 2.3), these correspond to penalties of the form $w \mapsto f(|w|) = \sum_{G \subset V} D(G) \|w_G\|_\infty$, thus leading to overlapping group Lasso formulations (see, e.g., [140, 74, 68, 71, 82, 76, 95]). For example, when $D(G) = 1$ for elements of a given partition, and zero otherwise, then $F(A)$ counts the number of elements of the partition with non-empty intersection with A. This leads to the classical non-overlapping grouped ℓ_1/ℓ_∞-norm.

Möbius inversion.   Note that any set-function F may be written as

$$F(A) = \sum_{G \subset V, \ G \cap A \neq \varnothing} D(G) = \sum_{G \subset V} D(G) - \sum_{G \subset V \backslash A} D(G),$$

i.e., $F(V) - F(V \backslash A) = \sum_{G \subset A} D(G),$

for a certain set-function D, *which is not usually non-negative*. Indeed, by Möbius inversion formula[1] (see, e.g., [47]), we have:

$$D(G) = \sum_{A \subset G} (-1)^{|G| - |A|} \big[F(V) - F(V \backslash A) \big].$$

---

[1] If F and G are any set functions such that $\forall A \subset V$, $F(A) = \sum_{B \subset A} G(B)$, then $\forall A \subset V$, $G(A) = \sum_{B \subset A} (-1)^{|A \backslash B|} F(B)$.

Thus, functions for which D is non-negative form a specific subset of submodular functions (note that for all submodular functions, the function $D(G)$ is non-negative for all pairs $G = \{i, j\}$, for $j \neq i$, as a consequence of Prop. 1.2). Moreover, these functions are always non-decreasing. For further links, see [49], where it is notably shown that $D(G) = 0$ for all sets G of cardinality greater or equal to three for cut functions (which are second-order polynomials in the indicator vector).

Reinterpretation in terms of set-covers. Let W be any "base" set. Given for each $k \in V$, a set $S_k \subset W$, we define the cover as $F(A) = \left| \bigcup_{k \in A} S_k \right|$. More generally, we can define $F(A) = \sum_{j \in W} \Delta(j) 1_{\exists k \in A, S_k \ni j}$, if we have weights $\Delta(j) \in \mathbb{R}_+$ for $j \in W$ (this corresponds to replacing the cardinality function on W, by a weighted cardinality function, with weights defined by Δ). Then, F is submodular (as a consequence of the equivalence with the previously defined functions, which we now prove).

These two types of functions are in fact equivalent. Indeed, for a weight function $D : 2^V \to \mathbb{R}_+$, we consider the base set W to be the power-set of V, i.e., $W = 2^V$, and $S_k = \{G \subset V, G \ni k\}$, and $\Delta(G) = D(G)$, to obtain a set cover, since we then have

$$
\begin{aligned}
F(A) &= \sum_{G \subset V} D(G) 1_{A \cap G \neq \varnothing} = \sum_{G \subset V} D(G) 1_{\exists k \in A, k \in G} \\
&= \sum_{G \subset V} D(G) 1_{\exists k \in A, G \in S_k}.
\end{aligned}
$$

Moreover, for a certain set cover defined by W, $S_k \subset W$, $k \in V$, and $\Delta : W \mapsto \mathbb{R}_+$, define $G_j = \{k \in V, S_k \ni j\}$ the subset of V of points that cover $j \in W$. We can then write the set cover as

$$
F(A) = \sum_{j \in W} \Delta(j) 1_{\exists k \in A, S_k \ni j} = \sum_{j \in W} \Delta(j) 1_{A \cap G_j \neq \varnothing},
$$

to obtain a set-function expressed in terms of groups and non-negative weight functions.

Applications to machine learning. Submodular set-functions which can be expressed as set covers (or equivalently as a sum of max-

imum of certain components) have several applications, mostly as regular set-covers or through their use in sparsity-inducing norms.

When used as set covers, submodular functions are traditionally used because algorithms for maximization with theoretical guarantees may be used (see Section 8). See [88] for several applications.

When used through their Lovász extensions, we obtain structured sparsity-inducing norms which can be used to impose specific prior knowledge into learning problems: indeed, as shown in Section 2.3, they correspond to a convex relaxation to the set-function applied to the support of the predictor. Morever, as shown in [74, 6] and in Section 4.3, they lead to specific sparsity patterns (i.e., supports), which are stable for the submodular function, i.e., such that they cannot be increased without increasing the set-function. For this particular example, stable sets are exactly intersection of complements of groups G such that $D(G) > 0$ (see more details in [74]), that is, some of the groups with non-zero weights carve out the set V to obtain the support of the predictor. Note that following [95], all of these may be interpreted in terms of flows (see Section 3.4) in order to obtain fast algorithms to solve the proximal problems.

By choosing certain set of groups G such that $D(G) > 0$, we can model several interesting behaviors (see more details in [9]):

- **Line segments**: Given p variables organized in a sequence, using the set of groups of Figure 3.4, it is only possible to select *contiguous nonzero patterns*. In this case, we have p groups with non-zero weight, and the submodular function is equal, up to constants, to the length of the range of A (i.e., the distance beween the rightmost element of A and the leftmost element of A).

- **Two-dimensional convex supports**: Similarly, assume now that the p variables are organized on a two-dimensional grid. To constrain the allowed supports to be the set of all rectangles on this grid, a possible set of groups to consider may be composed of half planes with specific orientations: if only vertical and horizontal orientations are used, the set of allowed patterns is the set of rectangles, while with more
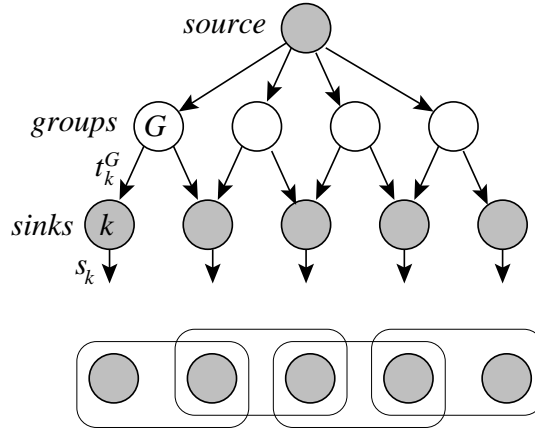
Fig. 3.3: Flow (top) and set of groups (bottom) for sequences. When these groups have unit weights (i.e., $D(G) = 1$ for these groups and zero for all others), then the submodular function $F(A)$ is equal to the number of sequential pairs with at least one present element. When applied to sparsity-inducing norms, this leads to supports that have no isolated points (see applications in [95]).

general orientations, more general convex patterns may be obtained. These can be applied for images, and in particular in structured sparse component analysis where the dictionary elements can be assumed to be localized in space [78].

- **Two-dimensional block structures on a grid**: Using sparsity-inducing regularizations built upon groups which are composed of variables together with their spatial neighbors (see Figure 3.4) leads to good performances for background subtraction [20, 10, 68, 95], topographic dictionary learning [79, 96], wavelet-based denoising [119].

- **Hierarchical structures**: here we assume that the variables are organized in a hierarchy. Precisely, we assume that the p variables can be assigned to the nodes of a tree (or a forest of trees), and that a given variable may be selected only if all its ancestors in the tree have already been selected. This corresponds to a set-function which counts the number
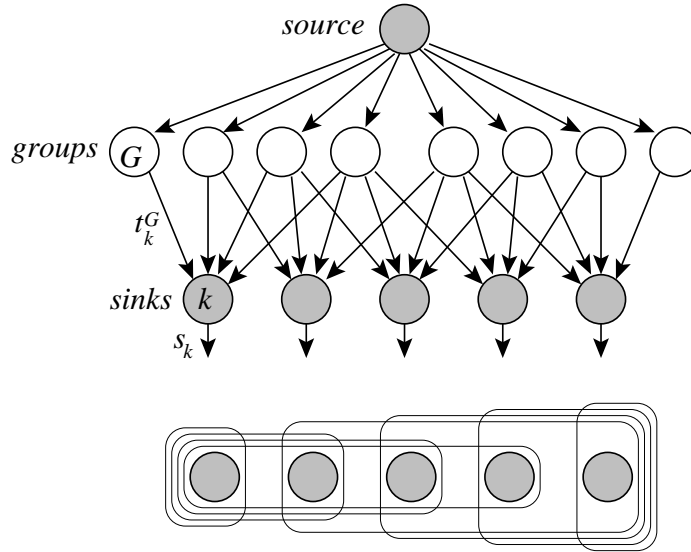
Fig. 3.4: Flow (top) and set of groups (bottom) for sequences. When these groups have unit weights (i.e., $D(G) = 1$ for these groups and zero for all others), then the submodular function $F(A)$ is equal (up to constants) to the length of the range of A (i.e., the distance beween the rightmost element of A and the leftmost element of A). When applied to sparsity-inducing norms, this leads to supports which are contiguous segments (see applications in [78]).

of ancestors of a given set A (note that, as shown in Section 4.3, the stable sets of this set-function are exactly the ones described above).

This hierarchical rule is exactly respected when using the family of groups displayed on Figure 3.5. The corresponding penalty was first used in [140]; one of it simplest instance in the context of regression is the sparse group Lasso [129, 48]; it has found numerous applications, for instance, wavelet-based denoising [140, 10, 68, 77], hierarchical dictionary learning for both topic modelling and image restoration [76, 77], log-linear models for the selection of potential orders [122], bioin-

formatics, to exploit the tree structure of gene networks for multi-task regression [82], and multi-scale mining of fMRI data for the prediction of simple cognitive tasks [75]. See also Section 9.3 for an application to non-parametric estimation with a wavelet basis.

- **Extensions**: Possible choices for the sets of groups (and thus the set functions) are not limited to the aforementioned examples; more complicated topologies can be considered, for example three-dimensional spaces discretized in cubes or spherical volumes discretized in slices (see an application to neuroimaging by [134]), and more complicated hierarchical structures based on directed acyclic graphs can be encoded as further developed in [5] to perform non-linear variable selection.

Covers vs. covers. Set covers also classically occur in the context of submodular function maximization, where the goal is, given certain subsets of V, to find the least number of these that completely cover V. Note that the main difference is that in the context of set covers considered here, the cover is considered on a potentially different set W than V, and each element of V indexes a subset of W.

3.4 Flows

Following [98], we can obtain a family of non-decreasing submodular set-functions (which include set covers) from multi-sink multi-source networks. We define a weight function on a set W, which includes a set S of sources and a set V of sinks (which will be the set on which the submodular function will be defined). We assume that we are given capacities, i.e., a function c from $W \times W$ to $\mathbb{R}_+$. For all functions $\varphi : W \times W \to \mathbb{R}$, we use the notation $\varphi(A, B) = \sum_{k \in A, \ j \in B} \varphi(k, j)$.

A flow is a function $\varphi : W \times W \to \mathbb{R}_+$ such that (a) $\varphi \leqslant c$ for all arcs, (b) for all $w \in W \backslash (S \cup V)$, the net-flow at w, i.e., $\varphi(W, \{w\}) - \varphi(\{w\}, W)$, is null, (c) for all sources $s \in S$, the net-flow at s is non-positive, i.e., $\varphi(W, \{s\}) - \varphi(\{s\}, W) \leqslant 0$, (d) for all sinks $t \in V$, the net-flow at t is non-negative, i.e., $\varphi(W, \{t\}) - \varphi(\{t\}, W) \geqslant 0$. We denote
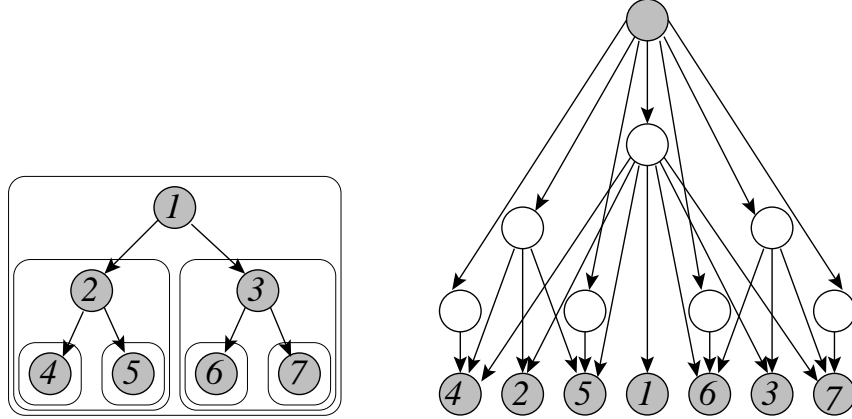
Fig. 3.5: Left: Groups corresponding to a hierarchy. Right: network flow interpretation of same submodular function (see Section 3.4). When these groups have unit weights (i.e., $D(G) = 1$ for these groups and zero for all others), then the submodular function $F(A)$ is equal to the cardinality of the union of all ancestors of A. When applied to sparsity-inducing norms, this leads to supports that select a variable only after all of its ancestors have been selected (see applications in [76]).

by $\mathcal{F}$ the set of flows.

For $A \subset V$ (the set of sinks), we define

$$F(A) = \max_{\varphi \in \mathcal{F}} \; \varphi(W, A) - \varphi(A, W),$$

which is the maximal net-flow getting out of A. From the max-flow/min-cut theorem (see, e.g., [29]), we have immediately that

$$F(A) = \min_{X \in W, \; S \subset X, \; A \subset W \setminus X} c(X, W \setminus X).$$

One then obtain that F is submodular (as the partial minimization of a cut function, see Prop. B.4) and non-decreasing by construction. One particularity is that for this type of submodular non-decreasing functions, we have an explicit description of the intersection of the positive orthant and the submodular polyhedron (potentially simpler than through the supporting hyperplanes $\{s(A) = F(A)\}$). Indeed,

$s \in \mathbb{R}_+^p$ belongs to $P(F)$ if and only if, there exists a flow $\varphi \in \mathcal{F}$ such that for all $k \in V$, $s_k = \varphi(W, \{k\}) - \varphi(\{k\}, W)$ is the net-flow getting out of k.

Similarly to other cut-derived functions, there are dedicated algorithms for proximal methods and submodular minimization [63]. See also Section 6.1 for a general divide-and-conquer strategy for solving separable optimization problems based on a sequence of submodular function minimization problems (here, min cut/max flow problems).

Flow interpretation of set-covers. Following [95], we now show that the submodular functions defined in this section includes the ones defined in Section 3.3. Indeed, consider a non-negative function $D : 2^V \to \mathbb{R}_+$, and define $F(A) = \sum_{G \subset V, \, G \cap A \neq \varnothing} D(G)$. The Lovász extension may be written as, for all $w \in \mathbb{R}_+^p$ (introducing variables t^G in a scaled simplex reduced to variables indexed by G):

$$
\begin{aligned}
f(w) &= \sum_{G \subset V} D(G) \max_{k \in G} w_k \\
&= \sum_{G \subset V} \max_{t^G \in \mathbb{R}_+^p, \, t^G_{V \setminus G} = 0, \, t^G(G) = D(G)} w^\top t^G \\
&= \max_{t^G \in \mathbb{R}_+^p, \, t^G_{V \setminus G} = 0, \, t^G(G) = D(G), \, G \subset V} \sum_{G \subset V} w^\top t^G \\
&= \max_{t^G \in \mathbb{R}_+^p, \, t^G_{V \setminus G} = 0, \, t^G(G) = D(G), \, G \subset V} \sum_{k \in V} \left(\sum_{G \subset V, \, G \ni k} t^G_k \right) w_k.
\end{aligned}
$$

Because of the representation of f as a maximum of linear functions shown in Prop. 2.2, $s \in P(F) \cap \mathbb{R}_+^p$, if and only there exists $t^G \in \mathbb{R}_+^p$, $t^G_{V \setminus G} = 0$, $t^G(G) = D(G)$ for all $G \subset V$, such that for all $k \in V$, $s_k = \sum_{G \subset V, \, G \ni k} t^G_k$. This can be given a network flow interpretation on the graph composed of a single source, one node per subset $G \subset V$ such that $D(G) > 0$, and the sink set V. The source is connected to all subsets G, with capacity $D(G)$, and each subset is connected to the variables it contains, with infinite capacity. In this representation, t^G_k is the flow from node corresponding to G, to the node corresponding to the sink node k; and $s_k = \sum_{G \subset V} t^G_k$ is the net-flow in the sink k. Thus, $s \in P(F) \cap \mathbb{R}_+^p$ if and only if, there exists a flow in this graph so

that the net-flow getting out of k is s_k, which corresponds exactly to a network flow submodular function.

We give examples of such networks in Figure 3.3 and Figure 3.4. This reinterpretation allows the use of fast algorithms for proximal problems (as there exists fast algorithms for maximum flow problems). The number of nodes in the network flow is the number of groups G such that $D(G) > 0$, but this number may be reduced in some situations. See [95, 96] for more details on such graph constructions (in particular in how to reduce the number of edges in many situations).

Application to machine learning. Applications to sparsity-inducing norms (as decribed in Section 3.3) lead to applications to hierarchical dictionary learning and topic models [76], structured priors for image denoising [76, 77], background subtraction [95], and bioinformatics [71, 82]. Moreover, many submodular functions may be interpreted in terms of flows, allowing the use of fast algorithms (see, e.g., [63, 2] for more details).

3.5   Entropies

Given p random variables $X_1, \ldots, X_p$ which all take a finite number of values, we define $F(A)$ as the joint entropy of the variables $(X_k)_{k \in A}$ (see, e.g., [33]). This function is submodular because, if $A \subset B$ and $k \notin B$, $F(A \cup \{k\}) - F(A) = H(X_A, X_k) - H(X_A) = H(X_k|X_A) \geqslant H(X_k|X_B) = F(B \cup \{k\}) - F(B)$ (by the data processing inequality [32]). Moreover, its symmetrization[2] leads to the mutual information between variables indexed by A and variables indexed by $V \backslash A$.

This can be extended to any distribution by considering differential entropies. One application is for Gaussian random variables, leading to the submodularity of the function defined through $F(A) = \log \det Q_{AA}$, for some positive definite matrix $Q \in \mathbb{R}^{p \times p}$ (see further related examples in Section 3.6).

---

[2] For any submodular function F, one may defined its symmetrized version as $G(A) = F(A) + F(V \backslash A) - F(V)$, which is submodular and symmetric. See further details in Section 7.4 and Appendix B.2.

Entropies are less general than submodular functions. Entropies of discrete variables are non-increasing, non-negative submodular set-functions. However, they are more restricted than this, i.e., they satisfy other properties which are not satisfied by all submodular functions [139]. Note also that it is not known if their special structure can be fruitfully exploited to speed up certain of the algorithms presented in Section 7.

Applications to probabilistic modelling. In the context of probabilistic graphical models, entropies occur in particular in algorithms for structure learning: indeed, for directed graphical models, given the directed acyclic graph, the minimum Kullback-Leibler divergence between a given distribution and a distribution that factorizes into the graphical model may be expressed in closed form through entropies [89, 61]. Applications of submodular function optimization may be found in this context, with both maximization [105] for learning bounded-treewidth graphical model and minimization for learning naive Bayes models [86], or both (i.e., minimizing differences of submodular functions, as shown in Section 8) for discriminative learning of structure [106].

Entropies also occur in *experimental design* in Gaussian linear models [125]. Given a design matrix $X \in \mathbb{R}^{n \times p}$, assume that the vector $y \in \mathbb{R}^n$ is distributed as $Xw + \sigma\varepsilon$, where w has normal prior distribution with mean zero and covariance matrix $\sigma^2\lambda^{-1}I$, and $\varepsilon \in \mathbb{R}^n$ is a standard normal vector. The posterior distribution of w given y is normal with mean $\lambda^{-1}\sigma^2 X(\sigma^2\lambda^{-1}X^\top X + \sigma^2 I)^{-1}y$ and covariance matrix $\lambda^{-1}\sigma^2 I - \lambda^{-2}\sigma^4 X(\sigma^2\lambda^{-1}X^\top X + \sigma^2 I)^{-1}X^\top = \lambda^{-1}\sigma^2\big[I - X(X^\top X + \lambda I)^{-1}X^\top\big] = \lambda^{-1}\sigma^2\big[I - (XX^\top + \lambda I)^{-1}XX^\top\big] = \sigma^2(XX^\top + \lambda I)^{-1}$. The posterior entropy of w given y is thus equal (up to constants) to $n\log\sigma^2 - \log\det(XX^\top + \lambda I)$. If only the observations in A are observed, then the posterior entropy of w given y_A is equal to $|A|\log\sigma^2 - \log\det(X_A X_A^\top + \lambda I)$, which is supermodular because the entropy of a Gaussian random variable is the logarithm of its determinant. In experimental design, the goal is to select the set A of observations so that the posterior entropy of w given y_A is minimal (see, e.g., [43]), and is thus equivalent to maximizing a submodular function (for which

forward selection has theoretical guarantees, see Section 8.2). Note the difference with subset selection (Section 3.7) where the goal is to select columns of the design matrix instead of rows.

Application to semi-supervised clustering.   Given p data points $x_1, \ldots, x_p$ in a certain set $\mathcal{X}$, we assume that we are given a Gaussian process $(f_x)_{x \in \mathcal{X}}$. For any subset $A \subset V$, then f_{x_A} is normally distributed with mean zero and covariance matrix K_{AA} where K is the $p \times p$ kernel matrix of the p data points, i.e., $K_{ij} = k(x_i, x_j)$ where k is the kernel function associated with the Gaussian process (see, e.g., [120]). We assume a modular prior distribution on subset of the form $p(A) \propto \prod_{k \in A} \eta_k \prod_{k \notin A} (1 - \eta_k)$ (i.e., each element k has a certain prior probability η_k of being present, with all decisions being statistically independent).

Once a set A is selected, we only assume that we want to model the two parts, A and $V \backslash A$ as two *independent* Gaussian processes with covariance matrices Σ_A and $\Sigma_{V \backslash A}$. In order to maximize the likelihood under the joint Gaussian process, the best estimates are $\Sigma_A = K_{AA}$ and $\Sigma_{V \backslash A} = K_{V \backslash A, V \backslash A}$. This leads to the following negative log-likelihood

$$I(f_A, f_{V \backslash A}) - \sum_{k \in A} \log \eta_k - \sum_{k \in V \backslash A} \log(1 - \eta_k),$$

where $I(f_A, f_{V \backslash A})$ is the mutual information between two Gaussian processes (see similar reasoning in the context of independent component analysis [19]).

We thus need to minimize a modular function plus a mutual information between the variables indexed by A and the ones indexed by $V \backslash A$, which is submodular and symmetric. Thus in this Gaussian process interpretation, clustering may be cast as submodular function minimization. This probabilistic interpretation extends the minimum description length interpretation of [108] to semi-supervised clustering.

Note here that similarly to the unsupervised clustering framework of [108], the mutual information may be replaced by any symmetric submodular function, such as a cut function obtained from appropriately defined weigths. In Figure 3.6, we consider $\mathcal{X} = \mathbb{R}^2$ and sample points from a traditional distribution in semi-supervised clustering, i.e.,
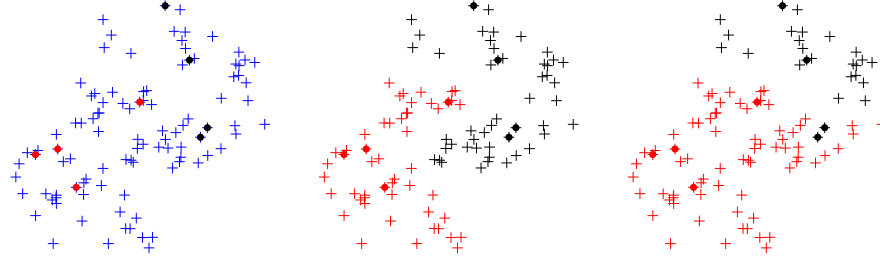
Fig. 3.6: Examples of semi-supervised clustering : (left) observations, (middle) results of the semi-supervised clustering algorithm based on submodular function minimization, with eight labelled data points, with the mutual information, (right) same procedure with the cut function.

twe "two moons" dataset. We consider 100 points and 8 randomly chosen labelled points, for which we impose $\eta_k \in \{0, 1\}$, the rest of the η_k being equal to $1/2$ (i.e, we impose a hard constraint on the labelled points to be on the correct clusters). We consider a Gaussian kernel $k(x, y) = \exp(-\alpha \|x - y\|_2^2)$, and we compare two symmetric submodular functions: mutual information and the weighted cuts obtained from the same matrix K (note that the two functions use different assumptions regarding the kernel matrix, positive definiteness for the mutual information, and pointwise positivity for the cut). As shown in Figure 3.6, by using more than second-order interactions, the mutual information is better able to capture the structure of the two clusters. This example is used as an illustration and more experiments and analysis would be needed to obtain sharper statements. In Section 9, we use this example for comparing different submodular function minimization procedures. Note that even in the case of symmetric submodular functions F, where more efficient algorithms in $O(p^3)$ for submodular function minimization (SFM) exist [117] (see also Section 7.4), the minimization of functions of the form $F(A) - z(A)$, for $z \in \mathbb{R}^p$ is provably as hard as general SFM [117].

3.6   Spectral functions of submatrices

Given a positive semidefinite matrix $Q \in \mathbb{R}^{p \times p}$ and a real-valued function h from $\mathbb{R}_+$ to $\mathbb{R}$, one may define the matrix function [54] $Q \mapsto h(Q)$ defined on positive semi-definite matrices by leaving unchanged the eigenvectors of Q and applying h to each of the eigenvalues. This leads to the expression of $\mathrm{tr}[h(Q)]$ as $\sum_{i=1}^{p} h(\lambda_i)$ where $\lambda_1, \ldots, \lambda_p$ are the (nonnegative) eigenvalues of Q [66]. We can thus define the function $F(A) = \mathrm{tr}\, h(Q_{AA})$ for $A \subset V$. Note that for Q diagonal, we exactly recover functions of modular functions considered in Section 3.1.

The concavity of h is not sufficient however in general to ensure the submodularity of F, as can be seen by generating random examples with $h(\lambda) = \lambda/(\lambda + 1)$.

Nevertheless, we know that the functions $h(\lambda) = \log(\lambda + t)$ for $t \geqslant 0$ lead to submodular functions since they lead to the entropy of a Gaussian random variable with joint covariance matrix $Q + \lambda I$. Thus, since for $\rho \in (0, 1)$, $\lambda^\rho = \frac{\rho \sin \rho \pi}{\pi} \int_0^\infty \log(1 + \lambda/t) t^{\rho-1} dt$ (see, e.g., [3]), $h(\lambda) = \lambda^\rho$ for $\rho \in (0, 1]$ is a positive linear combination of functions that lead to non-decreasing submodular set-functions. We thus obtain a non-decreasing submodular function.

This can be generalized to functions of the singular values of $X(A, B)$ where X is a rectangular matrix, by considering the fact that singular values of a matrix X are related to the eigenvalues of $\begin{pmatrix} 0 & X \\ X^\top & 0 \end{pmatrix}$ (see, e.g., [54]).

Application to machine learning (Bayesian variable selection). As shown in [6], such functions naturally appear in the context of variable selection using the Bayesian marginal likelihood (see, e.g., [52]). Indeed, given a subset A, assume that the vector $y \in \mathbb{R}^n$ is distributed as $X_A w_A + \sigma \varepsilon$, where X is a design matrix in $\mathbb{R}^{n \times p}$ and w_A a vector with support in A, and $\varepsilon \in \mathbb{R}^n$ is a standard normal vector; if a normal prior with covariance matrix $\sigma^2 \lambda^{-1} I$ is imposed on w_A, then the negative log-marginal likelihood of y given A (i.e., obtained by marginalizing

w_A), is equal to (up to constants) [126]:

$$\min_{w_A \in \mathbb{R}^{|A|}} \frac{1}{2\sigma^2} \|y - X_A w_A\|_2^2 + \frac{\lambda}{2\sigma^2} \|w_A\|^2 + \frac{1}{2} \log \det[\sigma^2 \lambda^{-1} X_A X_A^\top + \sigma^2 I].$$

Thus, in a Bayesian model selection setting, in order to find the best subset A, it is necessary to minimize with respect to w:

$$\min_{w \in \mathbb{R}^p} \frac{1}{2\sigma^2} \|y - Xw\|_2^2 + \frac{\lambda}{2\sigma^2} \|w\|^2 + \frac{1}{2} \log \det[\lambda^{-1} \sigma^2 X_{\mathrm{Supp}(w)} X_{\mathrm{Supp}(w)}^\top + \sigma^2 I],$$

which, in the framework outlined in Section 2.3, leads to the submodular function $F(A) = \frac{1}{2} \log \det[\lambda^{-1} \sigma^2 X_A X_A^\top + \sigma^2 I] = \frac{1}{2} \log \det[X_A X_A^\top + \lambda I] + \frac{n}{2} \log(\lambda^{-1} \sigma^2)$. Note also that, since we use a penalty which is the sum of a squared ℓ_2-norm and a submodular function applied to the support, then a direct convex relaxation may be obtained through reweighted least-squares formulations using the ℓ_2-relaxation of combinatorial penalties presented in Section 2.3 (see also [115]). See also related simulation experiments for random designs from the Gaussian ensemble in [6].

Note that a traditional frequentist criterion is to penalize larger subsets A by the Mallow's C_L criterion [97], which is equal to $A \mapsto \mathrm{tr}(X_A X_A^\top + \lambda I)^{-1} X_A X_A^\top$, which is *not* a submodular function.

3.7 Best subset selection

Following [36], we consider p random variables (covariates) $X_1, \ldots, X_p$, and a random response Y with unit variance, i.e., $\mathrm{var}(Y) = 1$. We consider predicting Y linearly from X. We consider $F(A) = \mathrm{var}(Y) - \mathrm{var}(Y|X_A)$. The function F is a non-decreasing function (the conditional variance of Y decreases as we observed more variables). In order to show the submodularity of F using Prop. 1.2, we compute, for all $A \subset V$, and i, j distinct elemetns in $V \backslash A$, the following quantity:

$$\begin{aligned} & F(A \cup \{j, k\}) - F(A \cup \{j\}) - F(A \cup \{k\}) + F(A) \\ = \ & [\mathrm{var}(Y|X_A, X_k) - \mathrm{var}(Y|X_A)] - [\mathrm{var}(Y|X_A, X_j, X_k) - \mathrm{var}(Y|X_A, X_j)] \\ = \ & -\mathrm{Corr}(Y, X_k|X_A)^2 + \mathrm{Corr}(Y, X_k|X_A, X_j)^2, \end{aligned}$$

using standard arguments for conditioning variances (see more details in [36]). Thus, the function is submodular if and only if the last quantity

is always non-positive, i.e., $|\mathrm{Corr}(Y, X_k | X_A, X_j)| \leqslant |\mathrm{Corr}(Y, X_k | X_A)|$, which is often referred to as the fact that the variables X_j is not a suppressor for the variable X_k given A.

Thus greedy algorithms for maximization have theoretical guarantees (see Section 8) *if* the assumption is met. Note however that the condition on suppressors is rather strong, although it can be appropriately relaxed in order to obtain more widely applicable guarantees for subset selection [37].

Subset selection as the difference of two submodular functions.   If we consider the linear model from the end of Section 3.6, then given a subset A, maximizing the log-likelihood with respect to w_A and σ^2, we obtain a negative log-likelihood of the form:

$$
\min_{w_A \in \mathbb{R}^{|A|}, \sigma^2 \in \mathbb{R}_+} \frac{n}{2} \log \sigma^2 + \frac{1}{2\sigma^2} \|y - X_A w_A\|_2^2 + \frac{\lambda}{2\sigma^2} \|w_A\|^2
$$

$$
= \min_{\sigma^2 \in \mathbb{R}_+} \frac{n}{2} \log \sigma^2 + \frac{1}{2\sigma^2} \|y\|_2^2 - \frac{1}{2\sigma^2} \operatorname{tr} y^\top X_A (X_A^\top X_A + \lambda I)^{-1} X_A^\top y
$$

$$
= \frac{n}{2} \log \frac{1}{n} y^\top (I - X_A (X_A^\top X_A + \lambda I)^{-1} X_A^\top) y + \frac{n}{2}
$$

$$
= \frac{n}{2} \log y^\top (I - X_A (X_A^\top X_A + \lambda I)^{-1} X_A^\top) y + \frac{n}{2} (1 - \log n)
$$

$$
= \frac{n}{2} \log \det \begin{pmatrix} X_A^\top X_A + \lambda I & X_A^\top y \\ y^\top X_A & y^\top y \end{pmatrix} - \frac{n}{2} \log \det(X_A^\top X_A + \lambda I) + \mathrm{cst},
$$

which is a difference of two submodular functions (see Section 8.3 for related optimization schemes). This function is non-increasing, so in order to perform variable selection, it is necessary to add another criterion, which can be the cardinality of A; or in a Bayesian setting, we can replace the above maximization with respect to w_A by a marginalization, which leads to an extra-term of the form $\frac{1}{2} \log \det(X_A^\top X_A + \lambda I)$, which does not change the type of minimization problems.

Note the difference between this formulation (aiming at minimizing a set-function directly by marginalizing out or maximizing out w) and the one from Section 3.6 which provides a convex relaxation of the maximum likelihood problem by maximizing the likelihood with respect to w.

3.8 Matroids

Given a set V, we consider a family $\mathfrak{I}$ of subsets of V such that (a) $\varnothing \in \mathfrak{I}$, (b) $I_1 \subset I_2 \in \mathfrak{I} \Rightarrow I_1 \in \mathfrak{I}$, and (c) for all $I_1, I_2 \in \mathfrak{I}$, $|I_1| < |I_2| \Rightarrow \exists k \in I_2 \backslash I_1,\ I_1 \cup \{k\} \in \mathfrak{I}$. The pair $(V, \mathfrak{I})$ is then referred to as a matroid, with $\mathfrak{I}$ its family of independent sets. Then, the rank function of the matroid, defined as $F(A) = \max_{I \subset A,\ A \in \mathfrak{I}} |I|$, is submodular.[3]

A classical example is the *graphic matroid*; it corresponds to V being an edge set of a certain graph, and $\mathfrak{I}$ being the set of subsets of edges which do not contain any cycle. The rank function $\rho(A)$ is then equal to p minus the number of connected components of the subgraph induced by A.

The other classical example is the *linear matroid*. Given a matrix M with p columns, then a set I is independent if and only if the columns indexed by I are linearly independent. The rank function $\rho(A)$ is then the rank of the columns indexed by A (this is also an instance of functions from Section 3.6 because the rank is the number of non-zero eigenvalues, and when $\rho \to 0^+$, then $\lambda^\rho \to 1_{\lambda>0}$). For more details on matroids, see, e.g., [124].

Greedy algorithm. For matroid rank functions, extreme points of the base polyhedron have components equal to zero or one (because $F(A \cup \{k\}) - F(A) \in \{0, 1\}$ for any $A \subset V$ and $k \in V$), and are incidence vectors of the maximal independent sets (maximal because of the constraint $s(V) = F(V)$). Thus, the greedy algorithm for maximizing linear functions on the base polyhedron may be used to find maximum weight maximal independent sets, where a certain weight is given to all elements of V. In this situation, the greedy algorithm is actually greedy, that it first orders the weights of each element of V in decreasing order and select elements of V following this order and skipping the elements which lead to non-independent sets.

For the graphic matroid, the base polyhedron is thus the convex

---

[3] This can be shown directly using Prop. 1.1. We first show that for any $A \subset V$, and $k \notin A$, then $F(A \cup \{k\}) - F(A) \in \{0, 1\}$ as a consequence of the property (c). Then, we only need to show that if $F(A \cup \{k\}) = F(A)$, then for all B greater than A (and that does not contain k), then $F(B \cup \{k\}) = F(B)$, which is a consequence of property (b).

hull of the incidence vectors of sets of edges which form a spanning tree, and is often referred to as the spanning tree polytope[4] [25]. The greedy algorithm is then exactly Kruskal's algorithm to find maximum weight spanning trees [29].

Minimizing matroid rank function minus a modular function.
General submodular functions may be minimized in polynomial time (see Section 7), but usually with large complexity, i.e., $O(p^6)$. For functions which are equal to the rank function of a matroid minus a modular function, then algorithms have better running-time complexities, i.e., $O(p^3)$ [34, 109].

---

[4] Note that algorithms presented in Section 6 lead to algorithms for several operations on this spanning tree polytopes, such as line searches and orthogonal projections.

4

Properties of associated polyhedra

We now study in more details submodular and base polyhedra defined in Section 1, as well as the symmetric independent polyhedron (which is the unit dual ball for the norms defined in Section 2.3). We firt review that the support functions may be computed by the greedy algorithm, and then characterize the set of maximizers of linear functions, from which we deduce a detailed facial structure of the base polytope $B(F)$ and the symmetric independence polyhedron $|P|(F)$.

4.1 Support functions

The next proposition completes Prop. 2.2 by computing the full support function of $B(F)$ and $P(F)$ (see [17, 16] for definitions of support functions), i.e., computing $\max_{s \in B(F)} w^\top s$ and $\max_{s \in P(F)} w^\top s$ for all possible w (with positive and/or negative coefficients). Note the different behaviors for $B(F)$ and $P(F)$.

Proposition 4.1. (Support functions of associated polyhedra)
Let F be a submodular function such that $F(\varnothing) = 0$. We have:
(a) for all $w \in \mathbb{R}^p$, $\max_{s \in B(F)} w^\top s = f(w)$,
(b) if $w \in \mathbb{R}^p_+$, $\max_{s \in P(F)} w^\top s = f(w)$,

(c) if there exists j such that $w_j < 0$, then $\max_{s \in P(F)} w^\top s = +\infty$,
(d) if F is non-decreasing, for all $w \in \mathbb{R}^p$, $\max_{s \in |P|(F)} w^\top s = f(|w|)$.

Proof. The only statement left to prove beyond Prop. 2.2 and Prop. 2.5 is (c): we just need to notice that $s(\lambda) = s_0 - \lambda \delta_j \in P(F)$ for $\lambda \to +\infty$ and $s_0 \in P(F)$ and that $w^\top s(\lambda) \to +\infty$. $\qquad\qquad\square$

The next proposition shows necessary and sufficient conditions for optimality in the definition of support functions. Note that Prop. 2.2 gave one example obtained from the greedy algorithm, and that we can now characterize all maximizers. Moreover, note that the maximizer is unique only when w has distinct values, and otherwise, the ordering of the components of w is not unique, and hence, the greedy algorithm may have multiple outputs (and all convex combinations of these are also solutions). The following proposition essentially shows what is exactly needed to be a maximizer. This proposition is key to deriving optimality conditions for the separable optimization problems that we consider in Section 5 and Section 6.

Proposition 4.2. (Maximizers of the support function of submodular and base polyhedra) Let F be a submodular function such that $F(\varnothing) = 0$. Let $w \in \mathbb{R}^p$, with unique values $v_1 > \cdots > v_m$, taken at sets $A_1, \ldots, A_m$ (i.e., $V = A_1 \cup \cdots \cup A_m$ and $\forall i \in \{1, \ldots, m\}$, $\forall k \in A_i$, $w_k = v_i$). Then,
(a) if $w \in (\mathbb{R}_+^*)^p$, s is optimal for $\max_{s \in P(F)} w^\top s$ if and only if for all $i = 1, \ldots, m$, $s(A_1 \cup \cdots \cup A_i) = F(A_1 \cup \cdots \cup A_i)$,
(b) s is optimal for $\max_{s \in B(F)} w^\top s$ if and only if for all $i = 1, \ldots, m$, $s(A_1 \cup \cdots \cup A_i) = F(A_1 \cup \cdots \cup A_i)$.

Proof. We first prove (a). Let $B_i = A_1 \cup \cdots \cup A_i$, for $i = 1, \ldots, m$. From the optimization problems defined in the proof of Prop. 2.2, let $\lambda_V = v_m > 0$, and $\lambda_{B_i} = v_i - v_{i+1} > 0$ for $i < m$, with all other λ_A, $A \subset V$, equal to zero. Such λ is optimal (because the dual function is equal to the primal objective $f(w)$).

Let $s \in P(F)$. We have:

$$
\begin{aligned}
\sum_{A \subset V} \lambda_A F(A) \;=\;& v_m F(V) + \sum_{i=1}^{m-1} F(B_i)(v_i - v_{i+1}) \\
=\;& v_m(F(V) - s(V)) + \sum_{i=1}^{m-1} [F(B_i) - s(B_i)](v_i - v_{i+1}) \\
& + v_m s(V) + \sum_{i=1}^{m-1} s(B_i)(v_i - v_{i+1}) \\
\geqslant\;& v_m s(V) + \sum_{i=1}^{m-1} s(B_i)(v_i - v_{i+1}) = s^\top w.
\end{aligned}
$$

Thus s is optimal, if and only if the primal objective value $s^\top w$ is equal to the optimal dual objective value $\sum_{A \subset V} \lambda_A F(A)$, and thus, if and only if there is equality in all above inequalities, hence the desired result. The proof for (b) follows the same arguments, except that we don't need to show that $s(V) = F(V)$, since this is always satisfied for $s \in B(F)$, hence we don't need $v_m > 0$.   □

Note that for (a), if $v_m = 0$ in Prop. 4.2 (i.e., we take $w \in \mathbb{R}_+^p$ and there is a w_k equal to zero), then the optimality condition is that for all $i = 1, \ldots, m-1$, $s(A_1 \cup \cdots \cup A_i) = F(A_1 \cup \cdots \cup A_i)$ (i.e., we don't need that $s(V) = F(V)$, i.e., the optimal solution is not necessarily in the base polyhedron).

4.2   Facial structure

In this section, we describe the facial structure of the base polyhedron. We first review the relevant concepts for convex polytopes.

Face lattice of a convex polytope. We quickly review the main concepts related to convex polytopes. For more details, see [56]. A convex polytope is the convex hull of a finite number of points. It may be also seen as the intersection of finitely many half-spaces (such intersections are referred to as polyhedra and are called polytopes if they are bounded). *Faces* of a polytope are sets of maximizers of $w^\top s$ for certain $w \in \mathbb{R}^p$. Faces are convex sets whose affine hulls are intersections

of the hyperplanes defining the half-spaces from the intersection of half-space representation. The dimension of a face is the dimension of its affine hull. The $(p-1)$-dimensional faces are often referred to as *facets*, while zero-dimensional faces are its *vertices*. A natural order may be defined on the set of faces, namely the inclusion order between the sets of hyperplanes defining the face. With this order, the set of faces is a distributive lattice [38], with appropriate notions of "join" (unique smallest face that contains the two faces) and "meet" (intersection of the two faces).

Dual polytope.   We now assume that we consider a polytope with zero in its interior (this can be done by projecting it onto its affine hull and translating it appropriately). The dual polytope of C is the polar set C° of the polytope C (see Appendix A). It turns out that faces of C° are in bijection with the faces of C, with vertices of C mapped to facets of C° and vice-versa. If C is represented as the convex hull of points s_i, $i \in \{1, \ldots, m\}$, then the polar of C is defined through the intersection of the half-space $\{w \in \mathbb{R}^p, \ s_i^\top w \leqslant 1\}$, for $i = 1, \ldots, m$. Analyses and algorithms related to polytopes may always be defined or looked through their dual polytopes. In our situation, we consider two polytopes, $B(F)$ for which the dual polytope is the set $\{w, f(w) \leqslant 1, w^\top 1_V = 0\}$ (see an example in Figure 2.2), and the symmetric independent polytope $|P|(F)$, whose dual polytope is the unit ball of the norm Ω defined in Section 2.3. See Figure 2.3 for examples of these polytopes, and also Section 4.3.

Faces of the base polyhedron.   Given the Prop. 4.2 that provides the maximizers of $\max_{s \in B(F)} w^\top s$, we may now give necessary and sufficient conditions for characterizing faces of the base polyhedron. We first characterize when the base polyhedron $B(F)$ has non-empty interior within the subspace $\{s(V) = F(V)\}$.

Definition 4.1. (Inseparable set) Let F be a submodular function such that $F(\varnothing) = 0$. A set $A \subset V$ is said separable if and only there is a set $B \subset A$, such that $B \neq \varnothing$, $B \neq A$ and $F(A) = F(B) + F(A \backslash B)$.

If A is non separable, A is said inseparable.

Proposition 4.3. (Full-dimensional base polyhedron) Let F be a submodular function such that $F(\varnothing) = 0$. The base polyhedron has non-empty interior in $\{s(V) = F(V)\}$ if and only if V is not separable.

Proof. If V is separable into A and $V\backslash A$, then, by submodularity of F, for all $s \in B(F)$, we must have $s(A) = F(A)$ (and thus also $F(V\backslash A) = s(V\backslash A)$) and hence the base polyhedron is included in the intersection of two affine hyperplanes, i.e., $B(F)$ does not have non-empty interior in $\{s(V) = F(V)\}$.

Since $B(F)$ is defined through supporting hyperplanes, it has non-empty interior in $\{s(V) = F(V)\}$ if it is not contained in any of the supporting hyperplances. We thus now assume that $B(F)$ is included in $\{s(A) = F(A)\}$, for A as a non-empty strict subset of V. Then $B(F)$ can be factorized in to $B(F_A) \times B(F^A)$ where F_A is the restriction of F to A and F^A the contraction of F on A (see definition and properties in Appendix B.2). Indeed, if $s \in B(F)$, then $s_A \in B(F_A)$ because $s(A) = F(A)$, and $s_{V\backslash A} \in B(F^A)$, because for $B \subset V\backslash A$, $s_{V\backslash A}(B) = s(B) = s(A \cup B) - s(A) \leqslant F(A \cup B) - F(A)$. Similarly, if $s \in B(F_A) \times B(F^A)$, then for all set $B \subset V$, $s(B) = s(A \cap B) + S((V\backslash A) \cap B) \leqslant F(A \cap B) + F(A \cup B) - F(A) \leqslant F(B)$ by submodularity, and $s(A) = F(A)$.

This shows that $f(w) = f_A(w_A) + f^A(w_{V\backslash A})$, which implies that $F(V) = F(A) + F(V\backslash A)$, when applied to $w = 1_V$, i.e., V is separable. □

We can now detail the facial structure of the base polyhedron, which will be dual to the one of the polyhedron defined by $\{w \in \mathbb{R}^p,\ f(w) \leqslant 1, w^\top 1_V = 0\}$ (i.e., the sub-level set of the Lovász extension projected on a subspace of dimension $p - 1$). As the base polyhedron $B(F)$ is a polytope in dimension $p - 1$ (because it is bounded and contained in the affine hyperplane $\{s(V) = F(V)\}$), one can define a set of *faces*. As described earlier, faces are the intersections of the polyhedron $B(F)$ with any of its supporting hyperplanes. Supporting hyperplanes are themselves defined as the hyperplanes $\{s(A) = F(A)\}$ for $A \subset V$.

From Prop. 4.2, faces are obtained as the intersection of $B(F)$ with $s(A_1 \cup \cdots \cup A_i) = F(A_1 \cup \cdots \cup A_i)$ for an ordered partition of V. Together with Prop. 4.3, we can now provide characterization of the faces of $B(F)$. See more details on the facial structure of $B(F)$ in [49].

Since the facial structure is invariant by translation, as done at the end of Section 2.1 we may translate $B(F)$ by a certain vector $t \in B(F)$, so that F may be taken to be non-negative and such that $F(V) = 0$ (as done at the end of Section 2.1), which we now assume.

Proposition 4.4. (Faces of the base polyhedron) Let $A_1 \cup \cdots \cup A_m$ be an ordered partition of V, such that for all $j \in \{1, \ldots, m\}$, A_j is inseparable for the function $G_j : B \mapsto F(A_1 \cup \cdots \cup A_{j-1} \cup B) - F(A_1 \cup \cdots \cup A_{j-1})$ defined on subsets of A_j, then the set of bases $s \in B(F)$ such that for all $j \in \{1, \ldots, m\}$, $s(A_1 \cup \cdots \cup A_i) = F(A_1 \cup \cdots \cup A_i)$ is a face of $B(F)$ with non-empty interior in the intersection of the m hyperplanes (i.e., the affine hull of the face is exactly the intersection of these m hyperplanes). Moreover, all faces of $B(F)$ may be obtained this way.

Proof. From Prop. 4.2, all faces may be obtained with supporting hyperplanes of the form $s(A_1 \cup \cdots \cup A_i) = F(A_1 \cup \cdots \cup A_i)$, $i = 1, \ldots, m$, for a certain partition $V = A_1 \cup \cdots \cup A_m$. Hovever, among these partitions, only some of them will lead to an affine hull of full dimension m. From Prop. 4.3 applied to the submodular function G_j, this only happens if G_j has no separable sets. $\square$

Note that in the previous proposition, several ordered partitions may lead to the exact same face. The maximal number of full-dimensional faces of $B(F)$ is always less than $2^p - 2$ (number of non-trivial subsets of V), but this number may be reduced in general (see examples in Figure 2.2). Moreover, the number of extreme points may also be large, e.g., $p!$ for the submodular function $A \mapsto -|A|^2$ (leading to the permutohedron [49]).

Note that the previous discussion implies that we have also a characterization of the faces of the dual polytope $\mathcal{U} = \{w \in \mathbb{R}^p, f(w) \leqslant 1, w^\top 1_V = 0\}$ (note that because we have assumed that F is non-negative and $F(V) = 0$, then f is pointwise positive and satisfies

$f(1_V) = 0$). In particular, the faces of $\mathcal{U}$ are obtained from the faces of $B(F)$ through the relationship defined in Prop. 4.2: that is, given a face of $B(F)$, and all the ordered partitions of Prop. 4.4 which lead to it, the corresponding face of $\mathcal{U}$ is the closure of the union of all w that satisfies the level set constraints imposed by the different ordered partitions. As shown in [7], the different ordered partitions all share the same elements but with a different order, thus inducing a set of partial constraints between the ordering of the m values w is allowed to take.

An important aspect is that the the separability criterion in Prop. 4.4 forbids some level sets from being characteritistic of a face. For example, for cuts in an undirected graph, this shows that all level sets within a face must be connected components of the graph. When the Lovász extension is used as a constraint for a smooth optimization problems, the solution has to happen in one of the faces. Moreover, within this face, all other affine constraints are very unlikely to happen, unless the smooth function has some specific directions of zero gradient (unlikely with random data, for some sharper statements, see [7]). Thus, when using the Lovász as a regularizer, only certain level sets are likely to happen, and in the context of cut functions, only connected sets are allowed, which is one of the justifications behind using the total variation.

4.3 Symmetric independence polyhedron

We now assume that the function F is non-decreasing, and consider the symmetric independence polyhedron $|P|(F)$, which is the unit ball of the dual norm Ω^* defined in Section 2.3. This polytope is dual to the unit ball of Ω, and it it thus of interest to characterize the facial structure of $|P|(F)$. We need the additional notion of stable sets.

Definition 4.2. (Stable sets) A set $A \subset V$ is said stable for a submodular function F, if $A \subset B$ and $A \neq B$ implies that $F(A) < F(B)$.

We first derive the same proposition than Prop. 4.2 for the symmetric independence polyhedron.

Proposition 4.5. (Maximizers of the support function of symmetric independence polyhedron) Let F be a non-decreasing submodular function such that $F(\varnothing) = 0$. Let $w \in \mathbb{R}_*^p$, with unique values for $|w|$, $v_1 > \cdots > v_m > 0$, taken at sets $A_1, \ldots, A_m$. Then s is optimal for $\max_{s \in |P|(F)} w^\top s$ if and only if for all $i = 1, \ldots, m$, $|s|(A_1 \cup \cdots \cup A_i) = F(A_1 \cup \cdots \cup A_i)$, and w and s have the same signs.

Proof. The proof follows the same arguments than for Prop. 4.2.   □

Note that in the previous proposition, if $v_m = 0$ in Prop. 4.2 (i.e., we take $w \in \mathbb{R}^p$ with some zero components, then the optimality condition is that for all $i = 1, \ldots, m - 1$, $|s|(A_1 \cup \cdots \cup A_i) = F(A_1 \cup \cdots \cup A_i)$ (i.e., we don't need that $|s|(V) = F(V)$, that is, the optimal solution is not necessarily in the base polyhedron). Moreover, the value of s_k when $w_k = 0$ is irrelevant (given that $|s| \in P(F)$).

We can now derive a characterization of the faces of $|P|(F)$.

Proposition 4.6. (Faces of the symmetric independence polyhedron) Let C be a stable set and Let $A_1 \cup \cdots \cup A_m$ be an ordered partition of C, such that for all $j \in \{1, \ldots, m\}$, A_j is inseparable for the function $G_j : B \mapsto F(A_1 \cup \cdots \cup A_{j-1} \cup B) - F(A_1 \cup \cdots \cup A_{j-1})$ defined on subsets of A_j, and $\varepsilon \in \{-1, 1\}^C$, then the set of bases $s \in B(F)$ such that for all $j \in \{1, \ldots, m\}$, $(\varepsilon \circ s)(A_1 \cup \cdots \cup A_i) = F(A_1 \cup \cdots \cup A_i)$ is a face of $|P|(F)$ with non-empty interior in the intersection of the m hyperplanes. Moreover, all faces of $|P|(F)$ may be obtained this way.

Proof. The proof follows the same structure than for Prop. 4.4, but by applying Prop. 4.5. The requirement for stability, comes from the fact that if C is not stable, then if D is a larger set such that $F(D) = F(C)$, we have the additional constraint $(s \circ \varepsilon)(D \backslash C) = 0$.   □

The last proposition has interesting consequences for the use of submodular functions for defining sparsity-inducing norms. Indeed, the faces of the unit-ball of Ω are dual to the ones of the dual ball of Ω^* (which is exactly $|P|(F)$). Moreover, as a consequence of Prop. 4.5, the set C in Prop. 4.6 corresponds to the non-zero elements of w in a

face of the unit-ball of Ω. This implies that all faces of the unit ball of Ω will only impose non-zero patterns which are stable sets. Note here the relationship with $w \mapsto F(\mathrm{Supp}(w))$, which would share the same property; that is, when this function is used to regularize a continuous objective function, then a stable set is always solution of the problem, as augmenting unstable sets does not increase the value of F, but can only increase the minimal value of the continuous objective function because of an extra variable to optimize upon.

However, the faces of $|P|(F)$ are not all related to non-zero patterns, and, as before, and as shown in Figure 2.3, there are additional singularities, which may come as desired or undesired (see [115]).

Stable inseparable sets. We end the description of the structure of $|P|(F)$ by noting that among the $2^p - 1$ constraints of the form $\|s_A\|_1 \leqslant F(A)$ defining it, we may restrict the sets A to be stable and inseparable. Indeed, if $\|s_A\|_1 \leqslant F(A)$ for all stable and inseparable sets A, then if B is not stable, then we may consider the smallest enclosing stable set (these are stable by intersection, hence the possibility of defining such smallest enclosing stable set) C, and we have $\|s_B\|_1 \leqslant \|s_C\|_1$, and $F(B) = F(C)$. We thus need to show that $\|s_C\|_1 \leqslant F(C)$ only for stable sets. If the set C is separable into $C = D_1 \cup \cdots \cup D_m$, where all D_i, $i = 1, \ldots, m$ are separable, they must all be stable (otherwise C would not be), and thus we have $\|s_C\|_1 = \|s_{D_1}\|_1 + \cdots + \|s_{D_m}\|_1 \leqslant F(D_1) + \cdots + F(D_m) = F(C)$.

5

Separable optimization problems - Analysis

In this section, we consider separable convex functions and the minimization of such functions penalized by the Lovász extension of a submodular function. When the separable functions are all quadratic functions, those problems are often referred to as *proximal problems* and are often used as inner loops in convex optimization problems regularized by the Lovász extension (see a brief introduction in Section 5.1 and, e.g., [28, 8] and references therein). In this section, we consider relationships between separable optimization problems and general submodular minimization problems, and focus on a detailed analysis of the equivalent between these; for corresponding algorithms, see Section 6.

5.1 Convex optimization with proximal methods

In this section, we briefly review *proximal methods* which are convex optimization methods particularly suited to the norms we have defined. They essentially allow to solve the problem regularized with a new norm at low implementation and computational costs. For a more complete presentation of optimization techniques adapted to sparsity-inducing norms, see [8]. Proximal-gradient methods constitute a class of first-

order techniques typically designed to solve problems of the following form [113, 11, 28]:

$$\min_{w\in\mathbb{R}^p} g(w) + h(w), \qquad (5.1)$$

where g is smooth. They take advantage of the structure of Eq. (5.1) as the sum of two convex terms, only one of which is assumed smooth. Thus, we will typically assume that g is differentiable (and in our situation in Eq. (2.6), that the loss function ℓ is convex and differentiable), with Lipschitz-continuous gradients (such as the logistic or square loss), while h will only be assumed convex.

Proximal methods have become increasingly popular over the past few years, both in the signal processing (see, e.g., [12, 137, 28] and numerous references therein) and in the machine learning communities (see, e.g., [8] and references therein). In a broad sense, these methods can be described as providing a natural extension of gradient-based techniques when the objective function to minimize has a non-smooth part. Proximal methods are iterative procedures. Their basic principle is to linearize, at each iteration, the function g around the current estimate $\hat{w}$, and to update this estimate as the (unique, by strong convexity) solution of the following *proximal problem*:

$$\min_{w\in\mathbb{R}^p} \left[f(\hat{w}) + (w - \hat{w})^\top f'(\hat{w}) + \lambda h(w) + \frac{L}{2}\|w - \hat{w}\|_2^2 \right]. \qquad (5.2)$$

The role of the added quadratic term is to keep the update in a neighborhood of $\hat{w}$ where f stays close to its current linear approximation; $L>0$ is a parameter which is an upper bound on the Lipschitz constant of the gradient f'.

Provided that we can solve efficiently the proximal problem in Eq. (5.2), this first iterative scheme constitutes a simple way of solving problem in Eq. (5.1). It appears under various names in the literature: proximal-gradient techniques [113], forward-backward splitting methods [28], and iterative shrinkage-thresholding algorithm [11]. Furthermore, it is possible to guarantee convergence rates for the function values [113, 11], and after t iterations, the precision be shown to be of order $O(1/t)$, which should contrasted with rates for the subgradient case, that are rather $O(1/\sqrt{t})$.

This first iterative scheme can actually be extended to "accelerated" versions [113, 11]. In that case, the update is not taken to be exactly the result from Eq. (5.2); instead, it is obtained as the solution of the proximal problem applied to a well-chosen linear combination of the previous estimates. In that case, the function values converge to the optimum with a rate of $O(1/t^2)$, where t is the iteration number. From [112], we know that this rate is optimal within the class of first-order techniques; in other words, accelerated proximal-gradient methods can be as fast as without non-smooth component.

We have so far given an overview of proximal methods, without specifying how we precisely handle its core part, namely the computation of the proximal problem, as defined in Eq. (5.2).

Proximal Problem.   We first rewrite problem in Eq. (5.2) as

$$\min_{w \in \mathbb{R}^p} \frac{1}{2} \left\| w - \left(\hat{w} - \frac{1}{L} f'(\hat{w}) \right) \right\|_2^2 + \frac{\lambda}{L} h(w).$$

Under this form, we can readily observe that when $\lambda = 0$, the solution of the proximal problem is identical to the standard gradient update rule. The problem above can be more generally viewed as an instance of the *proximal operator* [100] associated with λh:

$$\text{Prox}_{\lambda h} : u \in \mathbb{R}^p \mapsto \underset{v \in \mathbb{R}^p}{\text{argmin}} \, \frac{1}{2} \|u - v\|_2^2 + \lambda h(v).$$

For many choices of regularizers h, the proximal problem has a closed-form solution, which makes proximal methods particularly efficient. If Ω is chosen to be the ℓ_1-norm, the proximal operator is simply the soft-thresholding operator applied elementwise [39]. In this paper the function h will be either the Lovász extension f of the submodular function F, or, for non-decreasing submodular functions, the norm Ω defined in Section 2.3. In both cases, the proximal operator is exactly one of the separable optimization problems we consider in this section.

5.2   Optimality conditions for base polyhedra

Throughout this section, we make the simplifying assumption that the problem is strictly convex and differentiable (but not necessarily quadratic) and such that the derivatives are unbounded, but sharp

statements could also be made in the general case. The next proposition shows that by convex strong duality (see Appendix A), it is equivalent to the maximization of a separable concave function over the base polyhedron.

Proposition 5.1. (Dual of proximal optimization problem)
Let $\psi_1, \ldots, \psi_p$ be p continuously differentiable strictly convex functions on $\mathbb{R}$ such that for all $j \in V$, functions ψ_j are such that $\sup_{\alpha \in \mathbb{R}} \psi'_j(\alpha) = +\infty$ and $\inf_{\alpha \in \mathbb{R}} \psi'_j(\alpha) = -\infty$. Denote $\psi_1^*, \ldots, \psi_p^*$ their Fenchel-conjugates (which then have full domain). The two following optimization problems are dual of each other:

$$\min_{w \in \mathbb{R}^p} f(w) + \sum_{j=1}^{p} \psi_j(w_j), \tag{5.3}$$

$$\max_{s \in B(F)} -\sum_{j=1}^{p} \psi_j^*(-s_j). \tag{5.4}$$

The pair (w, s) is optimal if and only if (a) $s_k = -\psi'_k(w_k)$ for all $k \in \{1, \ldots, p\}$, and (b) $s \in B(F)$ is optimal for the maximization of $w^\top s$ over $s \in B(F)$ (see Prop. 4.2 for optimality conditions).

Proof. We have assumed that for all $j \in V$, functions ψ_j are such that $\sup_{\alpha \in \mathbb{R}} \psi'_j(\alpha) = +\infty$ and $\inf_{\alpha \in \mathbb{R}} \psi'_j(\alpha) = -\infty$. This implies that the Fenchel-conjugates ψ_j^* (which are already differentiable because of the strict convexity of ψ_j [16]) are defined and finite on $\mathbb{R}$, as well as strictly convex. We have (since strong duality applies because of Fenchel duality, see Appendix A.2 and [16]):

$$
\begin{aligned}
\min_{w \in \mathbb{R}^p} f(w) + \sum_{j=1}^{p} \psi_i(w_j) &= \min_{w \in \mathbb{R}^p} \max_{s \in B(F)} w^\top s + \sum_{j=1}^{p} \psi_j(w_j) \\
&= \max_{s \in B(F)} \min_{w \in \mathbb{R}^p} w^\top s + \sum_{j=1}^{p} \psi_j(w_j) \\
&= \max_{s \in B(F)} -\sum_{j=1}^{p} \psi_j^*(-s_j),
\end{aligned}
$$

where ψ_j^* is the Fenchel-conjugate of ψ_j (which may in general have a domain strictly included in $\mathbb{R}$). Thus the separably penalized problem defined in Eq. (5.3) is equivalent to a separable maximization over the base polyhedron (i.e., Eq. (5.4)). Moreover, the unique optimal s for Eq. (5.4) and the unique optimal w for Eq. (5.3) are related through $s_j = -\psi_j'(w_j)$ for all $j \in V$. $\square$

5.3   Equivalence with submodular function minimization

Following [21], we also consider a sequence of set optimization problems, parameterized by $\alpha \in \mathbb{R}$:

$$\min_{A \subset V} F(A) + \sum_{j \in A} \psi_j'(\alpha). \tag{5.5}$$

We denote by A^α any minimizer of Eq. (5.5). Note that A^α is a minimizer of a submodular function $F + \psi'(\alpha)$, where $\psi'(\alpha) \in \mathbb{R}^p$ is the vector of components $\psi_k'(\alpha)$, $k \in \{1, \ldots, p\}$.

   The key property we highlight in this section is that, as shown in [21], solving Eq. (5.3), which is a convex optimization problem, is equivalent to solving Eq. (5.5) for all possible $\alpha \in \mathbb{R}$, which are submodular optimization problems. We first show a monotonicity property of solutions of Eq. (5.5) (following [21]).

Proposition 5.2. (Monotonicity of solutions) Under the same assumptions than in Prop. 5.1, if $\alpha < \beta$, then any solutions A^α and A^β of Eq. (5.5) for α and β satisfy $A^\beta \subset A^\alpha$.

Proof. We have, by optimality of A^α and A^β:

$$F(A^\alpha) + \sum_{j \in A^\alpha} \psi_j'(\alpha) \;\leqslant\; F(A^\alpha \cup A^\beta) + \sum_{j \in A^\alpha \cup A^\beta} \psi_j'(\alpha)$$

$$F(A^\beta) + \sum_{j \in A^\beta} \psi_j'(\beta) \;\leqslant\; F(A^\alpha \cap A^\beta) + \sum_{j \in A^\alpha \cap A^\beta} \psi_j'(\beta),$$

and by summing the two inequalities and using the submodularity of F,

$$\sum_{j \in A^\alpha} \psi_j'(\alpha) + \sum_{j \in A^\beta} \psi_j'(\beta) \leqslant \sum_{j \in A^\alpha \cup A^\beta} \psi_j'(\alpha) + \sum_{j \in A^\alpha \cap A^\beta} \psi_j'(\beta),$$

which is equivalent to $\sum_{j \in A^\beta \setminus A^\alpha} (\psi'_j(\beta) - \psi'_j(\alpha)) \leqslant 0$, which implies, since for all $j \in V$, $\psi'_j(\beta) > \psi'_j(\alpha)$ (because of strict convexity), that $A^\beta \setminus A^\alpha = \varnothing$. □

The next proposition shows that we can obtain the unique solution of Eq. (5.3) from all solutions of Eq. (5.5).

Proposition 5.3. (Proximal problem from submodular function minimizations) Under the same assumptions than in Prop. 5.1, given any solutions A^α of problems in Eq. (5.5), for all $\alpha \in \mathbb{R}$, we define the vector $u \in \mathbb{R}^p$ as

$$u_j = \sup(\{\alpha \in \mathbb{R}, \ j \in A^\alpha\}).$$

Then u is the unique solution of the convex optimization problem in Eq. (5.3).

Proof. Because $\inf_{\alpha \in \mathbb{R}} \psi'_j(\alpha) = -\infty$, for α small enough, we must have $A^\alpha = V$, and thus u_j is well-defined and finite for all $j \in V$.

If $\alpha > u_j$, then, by definition of u_j, $j \notin A^\alpha$. This implies that $A^\alpha \subset \{j \in V, u_j \geqslant \alpha\} = \{u \geqslant \alpha\}$. Moreover, if $u_j > \alpha$, there exists $\beta \in (\alpha, u_j)$ such that $j \in A^\beta$. By the monotonicity property of Prop. 5.2, A^β is included in A^α. This implies $\{u > \alpha\} \subset A^\alpha$.

We have for all $w \in \mathbb{R}^p$, and β less than the smallest of $(w_j)_-$ and

the smallest of $(u_j)_-$:

$$f(u) + \sum_{j=1}^{p} \psi_j(u_j)$$

$$= \int_0^\infty F(\{u \geqslant \alpha\})d\alpha + \int_\beta^0 (F(\{u \geqslant \alpha\}) - F(V))d\alpha$$

$$+ \sum_{j=1}^{p} \left\{ \int_\beta^{u_j} \psi_j'(\alpha)d\alpha + \psi_j(\beta) \right\}$$

$$= C + \int_\beta^\infty \left[F(\{u \geqslant \alpha\}) + \sum_{j=1}^{p} (1_{w \geqslant \alpha})_j \psi_j'(\alpha) \right] d\alpha$$

$$\text{with } C = \int_0^\beta F(V)d\alpha + \sum_{j=1}^{p} \psi_j(\beta)$$

$$\leqslant C + \int_\beta^\infty \left[F(\{w \geqslant \alpha\}) + \sum_{j=1}^{p} (1_{w \geqslant \alpha})_j \psi_j'(\alpha) \right] d\alpha \text{ by optimality of } A^\alpha,$$

$$= f(w) + \sum_{j=1}^{p} \psi_j(w_j).$$

This shows that u is the unique optimum of problem in Eq. (5.3). □

From the previous proposition, we also get the following corollary, i.e., all solutions of Eq. (5.5) may be obtained from the unique solution of Eq. (5.3). Note that we immediately get the maximal and minimal minimizers, but that there is no general characterization of the set of minimizers (which is a lattice because of Prop. 7.1).

Proposition 5.4. (Submodular function minimizations from proximal problem) Under the same assumptions than in Prop. 5.1, if u is the unique minimizer of Eq. (5.3), then for all $\alpha \in \mathbb{R}$, the minimal minimizer of Eq. (5.5) is $\{u > \alpha\}$ and the maximal minimizer is $\{u \geqslant \alpha\}$, that is, for any minimizers A^α, we have $\{u > \alpha\} \subset A^\alpha \subset \{u \geqslant \alpha\}$.

Proof. From the definition of the supremum in Prop. 5.3, then we immediately obtain that $\{u > \alpha\} \subset A^\alpha \subset \{u \geqslant \alpha\}$ for any minimizer A^α. Moreover, if α is not a value taken by some u_j, $j \in V$, then this

defines uniquely A^α. If not, then we simply need to show that $\{u \geqslant \alpha\}$ and $\{u > \alpha\}$ are indeed maximizers, which can be obtained by taking limits of A^β when β tends to α from below and above. $\qquad\square$

Duality gap. We can further show that for any $s \in B(F)$ and $w \in \mathbb{R}^p$,

$$f(w) - w^\top s + \sum_{j=1}^{p} \left\{ \psi_j(w_j) + \psi_j^*(-s_j) + w_j s_j \right\} \qquad (5.6)$$

$$= \int_{-\infty}^{+\infty} \left\{ (F + \psi'(\alpha))(\{w \geqslant \alpha\}) - (s + \psi'(\alpha))_-(V) \right\} d\alpha.$$

Thus, the duality gap of the separable optimization problem in Prop. 5.1, may be written as the integral of a function of α. It turns out that, as a consequence of Prop. 7.3 (Section 7), this function of α is the duality gap for the minimization of the submodular function $F + \psi'(\alpha)$. Thus, we obtain another direct proof of the previous propositions. Eq. (5.6) will be particularly useful when relating approximat solution of the convex optimization problem to approximate solution of the combinatorial optimization problem of minimizing a submodular function (see Section 7.5).

5.4 Quadratic optimization problems

When specializing Prop. 5.1 and 5.4 to quadratic functions, we obtain the following corollary, which shows how to obtain minimizers of $F(A) + \lambda|A|$ for all possible $\lambda \in \mathbb{R}$ from a single convex optimization problem:

Proposition 5.5. (Quadratic optimization problem) Let F be a submodular function and $w \in \mathbb{R}^p$ the unique minimizer of $w \mapsto f(w) + \frac{1}{2}\|w\|_2^2$. Then:
(a) $s = -w$ is the point in $B(F)$ with minimum ℓ_2-norm,
(b) For all $\lambda \in \mathbb{R}$, the maximal minimizer of $A \mapsto F(A) + \lambda|A|$ is $\{w \geqslant -\lambda\}$ and the minimal minimizer of F is $\{w > -\lambda\}$.

One of the consequences of the last proposition is that some of the solutions to the problem of minimizing a submodular function sub-

ject to cardinality constraints may be obtained directly from the solution of the quadratic separable optimization problems (see more details in [104]).

Primal candidates from dual candidates.   From Prop. 5.5, given the *optimal* solution s of $\max_{s \in B(F)} -\frac{1}{2}\|s\|_2^2$, we obtain the *optimal* solution $w = -s$ of $\min_{w \in \mathbb{R}^p} f(w) + \frac{1}{2}\|w\|_2^2$. However, when using approximate algorithms such as the ones presented in Section 6, one may actually get only an *approximate* dual solution s, and in this case, one can improve on the natural candidate primal solution $w = -s$. Indeed, assume that the components of s are sorted in increasing order $s_{j_1} \leqslant \cdots \leqslant s_{j_p}$, and denote $t \in B(F)$ the vector defined by $t_{j_k} = F(\{j_1, \ldots, j_k\}) - F(\{j_1, \ldots, j_{k-1}\})$. Then we have $f(-s) = t^\top(-s)$, and for any w such that $w_{j_1} \geqslant \cdots \geqslant w_{j_p}$, we have $f(w) = w^\top t$. Thus, by minimizing $w^\top t + \frac{1}{2}\|w\|_2^2$ subject to this constraint, we improve on the choice $w = -s$. Note that this is exactly an isotonic regression problem with total order, which can be solved simply and efficiently in $O(p)$ by the "pool adjacent violators" algorithm (see, e.g., [14]). In Section 9, we show that this leads to much improved approximate duality gaps.

Additional properties.   Proximal problems with the square loss exhibit further interesting properties. For example, when considering problems of the form $\min_{w \in \mathbb{R}^p} \lambda f(w) + \frac{1}{2}\|w - z\|_2^2$, for varying λ, some set-functions (such as the cut in the chain graph) leads to an agglomerative path, i.e., as λ increases, components of the unique optimal solutions cluster together and never get separated [7].

Also, one may add an additional ℓ_1-norm penalty to the regularized quadratic separable problem defined above, and it is shown in [7] that, for any submodular function, the solution of the optimization problem may be obtained by soft-thresholding the result of the original proximal problem (note that this is not true for all separable optimization problems).

5.5 Separable problems on other polyhedra

We now show how to minimize a separable convex function on the submodular polyhedron or the symmetric independent polyhedron (rather than on the base polyhedron). We first show the following proposition for the submodular polyhedron of any submodular function (non necessarily non-decreasing), which relates the unrestricted proximal problem with the proximal problem restricted to $\mathbb{R}^p_+$.

Proposition 5.6. (Separable optimization on the submodular polyhedron) Assume that F is submodular. Let ψ_j, $j = 1,\ldots,p$ be p convex functions such that ψ_j^* is defined and finite on $\mathbb{R}$. Let (v,t) be a primal-dual optimal pair for the problem

$$\min_{v \in \mathbb{R}^p} f(v) + \sum_{k \in V} \psi_k(v_k) = \max_{t \in B(F)} - \sum_{k \in V} \psi_k^*(-t_k).$$

For $k \in V$, let s_k be a maximizer of $-\psi_k^*(-s_k)$ on $(-\infty, t_k]$. Define $w = v_+$. Then (w,s) is a primal-dual optimal pair for the problem

$$\min_{w \in \mathbb{R}^p_+} f(w) + \sum_{k \in V} \psi_k(w_k) = \max_{s \in P(F)} - \sum_{k \in V} \psi_k^*(-s_k).$$

Proof. The pair (w,s) is optimal if and only if (a) $w_k s_k + \psi_k(w_k) + \psi_k^*(-s_k) = 0$, i.e., (w_k, s_k) is a Fenchel-dual pair for ψ_k, and (b) $f(w) = s^\top w$. The first statement (a) is true by construction (indeed, if $s_k = t_k$, then this is a consequence of optimality for the first problem, and if $s_k < t_k$, then $w_k = (\psi_k^*)'(-s_k) = 0$).

For the second statement (b), notice that s is obtained from t by keeping the components of t corresponding to strictly positive values of v (let K denote that subset), and lowering the ones for $V \backslash K$. For $\alpha > 0$, the level sets $\{w \geqslant \alpha\}$ are equal to $\{v \geqslant \alpha\} \subset K$. Thus, by Prop. 4.2, all of these are tight for t and hence for s because these sets are included in K, and $s_K = t_K$. This shows, by Prop. 4.2, that $s \in P(F)$ is optimal for $\max_{s \in P(F)} w^\top s$. □

Note that Prop. 5.6 involves primal-dual pairs (w,s) and (v,t), but that we can define w from v only, and define s from t only; thus,

primal-only views and dual-only views are possible. This also applies to Prop. 5.7, which extends Prop. 5.6 to the symmetric independent polyhedron (we denote by $a \circ b$ the pointwise product between two vectors of same dimension).

Proposition 5.7. (Separable optimization on the symmetric independent polyhedron) Assume that F is submodular and non-decreasing. Let ψ_j, $j = 1, \ldots, p$ be p convex functions such that ψ_j^* is defined and finite on $\mathbb{R}$. Let $\varepsilon_k \in \{-1, 1\}$ denote the sign of $(\psi_k^*)'(0)$ (if it is equal to zero, then the sign can be -1 or 1). Let (v, t) be a primal-dual optimal pair for the problem

$$\min_{v \in \mathbb{R}^p} f(v) + \sum_{k \in V} \psi_k(\varepsilon_k v_k) = \max_{t \in B(F)} - \sum_{k \in V} \psi_k^*(-\varepsilon_k t_k).$$

Let $w = \varepsilon \circ (v_+)$ and s_k be ε_k times a maximizer of $-\psi_k^*(-s_k)$ on $(-\infty, t_k]$. Then (w, s) is a primal-dual optimal pair for the problem

$$\min_{w \in \mathbb{R}^p} f(|w|) + \sum_{k \in V} \psi_k(w_k) = \max_{s \in |P|(F)} - \sum_{k \in V} \psi_k^*(-s_k).$$

Proof. Because f is non-decreasing with respect to each of its component, we have:

$$\min_{w \in \mathbb{R}^p} f(|w|) + \sum_{k \in V} \psi_k(w_k) \quad = \quad \min_{v \in \mathbb{R}_+^p} f(v) + \sum_{k \in V} \psi_k(\varepsilon_k v_k).$$

We can thus apply Prop. 5.7 to $w_k \mapsto \psi_k(\varepsilon_k w_k)$, which has Fenchel conjugate $s_k \mapsto \psi_k^*(\varepsilon_k s_k)$ (because $\varepsilon_k^2 = 1$), to get the desired result. $\square$

Applications to sparsity-inducing norms. Prop. 5.7 is particularly adapted to sparsity-inducing norms defined in Section 2.3, as it describes how to solve the proximal problem for the norm $\Omega(w) = f(|w|)$. For a quadratic function, i.e., $\psi_k(w_k) = \frac{1}{2}(w_k - z_k)^2$ and $\psi_k^*(s_k) = \frac{1}{2}s_k^2 + s_k z_k$. Then ε_k is the sign of z_k, and we thus have to minimize

$$\min_{v \in \mathbb{R}^p} f(v) + \frac{1}{2} \sum_{k \in V} (v_k - |z_k|)^2,$$

which is the classical quadratic separable problem on the base polyhedron, and select $w = \varepsilon \circ v_+$. Thus, proximal operators for the norm Ω may be obtained from the proximal operator for the Lovász extension.

6

Separable optimization problems - Algorithms

In the previous section, we have analyzed a series of optimization problems which may be defined as the minimization of a separable function on the base polyhedron. In this section, we consider algorithms to solve these problems; most of them are based on the availability of an efficient algorithm for maximizing linear functions (greedy algorithm from Prop. 2.2). We focus on three types of algorithms. The algorithm we present in Section 6.1 is a divide-and-conquer non-approximate method that will recursively solve the separable optimization problems by defining smaller problems. This algorithm requires to be able to solve submodular function minimization problems of the form $\min_A F(A) - t(A)$, where $t \in \mathbb{R}^p$, and is thus applicable only when such algorithms are available (such as in the case of cuts, flows or cardinality-based functions). The next two sets of algorithms are iterative methods for convex optimization on convex sets for which the support function can be computed, and are often referred to as "Frank-Wolfe" algorithms. The min-norm-point algorithm that we present in Section 6.2 is dedicated to quadratic functions and converges after finitely many operations (but with no complexity bounds), while the conditional gradient algorithms that we consider in Section 6.3 do not exhibit finite convergence but

have known convergence rates.

Note that, from the use of the algorithms presented in this section, we can derive a series of operations on the two polyhedra, namely line searches and orthogonal projections (see also [103]).

6.1   Decomposition algorithm for proximal problems

We now consider an algorithm for proximal problems, which is based on a sequence of submodular function minimizations. It is based on a divide-and-conquer strategy. We adapt the algorithm of [55] and [49, Sec. 8.2]. Note that it can be slightly modified for problems with non-decreasing submodular functions [55] (otherwise, Prop. 5.7 may be used).

For simplicity, we consider *strictly convex differentiable* functions ψ_j^*, $j = 1, \ldots, p$, (so that the minimum in s is unique) and the following recursive algorithm:

(1) Find the unique minimizer $t \in \mathbb{R}^p$ of $\sum_{j \in V} \psi_j^*(-t_j)$ such that $t(V) = F(V)$.

(2) Minimize the submodular function $F - t$, i.e., find the *largest* $A \subset V$ that minimizes $F(A) - t(A)$.

(3) If $A = V$, then t is optimal. Exit.

(4) Find a minimizer s_A of $\sum_{j \in A} \psi_j^*(-s_j)$ over s in the base polyhedron associated to F_A, the restriction of F to A.

(5) Find the unique minimizer $s_{V \setminus A}$ of $\sum_{j \in V \setminus A} \psi_j^*(-s_j)$ over s in the base polyhedron associated to the contraction F^A of F on A, defined as $F^A(B) = F(A \cup B) - F(A)$, for $B \subset V \setminus A$.

(6) Concatenate s_A and $s_{V \setminus A}$. Exit.

The algorithm must stop after *at most p* iterations. Indeed, if $A \neq V$ in step 3, then we must have $A \neq \varnothing$ (indeed, $A = \varnothing$ implies that $t \in P(F)$, which in turns implies that $A = V$ because by construction $t(V) = F(V)$, which leads to a contradiction). Thus we actually split V into two non-trivial parts A and $V \setminus A$. Step 1 is a separable optimization problem with one linear constraint. When ψ_j^* is a quadratic polynomial, it may be obtained in closed form; more precisely, one may minimize $\frac{1}{2}\|t - z\|_2^2$ subject to $t(V) = F(V)$ by taking $t = \frac{F(V)}{p} 1_V + z - \frac{1_V 1_V^\top}{p} z$.

Proof of correctness.   Let s be the output of the algorithm. We first show that $s \in B(F)$. We have for any $B \subset V$:

$$
\begin{aligned}
s(B) &= s(B \cap A) + s(B \cap (V \backslash A)) \\
&\leqslant F(B \cap A) + F(A \cup B) - F(A) \text{ by definition of } s_A \text{ and } s_{V \backslash A} \\
&\leqslant F(B) \text{ by submodularity.}
\end{aligned}
$$

Thus s is indeed in the submodular polyhedron $P(F)$. Moreover, we have $s(V) = s_A(A) + s_{V \backslash A}(V \backslash A) = F(A) + F(V) - F(A) = F(V)$, i.e., s is in the base polyhedron $B(F)$.

Following [55], we now construct a second base $\bar{s} \in B(F)$ as follows: $\bar{s}_A$ is the minimizer of $\sum_{j \in A} \psi_j^*(-s_j)$ over s_A in the base polyhedron associated to the submodular polyhedron $P(F_A) \cap \{s_A \leqslant t_A\}$. From Prop. B.5, the associated submodular function is $H_A(B) = \min_{C \subset B} F(C) + t(A \backslash C)$. We have $H_A(A) = \min_{C \subset A} F(C) - t(C) + t(A) = F(A)$ because A is the largest minimizer of $F - t$. Thus, the base polyhedron associated with H_A is simply $B(F_A) \cap \{s_A \leqslant t_A\}$.

Morover, we define $\bar{s}_{V \backslash A}$ as the minimizer of $\sum_{j \in V \backslash A} \psi_j^*(-s_j)$ over the base polyhedron $B(J^A)$ where we define the submodular function J^A on $V \backslash A$ as follows: $J^A(B) = \min_{C \supset B} F(C \cup A) - F(A) - t(C) + t(B)$. Then $J^A - t$ is non-decreasing and submodular (by Proposition B.6). Moreover, $J^A(V \backslash A) = F(V) - F(A)$ and $J^A \leqslant F^A$. Finally $B(F^A) \cap \{s_{V \backslash A} \geqslant t_{V \backslash A}\} = B(J^A)$.

We now show that $\bar{s}$ is optimal for the problem. Since $\bar{s}$ has a higher objective value than s (because s is minimized on a larger set), the base s will then be optimal as well. In order to show optimality, we need to show that if w denotes the vector of gradients (i.e., $w_k = -(\psi_k^*)'(-\bar{s}_k)$), then $\bar{s}$ is a maximizer of $s \mapsto w^\top s$ over $s \in B(F)$. Given Prop. 4.2, we simply need to show that $\bar{s}$ is tight for all level sets $\{w \leqslant \alpha\}$. Since, by construction $\bar{s}_k \leqslant \bar{s}_q$ for all $s \in A$ and $q \in V \backslash A$, level sets are included in A or in $V \backslash$. Thus, by optimality of $\bar{s}_A$ and $\bar{s}_{V \backslash A}$, these level sets are indeed tight, hence optimality.

Note finally that similar algorithms may be applied when we restrict s to be integers (see, e.g., [55, 62]).

6.2   Iterative algorithms - Exact minimization

In this section, we focus on quadratic separable problems. Note that modifying the submodular function by adding a modular term[1], we can consider $\psi_k = \frac{1}{2}w_k^2$. As shown in Prop. 5.1, minimizing $f(w) + \frac{1}{2}\|w\|_2^2$ is equivalent to minimizing $\frac{1}{2}\|s\|_2^2$ such that $s \in B(F)$.

Thus, we can minimize $f(w) + \frac{1}{2}\|w\|_2^2$ by computing the minimum ℓ_2-norm element of the polytope $B(F)$, or equivalently the orthogonal projection of 0 onto $B(F)$. Although $B(F)$ may have exponentially many extreme points, the greedy algorithm of Prop. 2.2 allows to maximize a linear function over $B(F)$ at the cost of p function evaluations. The minimum-norm point algorithm of [135] is dedicated to such a situation, as outlined by [50]. It turns out that the minimum-norm point algorithm can be interpreted as a standard active set algorithm for quadratic programming, which we now describe.

Frank Wolfe algorithm as an active set algorithm.   We consider m points $x_1, \ldots, x_m$ in $\mathbb{R}^p$ and the following optimization problem:

$$\min_{\eta \in \mathbb{R}_+} \frac{1}{2}\Big\|\sum_{i=1}^m \eta_i x_i\Big\|_2^2 \text{ such that } \eta \geqslant 0, \ \eta^\top 1 = 1.$$

In our situation, the vectors x_i will be the extreme points of $B(F)$, i.e., outputs of the greedy algorithm, but they will always be used *implicitly* through the maximization of linear functions over $B(F)$. We will exactly apply the primal active set strategy outlined in Section 16.4 of [114], which is exactly the algorithm of [135]. The active set strategy hinges on the fact that if the set of indices $j \in J$ for which $\eta_j > 0$ is known, the solution η_J may be obtained in closed form by computing the affine projection on the set of points indexed by I (which can be implemented by solving a positive definite linear system, see step 2 in the algorithm below). Two cases occur: (a) If the affine projection happens to have non-negative components, i.e., $\eta_J \geqslant 0$ (step 3), then we obtain in fact the projection onto the convex hull of the points

---

[1] Indeed, we have $\frac{1}{2}\|w-z\|_2^2 + f(w) = \frac{1}{2}\|w\|_2^2 + (f(w) - w^\top z) + \frac{1}{2}\|z\|^2$, which corresponds (up to the irrelevant constant term $\frac{1}{2}\|z\|_2^2$) to the proximal problem for the Lovász extension of $A \mapsto F(A) - z(A)$.

indexed by J, and we simply need to check optimality conditions and make sure that no other point needs to enter the hull (step 5), and potentially add it to go back to step 2. (b) If the projection is not in the convex hull, then we make a move towards this point until we exit the convex hull (step 4) and start again at step 2. We describe in Figure 6.1 an example of several iterations.

(1) **Initialization**: We start from a feasible point $\eta \in \mathbb{R}^p_+$ such that $\eta^\top 1 = 1$, and denote J the set of indices such that $\eta_j > 0$ (more precisely a subset of J such that the set of vectors indexed by the subset is linearly independent). Typically, we select one of the original points, and J is a singleton.

(2) **Projection onto affine hull**: Compute ζ_J the unique minimizer $\frac{1}{2}\big\| \sum_{j \in J} \eta_j x_j \big\|_2^2$ such that $1^\top \eta_J = 1$, i.e., the orthogonal projection of 0 onto the *affine* hull of the points $(x_i)_{i \in J}$.

(3) **Test membership in convex hull**: If $\zeta_J \geqslant 0$ (we in fact have an element of the convex hull), go to step 5

(4) **Line search**: Let $\alpha \in [0, 1)$ be the largest α such that $\eta_J + \alpha(\zeta_J - \eta_J) \geqslant 0$. Let K the sets of j such that $\eta_j + \alpha(\zeta_j - \eta_j) = 0$. Replace J by $J \backslash K$ and η by $\eta + \alpha(\zeta - \eta)$, and go to step 2.

(5) **Check optimality**: Let $y = \sum_{j \in J} \eta_j x_j$. Compute a minimizer i of $y^\top x_i$. If $y^\top x_i = y^\top \eta$, then η is optimal. Otherwise, replace J by $J \cup \{j\}$, and go to step 2.

The previous algorithm terminates in a finite number of iterations because it strictly decreases the quadratic cost function at each iteration; however, there is no known bounds regarding the number of iterations (see more details in [114]). Note that in pratice, the algorithm is stopped after either (a) a certain duality gap has been achieved—given the candidate η, the duality gap for η is equal to $\|\bar{x}\|_2^2 + \max_{i \in \{1,...,m\}} \bar{x}_i$, where $\bar{x} = \sum_{i=1}^m \eta_i x_i$ (in the context of application to orthogonal projection on $B(F)$, following Section 5.4, one may get an improved duality gap by solving an isotonic regression problem); or (b), the affine projection cannot be performed reliably because of bad condition number (for more details regarding stopping criteria, see [135]).
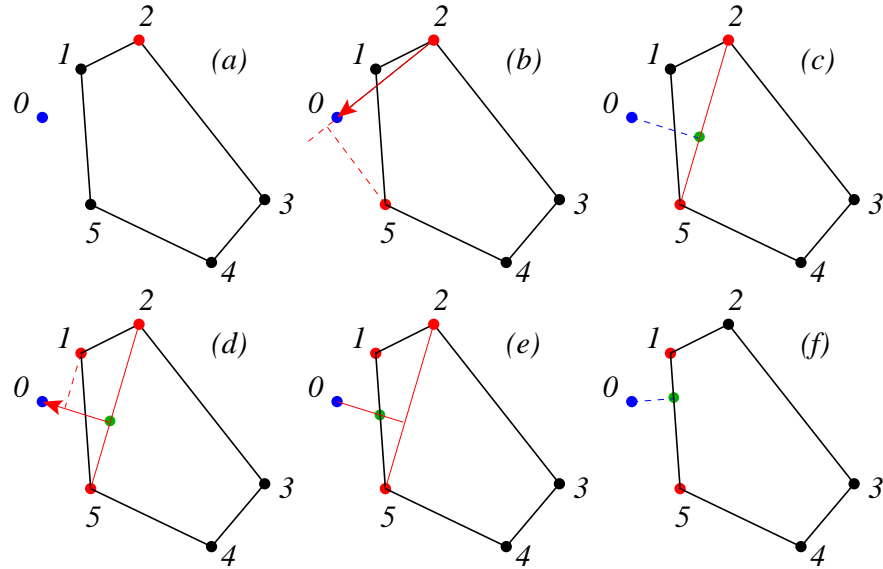
Fig. 6.1: Illustration of Frank-Wolfe minimum norm point algorithm: (a) initialization with $J = \{2\}$ (step 1), (b) check optimality (step 5) and take $J = \{2,5\}$, (c) compute affine projection (step 2), (d) check optimality and take $J = \{1,2,5\}$, (e) perform line search (step 3) and take $J = \{1,5\}$, (f) compute affine projection (step 2) and obtain optimal solution.

6.3   Iterative algorithms - Approximate minimization

In this section, we describe an algorithm strongly related to the minimum-norm point algorithm presented in Section 7.2. This "conditional gradient" algorithm is dedicated to minimization of any convex smooth functions on the base polyhedron. Following the same argument than for the proof of Prop. 5.1, this is equivalent to the minimization of any convex strictly convex separable function regularized by the Lovász extension. As opposed to the mininum-norm point algorithm, it is not convergent in finitely many iterations; however, as shown in Appendix A.2, it comes with approximation guarantees.

When performing optimization on the convex set $B(F)$, it is usually necessary to bound the convex set in some way. In our situation,

the base polyhedron is included in the hyper-rectangle $\prod_{k \in V}[F(V) - F(V \backslash \{k\}), F(\{k\})]$ (as a consequence of the greedy algorithm applied to $1_{\{k\}}$ and $-1_{\{k\}}$). We denote by α_k the length of the interval for variable k, i.e., $\alpha_k = F(\{k\}) + F(V \backslash \{k\}) - F(V)$.

In this section, we also denote $D^2 = \sum_{k=1}^{p} \max\{|F(\{k\})|, |F(V \backslash \{k\})|\}^2$. We then have that $B(F)$ is included in the ℓ_2-ball of center zero and radius D.

Conditional gradient algorithms.   If g is a smooth convex function defined on $\mathbb{R}^p$ with Lipschitz-continuous gradient (with constant L), then the conditional gradient algorithm is an iterative algorithm that will (a) start from a certain $s_0 \in B(F)$, and (b) iterate the following procedure for $t \geqslant 1$: find a minimizer $\bar{s}_{t-1}$ over the (compact) polytope $B(F)$ of the Taylor expansion of g around s_{t-1}, i.e, $s \mapsto g(s_{t-1}) + g'(s_{t-1})^\top(s - s_{t-1})$, and perform a step towards $\bar{s}_{t-1}$, i.e., compute $s_t = \omega_{t-1}\bar{s}_{t-1} + (1 - \omega_{t-1})s_{t-1}$.

There are several strategies for computing ω_{t-1}. The first is to take $\omega_{t-1} = 1/t$ [41, 72], while the second one is to perform a line search on the quadratic upper-bound on g obtained from the L-Lipschitz continuity of g (see Appendix A.2 for details). They both exhibit the same upper bound on the sub-optimality of the iterate s_t, together with $g'(w_t)$ playing the role of a certificate of optimality. More precisely, we have for the line search method (see Appendix A.2):

$$g(s_t) - \min_{s \in B(F)} g(s) \leqslant \frac{L \sum_{k=1}^{p} \alpha_k^2}{t + 1},$$

and the *computable* quantity $\max_{s \in B(F)} g'(s_t)^\top(s - s_t)$ provides a certificate of optimality, that is, we always as $g(s_t) - \min_{s \in B(F)} g(s) \leqslant \max_{s \in B(F)} g'(s_t)^\top(s - s_t)$, and the latter quantity has (up to constants) the same convergence rate. Note that while this certificate comes with an offline approximation guarantee, it can be significantly improved, following Section 5.4, by solving an appropriate isotonic regression problem (see simulations in Section 9).

In Figure 6.2, we consider the conditional gradient algorithm (with line search) for the quadratic problem considered in Section 6.2. These two algorithms are very similar as they both consider a sequence of

extreme points of $B(F)$ obtained from the greedy algorithm, but they differ in the following way: the min-norm-point algorithm is finitely convergent but with no convergence rate, while the conditional gradient algorithm is not finitely convergent, but with a convergence rate. Moreover, the cost per iteration for the min-norm-point algorithm is much higher as it requires linear system inversions. In context where the function F is cheap to evaluate, this may become a computational bottleneck.

Subgradient descent algorithm.   Under the same assumption as before, the Fenchel conjugate of g is strongly convex with constant $1/L$ (see Appendix A.2 for the definition of strong convexity). Moreover, we may restrict optimization to the ball of radius D (if g is D-Lipschitz-continuous). We can thus apply the subgradient descent algorithm described in Appendix A.2, with iteration $w_t = w_{t-1} - \frac{1}{t}\big[(g^*)'(w_{t-1}) + \bar{s}_{t-1}\big]$ (where $\bar{s}_{t-1}$ is a subgradient of f at w_{t-1}, i.e., a maximiser of $s^\top w_{t-1}$ over $s \in B(F)$), and obtain a convergence rate

$$ g^*(x_t) - \min_{u \in \{0,\dots,t\}} g^*(x_u) \leqslant \frac{LD^2}{2} \frac{1 + \log t}{t}. $$

The convergence rate is similar to the one for the conditional gradient, but these are only upper bounds, and, as shown in the experiments, the conditional gradient is faster. Note moreover, that when applied to quadratic functions, the subgradient algorithm is then equivalent to applying a conditional gradient algorithm with no line search to the dual problem (and a constant step size); indeed, we may rewrite the recursion as $(-w_t) = (-w_{t-1}) + \frac{1}{t}\big[\bar{s}_{t-1} - (-w_{t-1})\big]$, i.e., $-w_t$ is the iterate of a conditional gradient algorithm.
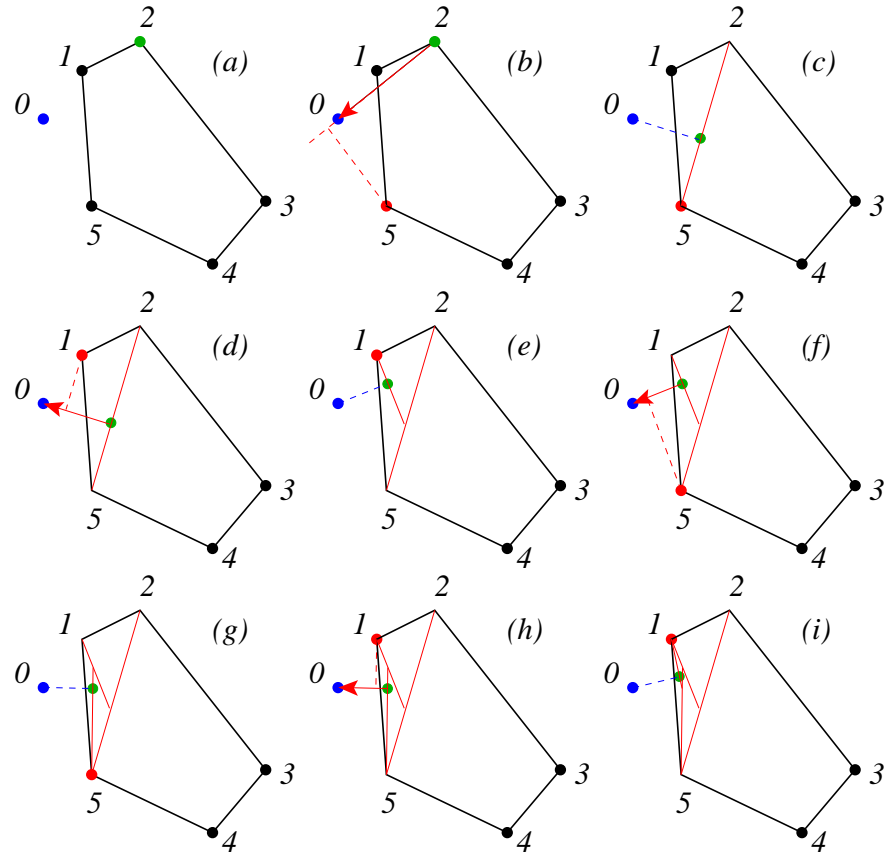
Fig. 6.2: Illustration of Frank-Wolfe conditional gradient algorithm: starting from the inialization (a), in steps (b),(d),(f),(h), an extreme point on the polytope is found an in steps (c),(e),(g),(i), a line search is performed. Note the oscillations to converge to the optimal point (especially compared to Figure 6.1).

7

Submodular function minimization

Several generic algorithms may be used for the minimization of a sub-modular function. They are all based on a sequence of evaluations of $F(A)$ for certain subsets $A \subset V$. For specific functions, such as the ones defined from cuts or matroids, faster algorithms exist (see, e.g., [51, 62], Section 3.2 and Section 3.8). For other special cases, such as functions obtained as the sum of functions that depend on the intersection with small subsets of V, faster algorithms also exist (see, e.g., [130, 83]).

Submodular function minimization algorithms may be divided in two main categories: exact algorithms aim at obtaining a global mini-mizer, while approximate algorithms only aim at obtaining an approx-imate solution, that is, a set A such that $F(A) - \min_{B \subset V} F(B) \leqslant \varepsilon$, where ε is as small as possible. Note that if ε is less than the minimal absolute difference δ between non-equal values of F, then this leads to an exact solution, but that in many cases, this difference δ may be arbitrarily small.

An important practical aspect of submodular function minimization is that most algorithms come with online approximation guarantees; indeed, because of a duality relationship detailed in Section 7.1, in a very similar way to convex optimization, a base $s \in B(F)$ may serve as a

certificate for optimality. Note that all algorithms except the minimum-norm-point algorithm from Section 7.2 come with offline approximation guarantees.

In Section 7.3, we review combinatorial algorithms for submodular function minimization that come with complexity bounds. Those are however not used in practice in particular due to their high theoretical complexity (i.e., $O(p^6)$), except for the particular class of posimodular functions, where algorithms scale as $O(p^3)$ (see Section 7.4).

In Section 7.5, we describe optimization algorithms based on separable optimization problems regularized by the Lovász extension. Using directly the equivalence presented in Prop. 2.4, we can minimize the Lovász extension f on the hypercube $[0, 1]^p$ using subgradient descent with approximate optimality for submodular function minimization of $O(1/\sqrt{t})$ after t iterations. Using quadratic separable problems, we can use the algorithms of Section 6.3 to obtain new submodular function minimization algorithms with convergence of the convex optimization problem at rate $O(1/t)$, which translates through the analysis of Section 5 to the same convergence rate of $O(1/\sqrt{t})$ for submodular function minimization, although with improved behavior and better empirical performance (see Section 7.5 and Section 9).

We also consider in Section 7.2 a formulation based on quadratic separable problem on the base polyhedron, but using the minimum-norm-point algorithm described in Section 6.2: this is one of the fastest in practice, but it comes with no complexity bounds.

Note that *maximizing* submodular functions is a hard combinatorial problem in general. However, when maximizing a non-decreasing submodular function under a cardinality constraint, the simple greedy method allows to obtain a $(1 - 1/e)$-approximation [111] (see more details in Section 8).

7.1   Minimizers of submodular functions

In this section, we review some relevant results for submodular function minimization (for which algorithms are presented in next sections).

Proposition 7.1. (Lattice of minimizers of submodular func-

tions) Let F be a submodular function such that $F(\varnothing) = 0$. The set of minimizers of F is a lattice, i.e., if A and B are minimizers, so are $A \cup B$ and $A \cap B$.

Proof. Given minimizers A and B of F, then, by submodularity, we have $2\min_{C \subset V} F(C) \leqslant F(A \cup B) + F(A \cap B) \leqslant F(A) + F(B) = 2\min_{C \subset V} F(C)$, hence equality in the first inequality, which leads to the desired result. $\square$

The following proposition shows that some form of local optimality implies global optimality.

Proposition 7.2. (Property of minimizers of submodular functions) Let F be a submodular function such that $F(\varnothing) = 0$. The set $A \subset V$ is a minimizer of F on 2^V if and only if A is a minimizer of the function from 2^A to $\mathbb{R}$ defined as $B \subset A \mapsto F(B)$, and if $\varnothing$ is a minimizer of the function from $2^{V \setminus A}$ to $\mathbb{R}$ defined as $B \subset V \setminus A \mapsto F(B \cup A) - F(A)$.

Proof. The set of two conditions is clearly necessary. To show that it is sufficient, we let $B \subset V$, we have: $F(A) + F(B) \geqslant F(A \cup B) + F(A \cap B) \geqslant F(A) + F(A)$, by using the submodularity of F and then the set of two conditions. This implies that $F(A) \leqslant F(B)$, for all $B \subset V$, hence the desired result. $\square$

The following proposition provides a useful step towards submodular function minimization. In fact, it is the starting point of most polynomial-time algorithms presented in Section 7.3. Note that submodular function minimization may also be obtained from minimizing $\|s\|_2^2$ over s in the base polyhedron (see Section 5 and Section 5.4).

Proposition 7.3. (Dual of minimization of submodular functions) Let F be a submodular function such that $F(\varnothing) = 0$. We have:

$$\min_{A \subset V} F(A) = \max_{s \in B(F)} s_-(V) = F(V) - \min_{s \in B(F)} \|s\|_1, \tag{7.1}$$

where $(s_-)_k = \min\{s_k, 0\}$ for $k \in V$. Moreover, given $A \subset V$ and $s \in B(F)$, we always have $F(A) \geqslant s_-(V)$ with equality if and only if $\{s < 0\} \subset A \subset \{s \leqslant 0\}$ and A is *tight* for s, i.e., $s(A) = F(A)$.

We also have

$$\min_{A \subset V} F(A) = \max_{s \in P(F),\ s \leqslant 0} s(V). \tag{7.2}$$

Moreover, given $A \subset V$ and $s \in P(F)$ such that $s \leqslant 0$, we always have $F(A) \geqslant s(V)$ with equality if and only if $\{s < 0\} \subset A$ and A is tight for s, i.e., $s(A) = F(A)$.

Proof. We have, by convex duality, and Props. 2.4 and 4.1:

$$\min_{A \subset V} F(A) = \min_{w \in [0,1]^p} f(w)$$

$$= \min_{w \in [0,1]^p} \max_{s \in B(F)} w^\top s = \max_{s \in B(F)} \min_{w \in [0,1]^p} w^\top s = \max_{s \in B(F)} s_-(V).$$

Strong duality indeed holds because of Slater's condition ($[0,1]^p$ has non-empty interior). Since $s(V) = F(V)$ for all $s \in B(F)$, we have $s_-(V) = F(V) - \|s\|_1$, hence the second equality.

Moreover, we have, for all $A \subset V$ and $s \in B(F)$:

$$F(A) \geqslant s(A) = s(A \cap \{s < 0\}) + s(A \cap \{s > 0\}) \geqslant s(A \cap \{s < 0\}) \geqslant s_-(V),$$

with equality if there is equality in the three inequalities. The first one leads to $s(A) = F(A)$. The second one leads to $A \cap \{s > 0\} = \varnothing$, and the last one leads to $\{s < 0\} \subset A$. Moreover,

$$\max_{s \in P(F),\ s \leqslant 0} s(V) = \max_{s \in P(F)} \min_{w \geqslant 0} s^\top 1_V - w^\top s = \min_{w \geqslant 0} \max_{s \in P(F)} s^\top 1_V - w^\top s$$

$$= \min_{1 \geqslant w \geqslant 0} f(1_V - w) \text{ because of property (c) in Prop. 4.1,}$$

$$= \min_{A \subset V} F(A) \text{ because of Prop. 2.4.}$$

Finally, given $s \in P(F)$ such that $s \leqslant 0$ and $A \subset V$, we have:

$$F(A) \geqslant s(A) = s(A \cap \{s < 0\}) \geqslant s(V),$$

with equality if and only if A is tight and $\{s < 0\} \subset A$. $\qquad\square$

7.2   Minimum-norm point algorithm

From Eq. (5.5) or Prop. 5.4, we obtain that if we know how to minimize $f(w)+\frac{1}{2}\|w\|_2^2$, or equivalently, minimize $\frac{1}{2}\|s\|_2^2$ such that $s \in B(F)$, then we get all minimizers of F from the negative components of s.

We can then apply the minimum-norm point algorithm detailed in Section 6.2 to the vertices of $B(F)$, and notice that step 5 does not require to list all extreme points, but simply to maximize (or minimize) a linear function, which we can do owing to the greedy algorithm. The complexity of each step of the algorithm is essentially $O(p)$ function evaluations and operations of order $O(p^3)$. However, there are no known upper bounds on the number of iterations. Finally, we obtain $s \in B(F)$ as a convex combination of extreme points.

Note that once we know which values of the optimum values s (or w) should be equal, greater or smaller, then, we obtain in closed form all values. Indeed, let $v_1 > v_2 > \cdots > v_m$ the m different values taken by w, and A_i the corresponding sets such that $w_k = v_j$ for $k \in A_j$. Since we can express $f(w) + \frac{1}{2}\|w\|_2^2 = \sum_{j=1}^{m} \{v_j[F(A_1 \cup \cdots \cup A_j) - F(A_1 \cup \cdots \cup A_{j-1})] + \frac{|A_j|}{2}c_j^2\}$, we then have:

$$v_j = \frac{-F(A_1 \cup \cdots \cup A_j) + F(A_1 \cup \cdots \cup A_{j-1})}{|A_j|}, \tag{7.3}$$

which allows to compute the values v_j knowing only the sets A_j (i.e., the ordered partition of constant sets of the solution). This shows in particular that minimizing $f(w) + \frac{1}{2}\|w\|_2^2$ may be seen as a certain search problem over ordered partitions.

7.3   Combinatorial algorithms

Most algorithms are based on Prop. 7.3, i.e., on the identity $\min_{A \subset V} F(A) = \max_{s \in B(F)} s_-(V)$. Combinatorial algorithms will usually output the subset A and a base $s \in B(F)$ such that A is tight for s and $\{s < 0\} \subset A \subset \{s \leqslant 0\}$, as a certificate of optimality.

Most algorithms, will also output the largest minimizer A of F, or sometimes describe the entire lattice of minimizers. Best algorithms have polynomial complexity [123, 70, 116], but still have high complexity (typically $O(p^6)$ or more). Most algorithms update a sequence of

convex combination of vertices of $B(F)$ obtained from the greedy algorithm using a specific order. Recent algorithms [73] consider efficient reformulations in terms of generalized graph cuts.

Note here the difference between the combinatorial algorithm which maximizes $s_-(V)$ and the ones based on the minimum-norm point algorithm which maximizes $-\frac{1}{2}\|s\|_2^2$ over the base polyhedron $B(F)$. In both cases, the submodular function minimizer A is obtained by taking the negative values of s. In fact, the unique minimizer of $\frac{1}{2}\|s\|_2^2$ is also a maximizer of $s_-(V)$, but not vice-versa.

7.4   Minimizing symmetric posimodular functions

A submodular function F is said symmetric if for all $B \subset V$, $F(V\backslash B) = F(B)$. By applying submodularity, get that $2F(B) = F(V\backslash B) + F(B) \geqslant F(V) + F(\varnothing) = 2F(\varnothing) = 0$, which implies that F is non-negative. Hence its global minimum is attained at V and $\varnothing$. Undirected cuts (see Section 3.2) are the main classical examples of such functions.

Such functions can be minimized in time $O(p^3)$ over all *non-trivial* (i.e., different from $\varnothing$ and V) subsets of V through a simple algorithm of Queyranne [117]. Moreover, the algorithm is valid for the regular minimization of *posimodular* functions [102], i.e., of functions that satisfies

$$\forall A, B \subset V, \ F(A) + F(B) \geqslant F(A\backslash B) + F(B\backslash A).$$

These include symmetric submodular functions as well as non-decreasing modular functions, and hence the sum of any of those (in particular, cuts with sinks and sources, as presented in Section 3.2). Note however that this does not include general modular functions (i.e., with potentially negative values); worse, minimization of functions of the form $\lambda F(A) - z(A)$ is provably as hard as general submodular function minimization [117]. Therefore this $O(p^3)$ algorithm is quite specific and may not be used for solving proximal problems with symmetric functions.

7.5 Approximate minimization through convex optimization

In this section, we consider two approaches to submodular function minimization based on iterative algorithms for convex optimization: a *direct* approach, which is based on minimizing the Lovász extension directly on $[0,1]^p$ (and thus using Prop. 2.4), and an *indirect approach*, which is based on quadratic separable optimization problems (and thus using Prop. 5.5). All these algorithms will access the submodular function through the greedy algorithm, once per iteration, with minor operations inbetween.

Restriction of the problem. Given a submodular function F, if $F(\{k\}) < 0$, then k must be in any minimizer of F, since, because of submodularity, if it is not, then adding it would reduce the value of F. Similarly, if $F(V) - F(V\backslash\{k\}) > 0$, then k must in the complement of any minimizer of F. Thus, if denote $A_{\min}$ the set of $k \in V$ such that $F(\{k\}) < 0$ and $A_{\max}$ the complement of the set of $k \in V$ such that $F(V) - F(V\backslash\{k\}) > 0$, then we may restrict the minimization of F to subset A such that $A_{\min} \subset A \subset A_{\max}$. This is equivalent to minimizing the submodular function $A \mapsto F(A \cup A_{\min}) - F(A_{\min})$ on $A_{\max}\backslash A_{\min}$.

From now on, (mostly for the convergence rate described below) we assume that we have done this restriction and that we are now minimizing a function F so that for all $k \in V$, $F(\{k\}) \geqslant 0$ and $F(V) - F(V\backslash\{k\}) \leqslant 0$. We denote by $\alpha_k = F(\{k\}) + F(V\backslash\{k\}) - F(V)$, which is non-negative by submodularity. Note that in practice, this restriction can be seamlessly done by starting regular iterative methods from specific starting points.

Direct approach. From Prop. 2.4, we can use any convex optimization algorithm to minimize $f(w)$ on $w \in [0,1]^p$. Following [60], we consider subgradient descent with step-size $\gamma_t = \frac{D\sqrt{2}}{\sqrt{pt}}$ (where $D^2 = \sum_{k \in V} \alpha_k^2$), i.e., (a) starting from any $w_0 \in [0,1]^p$, we iterate (a) the computation of a maximiser s_{t-1} of $w_{t-1}^\top s$ over $s \in B(F)$, and (b) the update $w_t = \Pi_{[0,1]^p}\left[w_{t-1} - \frac{D\sqrt{2}}{\sqrt{pt}}s_{t-1}\right]$, where $\Pi_{[0,1]^p}$ is the orthogonal projection onto the set $[0,1]^p$ (which may done by thresholding the components independently).

The following proposition shows that in order to obtain a certified ε-approximate set B, we need at most $\frac{4pD^2}{\varepsilon^2}$ iterations of subgradient descent (whose complexity is that of the greedy algorithm to find a base $s \in B(F)$).

Proposition 7.4. (Submodular function minimization by subgradient descent) After t steps of projected subgradient descent, among the p sup-level sets of w_t, there is a set B such that $F(B) - \min_{A \subset V} F(A) \leqslant \frac{Dp^{1/2}}{\sqrt{2t}}$. Moreover, we have a certificate of optimality $\bar{s}_t = \frac{1}{t+1} \sum_{u=0}^{t} s_u$, so that $F(B) - (\bar{s}_t)_-(V) \leqslant \frac{Dp^{1/2}}{\sqrt{2t}}$, with $D^2 = \sum_{k=1}^{p} \alpha_k^2$.

Proof. Given an approximate solution w so that $0 \leqslant f(w) - f^* \leqslant \varepsilon$, with $f^* = \min_{A \subset V} F(A) = \min_{w \in [0,1]^p} f(w)$, we can sort the elements of w in decreasing order, i.e., $1 \geqslant w_{j_1} \geqslant \cdots \geqslant w_{j_p} \geqslant 0$. We then have, with $B_k = \{j_1, \ldots, j_k\}$,

$$
\begin{aligned}
f(w) - f^* &= \sum_{k=1}^{p-1} (F(B_k) - f^*)(w_{j_k} - w_{j_{k+1}}) \\
&\quad + (F(V) - f^*)(w_{j_p} - 0) + (F(\varnothing) - f^*)(1 - w_{j_1}).
\end{aligned}
$$

Thus, as the sum of positive numbers, there must be at least one B_k such that $F(B_k) - f^* \leqslant \varepsilon$. Therefore, given w such that $0 \leqslant f(w) - f^* \leqslant \varepsilon$, there is at least on the sup-level set of w which has values for F which is ε-approximate.

The subgradients of f, i.e., elements s of $B(F)$ are such that $F(V) - F(V \backslash \{k\}) \leqslant s_k \leqslant F(\{k\})$. This implies that f is Lipschitz-continuous with constant D, with $D^2 = \sum_{k=1}^{p} \alpha_k^2$. Since $[0,1]^p$ is included in an ℓ_2-ball of radius $\sqrt{p}/2$, results from Appendix A.2 imply that we may take $\varepsilon = \frac{Dp^{1/2}}{\sqrt{2t}}$. Moreover, as shown in the Appendix A.2, the average of all subgradients provides a certificate of duality with the same known convergence rate (i.e., if we use it as a certificate, it may lead to much better certificates than the bound actually suggests).

Finally, if we replace the subgradient iteration by $w_t = \Pi_{[0,1]^p} \left[w_{t-1} - \mathrm{Diag}(\alpha)^{-1} \frac{\sqrt{2}}{\sqrt{t}} s_{t-1} \right]$, then this corresponds to a subgradient descent algorithm on the function $w \mapsto f(\mathrm{Diag}(\alpha)^{-1/2} w)$ on the

set $\prod_{k \in V}[0, \alpha_k^{1/2}]$, for which the diameter of the domain and the Lipschitz constant are equal to $\left(\sum_{k \in V} \alpha_k\right)^{1/2}$. We thus obtain the improved convergence rate of $\frac{\sum_{k \in V} \alpha_k}{\sqrt{2t}}$. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ $\square$

The previous proposition relies on the most simple algorithms for convex optimization, subgradient descent, which is applicable in most situations; however, its use is appropriate because the Lovász extension is not differentiable, and the dual problem is also not differentiable. We now consider separable quadratic optimization problems whose duals are the maximization of a concave quadratic function on $B(F)$, which is smooth. We can thus use the conditional gradient algorithm, with a better convergence rate; however, as we show below, when we threshold the solution to obtain a set A, we get the same scaling as before (i.e., $O(1/\sqrt{t})$), with an improved empirical behavior. See below and experimental comparisons in Section 9.

Conditional gradient. We now consider the set-up of Section 5 with $\psi_k(w_k) = \frac{1}{2L_k}w_k^2$, and thus $\psi_k^*(s_k) = \frac{L_k}{2}s_k^2$ for certain constants $L_k \geqslant 0$. That is, e consider the conditional gradient algorithm studied in Section 6.3 and Appendix A.2, with $g(s) = \frac{1}{2}\sum_{k \in V}\frac{L_k s_k^2}{2}$: (a) starting from any base $s_0 \in B(F)$, iterate (b) the greedy algorithm to obtain a mininizer $\bar{s}_{t-1}$ of $(s_{t-1} \circ L)^\top s$ with respect to $s \in B(F)$, and (c) perform a line search to minimize with respect to $\omega \in [0, 1]$, $[s_{t-1} + \omega(\bar{s}_{t-1} - s_{t-1})]^\top \mathrm{Diag}(L)[s_{t-1} + \omega(\bar{s}_{t-1} - s_{t-1})]$.

Let $\alpha_k = F(\{k\}) + F(V \backslash \{k\}) - F(V)$, $k = 1, \ldots, p$, be the widths of the hyper-rectangle enclosing $B(F)$. The following proposition shows how to obtain an approximate minimizer of F.

Proposition 7.5. (Submodular function minimization by conditional gradient descent) After t steps of the conditional gradient method described above, among the p sub-level sets of $L \circ s_t$, there is a set B such that $F(B) - \min_{A \subset V} F(A) \leqslant \frac{1}{\sqrt{t}}\sqrt{\frac{\sum_{k=1}^{p}\alpha_k^2 L_k}{2}}\sum_{k=1}^{p}\frac{1}{L_k}$. Moreover, s_t acts as a certificate of optimality, so that $F(B) - (s_t)_-(V) \leqslant \frac{1}{\sqrt{t}}\sqrt{\frac{\sum_{k=1}^{p}\alpha_k^2 L_k}{2}}\sum_{k=1}^{p}\frac{1}{L_k}$.

Proof. The convergence rate analysis of the conditional gradient method leads to an ε-approximate solution with $\varepsilon \leqslant \frac{\sum_{k=1}^{p} \alpha_k^2 L_k}{t+1}$. From Eq. (5.6), if we assume that $(F + \psi'(\alpha))(\{w \geqslant \alpha\}) - (s + \psi'(\alpha))_{-}(V) > \varepsilon/2\eta$ for all $\alpha \in [-\eta, \eta]$, then we obtain (with $\psi'(\alpha)_k = \frac{\alpha}{L_k}$):

$$\varepsilon \geqslant \int_{-\eta}^{+\eta} \left\{ (F + \psi'(\alpha))(\{w \geqslant \alpha\}) - (s + \psi'(\alpha))_{-}(V) \right\} d\alpha > \varepsilon,$$

which is a contradiction. Thus, there exists $\alpha \in [-\eta, \eta]$ such that $0 \leqslant (F + \psi'(\alpha))(\{w \geqslant \alpha\}) - (s + \psi'(\alpha))_{-}(V) \leqslant \varepsilon/2\eta$. Let A^* a minimizer of F. We have:

$$F(\{w \geqslant \alpha\}) + \psi'(\alpha)(\{w \geqslant \alpha\}) \leqslant F(A^*) + \psi'(\alpha)(A^*) + \varepsilon/2\eta,$$

leading to $F(\{w \geqslant \alpha\}) \leqslant F(A^*) + \sum_{k=1}^{p} \frac{\eta}{L_k} + \varepsilon/2\eta$. By choosing $\eta = \sqrt{\frac{\varepsilon}{2 \sum_{k=1}^{p} L_k^{-1}}}$, we obtain $F(\{w \geqslant \alpha\}) \leqslant F(A^*) + \sqrt{\frac{\varepsilon}{2} \sum_{k=1}^{p} L_k^{-1}} \leqslant F(A^*) + \sqrt{\frac{\sum_{k=1}^{p} \alpha_k^2 L_k}{2(t+1)} \sum_{k=1}^{p} L_k^{-1}}$. This leads to a an approximation of

$$\frac{1}{\sqrt{t}} \sqrt{\frac{\sum_{k=1}^{p} \alpha_k^2 L_k}{2} \sum_{k=1}^{p} \frac{1}{L_k}}.$$

$\square$

In the previous proposition, two natural choices for L_k emerge. The traditional choice $L_k = 1$, which leads to a convergence rate of $\sqrt{\frac{p \sum_{k=1}^{p} \alpha_k^2}{2(t+1)}}$, and $L_k \propto \alpha_k^{-1}$, leading to a convergence rate of $\left(\sum_{k=1}^{p} \alpha_k \right) \frac{1}{\sqrt{t+1}}$. Here the convergence rate is the same as for subgradient descent. See Section 9 for an empirical comparison, showing a better behavior for the conditional gradient method. As for subgradient descent, this algorithm provides certificates of optimality. Moreover, when offline (or online) certificates of optimality ensures that we an approximate solution, because the problem is strongly convex, we obtain also a bound $\sqrt{2\varepsilon}$ on $\|s_t - s^*\|_2$ where s^* is the optimal solution. In the case where $L_k = 1$ for all $k \in V$, this in turns allows to ensure that all indices k such that $s_t > \sqrt{2\varepsilon}$ cannot be in a minimizer of F, while

those indices k such that $s_t < -\sqrt{2\varepsilon}$ have to be in a minimizer, which can allow efficient reduction of the search space (although these have not been implemented in the simulations in Section 9).

Alternative algorithms for the same separable optimization problems may be used, i.e., conditional gradient without line search [72, 40], with similar convergence rates and behavior, but sometimes worse empirical peformance, and with a weaker link with the minimum-norm-point algorithm. Another alternative is to consider projected subgradient descent in w, with the same convergence rate (because the objective function is then strongly convex. Note that as shown before (Section 6.3), it is equivalent to a conditional gradient algorithm with no line search.

Smoothing for special case of submodular functions. For some specific submodular functions, it is possible to use alternative optimization algorithms. As outlined in [130], this is appropriate when F may be written as $F(A) = \sum_{G \in \mathcal{G}} F_G(A \cap G)$, where $F_G : G \to \mathbb{R}$ is submodular, the set $\mathcal{G}$ of subsets of V is composed of small subsets, and the Lovász extensions F_G are explicit enough so that one may compute a convex smooth (with Lipschitz-constant of the gradient less than L) approximation of F_G with uniform approximation error of $O(1/L)$. In this situation, the Lovász extension of F may be approximated within $O(1/L)$ by a smooth function on which an accelerated gradient technique such as described in Section 5.1 may be used with convergence rate $O(L/t^2)$ after t iterations. When choosing $L = 1/t$ (thus with a fixef horizon), this leads to an approximation guarantee for submodular function minimization of the form $O(1/t)$, instead of $O(1/t^2)$ in the general case.

8

Other submodular optimization problems

While submodular function *minimization* may be solved in polynomial time (see Section 7), submodular function *maximization* (which includes the maximum cut problem) is NP-hard. Nevertheless, submodularity may be used in order to obtain some local or global guarantees (see Section 8.1 and Section 8.2) or to derive local descent algorithms for more general problems (see Section 8.3).

8.1 Submodular function maximization

In this section, we consider a submodular function and the maximization problem:

$$\max_{A \subset V} F(A). \tag{8.1}$$

This problem is known to be NP-hard (note that it includes the maximum cut problem) [46]. However, several approximation algorithms exist with theoretical guarantees, in particular when the function is known to be non-negative (i.e., with non-negative values $F(A)$ for all $A \subset V$). For example, it is shown in [46] that selecting a random subset

already achieves at least $1/4$ of the optimal value[1], while local search techniques achieve at least $1/2$ of the optimal value.

Local search algorithm.   Given any set A, simple local search algorithms simply consider all sets of the form $A \cup \{k\}$ and $A \backslash \{k\}$ and select the one with largest value of F. If this value is lower than F, then the algorithm stops and we are by definition at a local minimum. While these local minima do not lead to any global guarantees in general, there is an interesting added guarantee based on submodularity, which we now prove (see more details in [53]).

Proposition 8.1. (Local minima for submodular function minimization) Let F be a submodular function and $A \subset V$ such that for all $k \in A$, $F(A \backslash \{k\}) \leqslant F(A)$ and for all $k \in V \backslash A$, $F(A \cup \{k\}) \leqslant F(A)$. Then for all $B \subset A$ and all $B \supset A$, $F(B) \leqslant F(A)$.

Proof. If $B = A \cup \{i_1, \ldots, i_q\}$, then

$$
\begin{aligned}
F(B) - F(A) &= \sum_{j=1}^{q} F(A \cup \{i_1, \ldots, i_j\}) - F(A \cup \{i_1, \ldots, i_{j-1}\}) \\
&\leqslant \sum_{j=1}^{q} F(A \cup \{i_j\}) - F(A) \leqslant 0,
\end{aligned}
$$

which leads to the first result. The second one may be obtained from the first one applied to $A \mapsto F(V \backslash A) - F(V)$. $\qquad\square$

Note that branch-and-bound algorithms (with worst-case exponential time complexity) may be designed that specifically take advantage of the property above [53].

Formulation using base polyhedron.   Given F and its Lovász extension f, we have (the first equality is true since maximization of

---

[1] Such a result for a random subset shows that having theoretical guarantees do not necessarily imply that an algorithm is doing anything subtle.

convex function leads to an extreme point [121]):

$$
\begin{aligned}
\max_{A \subset V} F(A) &= \max_{w \in [0,1]^p} f(w), \\
&= \max_{w \in [0,1]^p} \max_{s \in B(F)} w^\top s \text{ because of Prop. 2.2,} \\
&= \max_{s \in B(F)} s_+(V) = \max_{s \in B(F)} \frac{1}{2}(s + |s|)(B) \\
&= \frac{1}{2}F(V) + \frac{1}{2} \max_{s \in B(F)} \|s\|_1.
\end{aligned}
$$

Thus submodular function maximization may be seen as finding the *maximum* ℓ_1-norm point in the base polyhedron (which is not a convex optimization problem). See an illustration in Figure 8.1.

8.2   Submodular function maximization with cardinality constraints

In this section, we consider a specific instance of submodular maximization problems, with theoretical guarantees.

Greedy algorithm for non-decreasing functions. Submodular function maximization provides a classical example where greedy algorithms do have performance guarantees. We now consider a non-increasing submodular function F and the problem of minimizing $F(A)$ subject to the constraint $|A| \leqslant k$, for a certain k. The greedy algorithm will start with the empty set $A = \varnothing$ and iteratively add the element $k \in V \backslash A$ such that $F(A \cup \{k\}) - F(A)$ is maximal. It has an $(1-1/e)$-performance guarantee [111] (note that this guarantee cannot be improved in general, as it cannot for set cover, see [44]):

Proposition 8.2. (Performance guarantee for submodular function maximization) Let F be a non-decreasing submodular function. The greedy algorithm for maximizing $F(A)$ subset to $|A| \leqslant k$ outputs a set A such that

$$
F(A) \geqslant [1 - (1 - 1/k)^k] \max_{B \subset V, \, |B| \leqslant k} F(B) \geqslant (1 - 1/e) \max_{B \subset V, \, |B| \leqslant k} F(B).
$$

Proof. We follow the proof of [111, 136]. Let $A^* = \{b_1, \ldots, b_k\}$ be a maximizer of F with k elements, and a_j the j-th element selected during the greedy algorithm. We consider $\rho_j = F(\{a_1, \ldots, a_j\}) - F(\{a_1, \ldots, a_{j-1}\})$. We have for all $j \in \{1, \ldots, k\}$:

$$F(A^*)$$

$$\leqslant F(A^* \cup A_{j-1}) \text{ because } F \text{ is non-decreasing,}$$

$$= F(A_{j-1}) + \sum_{i=1}^{k} \left[F(A_{j-1} \cup \{b_1, \ldots, b_i\}) - F(A_{j-1} \cup \{b_1, \ldots, b_{i-1}\}) \right]$$

$$\leqslant F(A_{j-1}) + \sum_{i=1}^{k} \left[F(A_{j-1} \cup \{b_i\}) - F(A_{j-1}) \right] \text{ by submodularity,}$$

$$\leqslant F(A_{j-1}) + k\rho_j \text{ by definition of the greedy algorithm,}$$

$$= \sum_{i=1}^{j-1} \rho_i + k\rho_j.$$

We can now simply minimize $\sum_{i=1}^{k} \rho_i$ subject to the k constraints defined above (plus pointwise positivity), i.e., $\sum_{i=1}^{j-1} \rho_i + k\rho_j \geqslant F(A^*)$. It turns out, that taking all inequalities as equalities leads to an invertible linear system whose solution is $\rho_j = (k-1)^{j-1} k^{-j} \geqslant 0$, leading to $\sum_{i=1}^{k} \rho_i = \sum_{i=1}^{k} (1-1/k)^{i-1} k^{-1} = (1-1/k)^k$, hence the desired result since $(1-1/k)^k = \exp(k \log(1-1/k)) \leqslant \exp(k \times (-1/k)) = 1/e$. $\quad\square$

Extensions. Given the previous result on cardinality constraints, several extensions have been considered, such as knapsack constraints or matroid constraints (see [23] and references therein). Moreover, fast algorithms and online data-dependent bounds can be further derived [99].

8.3 Difference of submodular functions

In regular continuous optimization, differences of convex functions play an important role, and appear in various disguises, such as DC-programming [67], concave-convex procedures [138], or majorization-minimization algorithms [69]. They allow the expression of any continuous optimization problem with natural descent algorithms based on

upper-bounding a concave function by its tangents.

In the context of combinatorial optimization, [106] has shown that a similar situation holds for differences of submodular functions. We now review these properties.

Formulation of any combinatorial optimization problem.   Let $F : 2^V \to \mathbb{R}$ any set-function, and H a *strictly* submodular function, i.e., a function such that

$$\alpha = \min_{A \subset V} \min_{i,j \in V \setminus A} -H(A \cup \{i,j\}) + H(A \cup \{i\}) + H(A \cup \{j\}) - H(A) > 0.$$

A typical example would be $H(A) = -\frac{1}{2}|A|^2$, where $\alpha = 1$. If

$$\beta = \min_{A \subset V} \min_{i,j \in V \setminus A} -F(A \cup \{i,j\}) + F(A \cup \{i\}) + F(A \cup \{j\}) - F(A)$$

is non-negative, then F is submodular (see Prop. 1.2). If $\beta < 0$, then $F(A) - \frac{\beta}{\alpha}H(A)$ is submodular, and thus, we have $F(A) = [F(A) - \frac{\beta}{\alpha}H(A)] - [-\frac{\beta}{\alpha}H(A)]$, which is a difference of two submodular functions. Thus any combinatorial optimization problem may be seen as a difference of submodular functions (with of course non-unique decomposition). However, some problems, such as subset selection in Section 3.7, or more generally discriminative learning of graphical model structure may naturally be seen as such [106].

Optimization algorithms.   Given two submodular set-functions F and G, we consider the following iterative algorithm, starting from a subset A:

(1) Compute modular lower-bound $B \mapsto s(B)$, of G which is tight at A: this might be done by using the greedy algorithm of Prop. 2.2 with $w = 1_A$. Several orderings of components of w may be used (see [106] for more details).

(2) Take A as any minimizer of $B \mapsto F(B) - s(B)$, using any algorithm of Section 7.

It converges to a local minimum, in the sense that at convergence to a set A, all sets $A \cup \{k\}$ and $A \setminus \{k\}$ have smaller function values.
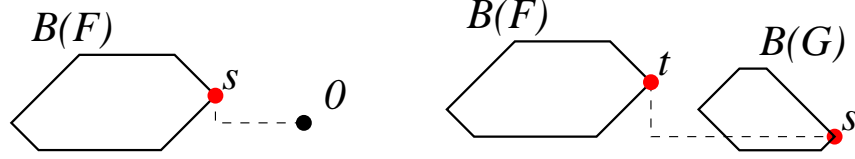
Fig. 8.1: Geometric interpretation of submodular function maximization (left) and optimization of differences of submodular functions (right). See text for details.

Formulation using base polyhedron.   We can give a similar geometric interpretation than for submodular function maximization; given F, G and their Lovász extensions f, g, we have:

$$
\begin{aligned}
\min_{A \subset V} F(A) - G(A) &= \min_{A \subset V} \min_{s \in B(G)} F(A) - s(A) \text{ because of Prop. 2.2,} \\
&= \min_{w \in [0,1]^p} \min_{s \in B(G)} f(w) - s^\top w \text{ because of Prop. 2.4,} \\
&= \min_{s \in B(G)} \min_{w \in [0,1]^p} f(w) - s^\top w \\
&= \min_{s \in B(G)} \min_{w \in [0,1]^p} \max_{t \in B(F)} t^\top w - s^\top w \\
&= \min_{s \in B(G)} \max_{t \in B(F)} \min_{w \in [0,1]^p} t^\top w - s^\top w \text{ by strong duality,} \\
&= \min_{s \in B(G)} \max_{t \in B(F)} (t - s)_-(V) \\
&= \frac{F(V) - G(V)}{2} - \frac{1}{2} \min_{s \in B(G)} \max_{t \in B(F)} \|t - s\|_1.
\end{aligned}
$$

Thus optimization of the difference of submodular functions may be seen as computing the Hausdorff distance (see, e.g., [101]) between $B(G)$ and $B(F)$. See an illustration in Figure 8.1.

9

Experiments

In this section, we provide illustrations of the optimization algorithms described earlier, for submodular function minimization (Section 9.1), as well as for convex optimization problems, quadratic separable ones such as the ones used for proximal methods or within submodular function minimization (Section 9.2), and an application of sparsity-inducing norms to wavelet-based estimators (Section 9.3). The Matlab code for all these experiments may be found at `http://www.di.ens.fr/~fbach/submodular/`.

9.1 Submodular function minimization

We compare several simple though effective approaches to submodular function minimization described in Section 7, namely:

- **min-norm-point**: the minimum-norm-point algorithm to maximize $-\frac{1}{2}\|s\|_2^2$ over $s \in B(F)$, described in Section 7.2.
- **subgrad-des**: the projected gradient descent algorithm to minimize $f(w)$ over $w \in [0,1]^p$, described in Section 7.5.
- **cond-grad**: the conditional gradient algorithm to maximize $-\frac{1}{2}\|s\|_2^2$ over $s \in B(F)$, with line search, described in Sec-
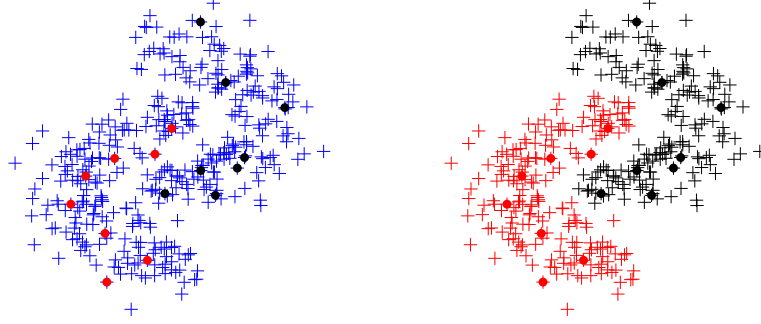
Fig. 9.1: Examples of semi-supervised clustering : (left) observations, (right) results of the semi-supervised clustering algorithm based on submodular function minimization, with eight labelled data points.

tion 7.5.

- **cond-grad-1/t**: the conditional gradient algorithm to maximize $-\frac{1}{2}\|s\|_2^2$ over $s \in B(F)$, with step size $1/t$, described in Section 7.5.
- **cond-grad-w**: the conditional gradient algorithm to maximize $-\frac{1}{2}s^\top \mathrm{Diag}(\alpha)^{-1}s$ over $s \in B(F)$, with line search.

From all these algorithms, we look for the sub-level sets of s to obtain the best value for the set-function F. We also use the base $s \in B(F)$ as a certificate for optimality, through $F(A) - s_-(V)$ (see Prop. 7.3).

We test these algorithms on three data sets:

- **Two moons** (clustering with mutual information criterion): we generated data from a standard synthetic examples in semi-supervised learning (see Figure 9.1) with $p = 400$ data points, and 16 labelled data points, using the method presented in Section 3.5 (based on the mutual information between two Gaussian processes), with a Gaussian-RBF kernel.
- **Genrmf-wide** and **Genrmf-long** (min-cut/max-flow standard benchmark): following [50], we generated cut problem using the generator GENRMF available from DIMACS chal-

lenge[1]. Two types of network were generated, "long" and "wide", with respectively $p = 575$ vertices and 2390 edges, and $p = 430$ and 1872 edges (see [50] for more details).

In Figures 9.2, 9.4 and 9.6, we compare the five algorithms on the three datasets. We denote by Opt the optimal value of the optimization problem, i.e., $\mathrm{Opt} = \min_{w \in \mathbb{R}^p} f(w) = \max_{s \in B(F)} s_-(V)$. On the left plots, we display the dual suboptimality, i.e, $\log_{10}(\mathrm{Opt} - s_-(V))$, together with the certified duality gap (in dashed). In the right plots we display the primal suboptimality $\log_{10}(F(B) - \mathrm{Opt})$. Note that in all the plots in Figures 9.2, 9.3, 9.4, 9.5, 9.6 and 9.7, we plot the best values achieved so far, i.e., we make all curves non-increasing.

Since all algorithms perform a sequence of greedy algorithms (for finding maximum weight bases), we replace running times by numbers of iterations[2]. On all datasets, the achieved primal function values are in fact much lower than the certified values, a situation common in convex optimization, while this is not the case for dual values. Thus primal values $F(A)$ are quickly very good and iterations are just needed to sharpen the certificate of optimality. On all datasets, the min-norm-point algorithm achieved quickest small duality gaps. On all datasets, among the three conditional gradient algorithms, the weighted one (with weights $L_k = 1/\alpha_k$) performs slightly better than the unweighted one, and these two versions with line-search perform significantly better than the algorithm with decaying step sizes. Finally, the direct approach based on subgradient descent performs worse in the two graph-cut examples, in particular in terms of certified duality gaps.

9.2   Separable optimization problems

In this section, we compare the iterative algorithms outlined in Section 6 for minimization on quadratic separable optimization problems, on the problems related to submodular function minimization from the

---

[1]   The   First   DIMACS   international   algorithm   implementa-
tion   challenge: The   core   experiments   (1990),   available   at
`ftp://dimacs.rutgers.edu/pub/netßow/generalinfo/core.tex`.

[2] Only the mininum-norm-point algorithm has a non trivial cost per iteration, and in our experiments, plots with running times would not be significantly different.
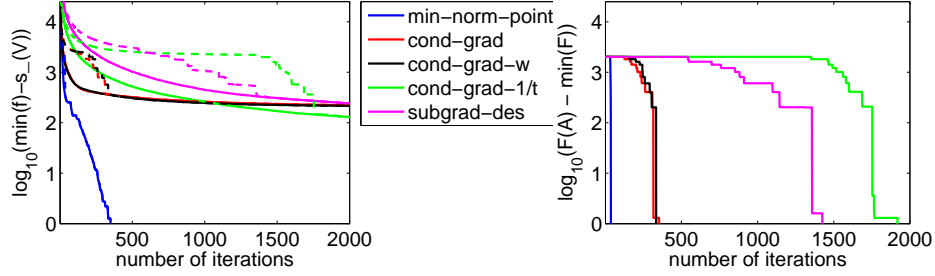
Fig. 9.2: Submodular function minimization results for "Genrmf-wide" example: (left) optimal value minus dual function values in log-scale vs. number of iterations, in dashed, certified duality gap in log-scale vs. number of iteration. (Right) Primal function values minus optimal value in log-scale vs. number of iterations.
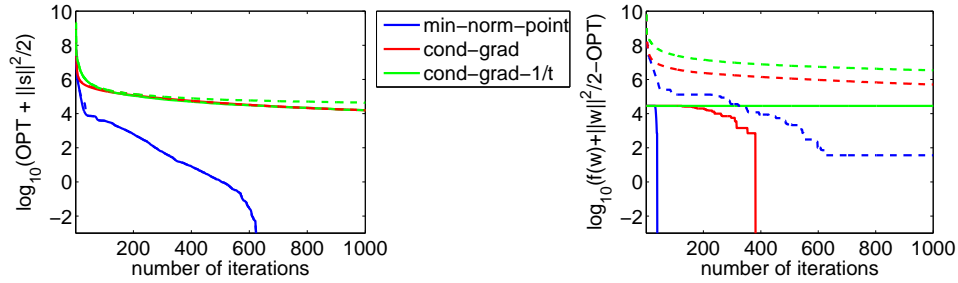

Fig. 9.3: Separable optimization problem for "Genrmf-wide" example. (Left) optimal value minus dual function values in log-scale vs. number of iterations, in dashed, certified duality gap in log-scale vs. number of iteration. (Right) Primal function values minus optimal value in log-scale vs. number of iterations, in dashed, before the "pool-adjacent-violator" correction.

previous section (i.e., minimizing $f(w) + \frac{1}{2}\|w\|_2^2$). In Figures 9.3, 9.5 and 9.7, we compare three algorithms on the three datasets, namely the mininum-norm-point algorithm, and two versions of conditional gradient (with and without line search). On the left plots, we display the achieved quantity $\log_{10}(f(w) + \frac{1}{2}\|w\|_2^2 - \min_{v \in \mathbb{R}^p} f(v) + \frac{1}{2}\|v\|_2^2)$ while in the right plots we display the logarithm of the certified duality gaps, for
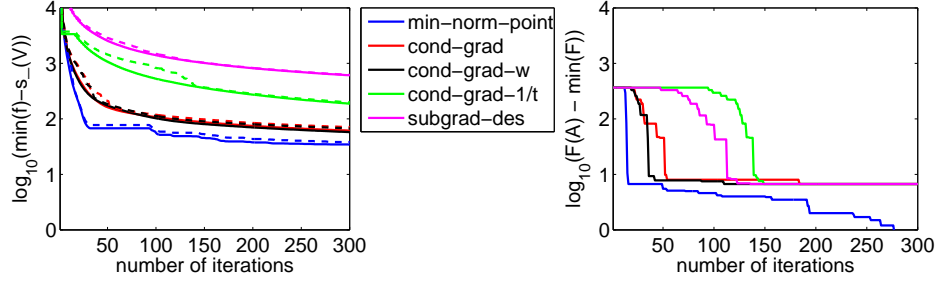
Fig. 9.4: Submodular function minimization results for "Genrmf-long" example: (left) optimal value minus dual function values in log-scale vs. number of iterations, in dashed, certified duality gap in log-scale vs. number of iteration. (Right) Primal function values minus optimal value in log-scale vs. number of iterations.
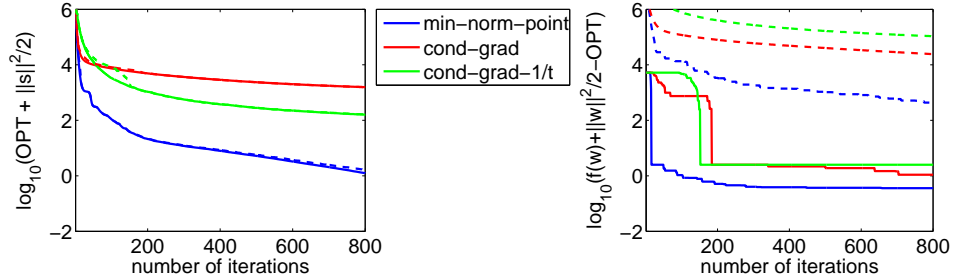

Fig. 9.5: Separable optimization problem for "Genrmf-long" example. (Left) optimal value minus dual function values in log-scale vs. number of iterations, in dashed, certified duality gap in log-scale vs. number of iteration. (Right) Primal function values minus optimal value in log-scale vs. number of iterations, in dashed, before the "pool-adjacent-violator" correction.

the same algorithms. Since all algorithms perform a sequence of greedy algorithms (for finding maximum weight bases), we replace running times by numbers of iterations. As in Section 9.1, on all datasets, the achieved primal function values are in fact much lower than the certified values, a situation common in convex optimization. On all datasets, the min-norm-point algorithm achieved quickest small duality gaps. On
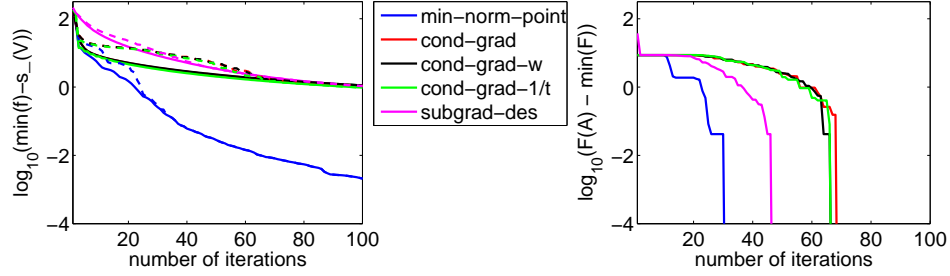
Fig. 9.6: Submodular function minimization results for "Two moons" example: (left) optimal value minus dual function values in log-scale vs. number of iterations, in dashed, certified duality gap in log-scale vs. number of iteration. (Right) Primal function values minus optimal value in log-scale vs. number of iterations.
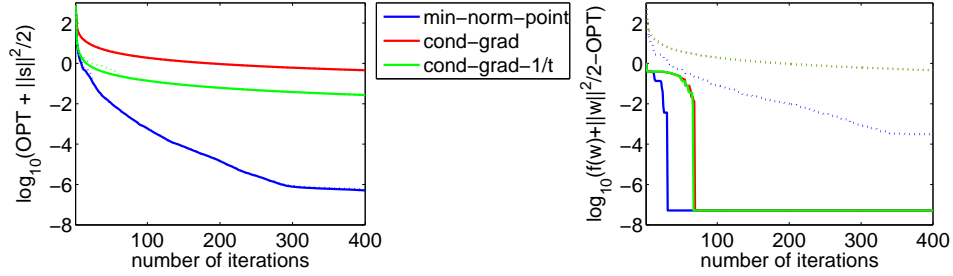

Fig. 9.7: Separable optimization problem for "Two moons" example. (Left) optimal value minus dual function values in log-scale vs. number of iterations, in dashed, certified duality gap in log-scale vs. number of iteration. (Right) Primal function values minus optimal value in log-scale vs. number of iterations, in dashed, before the "pool-adjacent-violator" correction.

all datasets, among the two conditional gradient algorithms, the version with line-search perform significantly better than the algorithm with decaying step sizes. Note also, that while the conditional gradient algorithm is not finitely convergent, its performance is not much worse than the minimum-norm-point algorith, with smaller running time complexity per iteration. Moreover, as shown on the right plots, the "pool-adjacent-violator" correction is crucial in obtaining much im-

proved primal candidates.

9.3   Regularized least-squares estimation

In this section, we illustrate the use of the Lovász extension in the context of sparsity-inducing norms detailed in Section 2.3, with the submodular function defined in Figure 3.5, which is based on a tree structure among the p variables, and encourages variables to be selected after their ancestors. We don't use any weights, and thus $F(A)$ is equal to the cardinality of the union of all ancestors Anc(A) of nodes indexed by elements of A.

Given a probability distribution (x, y) on $[0, 1] \times \mathbb{R}$, we aim to estimate $f(x) = \mathbb{E}(Y|X = x)$, by a piecewise constant function. Following [140], we consider a Haar wavelet estimator with maximal depth d. That is, given the Haar wavelet, defined on $\mathbb{R}$ as $\psi(t) = 1_{[0,1/2)}(t) - 1_{[1/2,1)}(t)$, we consider the functions $\psi_{ij}(t)$ defined as $\psi_{ij}(t) = \psi(2^{i-1}t - j)$, for $i = 1, \ldots, d$ and $j \in \{0, \ldots, 2^{i-1} - 1\}$, leading to $p = 2^d - 1$ basis functions. These functions come naturally in a binary tree structure, as shown in Figure 9.8 for $d = 3$. Imposing a tree-structured prior enforces that a wavelet with given support is selected only after all larger supports are selected; this avoids the selection of isolated wavelets with small supports.

We consider random inputs $x_i \in [0, 1]$, $i = 1, \ldots, n$, from a uniform distribution and compute $y_i = \sin(20\pi x_i^2) + \varepsilon_i$, where ε_i is Gaussian with mean zero and standard deviation 0.1. We consider the optimization problem

$$\min_{w \in \mathbb{R}^p, b \in \mathbb{R}} \frac{1}{2n} \sum_{k=1}^{n} \left(y_k - \sum_{i=1}^{d} \sum_{j=0}^{2^{i-1}-1} w_{ij}\psi_{ij}(x_k) - b \right)^2 + \lambda R(w), \quad (9.1)$$

where b is a constant term and $R(w)$ is a regularization function. In Figure 9.9, we compare several regularization terms, namely $R(w) = \frac{1}{2}\|w\|_2^2$ (ridge regression), $R(w) = \|w\|_1$ (Lasso) and $R(w) = \Omega(w) = f(|w|)$ defined from the hierarchical submodular function $F(A) = \text{Card}(\text{Anc}(A))$. For all of these, we select λ such that the generalization performance is maximized, and compare the estimated functions. The hierarchical prior leads to a lower estimation error with fewer artefacts.
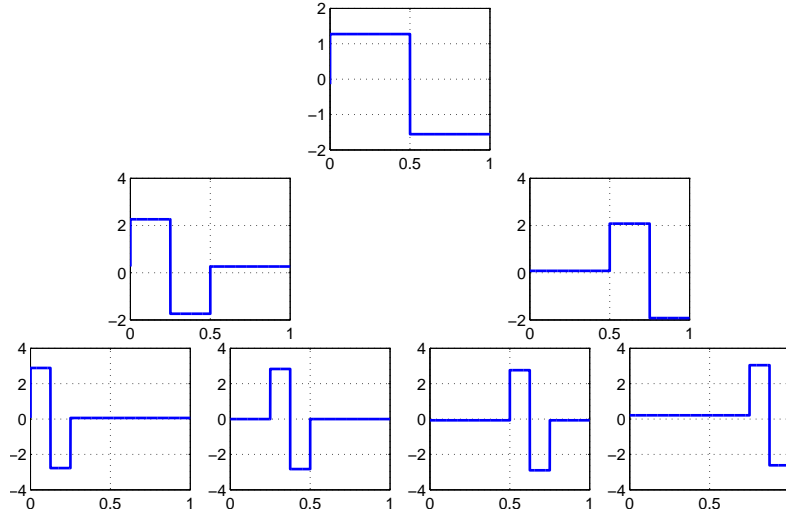
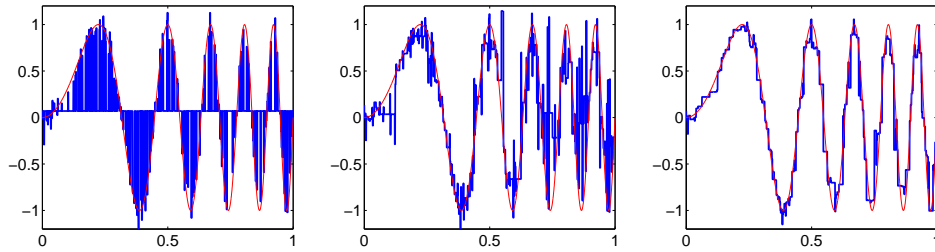Fig. 9.8: Wavelet bynary tree ($d = 3$). See text for details.

Fig. 9.9: Estimation with wavelet trees: (left) ridge regression, (middle) Lasso, (right) hierarchical penalty. See text for details.

In this section, our goal is mainly to compare several optimization schemes to minimize Eq. (9.1) for this particular example (for more simulations on other examples with similar conclusions, see [8, 96, 77, 7]). We compare in Figure 9.10 three ways of computing the proximal operator (within a proximal gradient method) and one direct optimization scheme based on subgradient descent:

- **Prox-hierarchical**: we use a dedicated proximal operator based on the composition of local proximal operators [77].
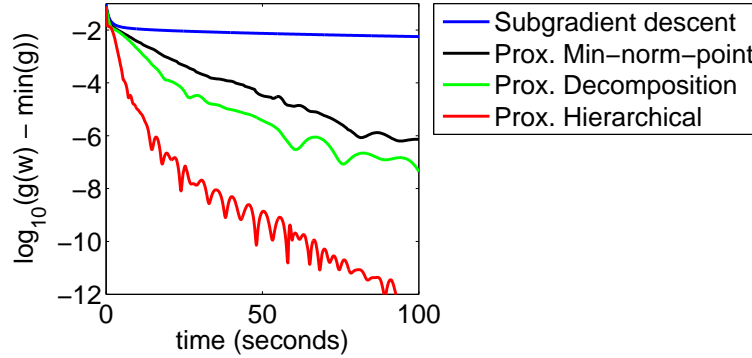
Fig. 9.10: Running times for convex optimization for a regularized problem: several methods are compared; see text for details.

- **Prox-decomposition**: we use the algorithm of Section 6.1 which uses the fact that for any vector t, $F - t$ may be minimized by dynamic programming [77].
- **Prox-min-norm-point**: we use the generic method which does not use any of the structure.
- **subgrad-descent**: we use a generic method which does not use any of the structure, and minimize directly Eq. (9.1) by subgdradient descent.

As expected, in Figure 9.10, we see that the most efficient algorithm is the dedicated proximal algorithm (which is usually not available except in particular cases like the tree-structured norm), while the methods based on submodular functions fare correctly, with an advantage for methods using the structure (i.e., the decomposition method, which is only applicable when submodular function minimization is efficient) over the generic method based on the min-norm-point algorithm (which is always applicable).

Conclusion

In this paper , we have explored various properties and applications of submodular functions. Key concepts are the Lovász extension and the associated submodular and base polyhedra. Given the numerous examples involving such functions, the analysis and algorithms presented in this paper allow the unification of several results in convex optimization, involving structured situations and notably sparsity-inducing norms. Several questions related to submodular functions remain open, such as efficient combinatorial optimization algorithms for submodular function minimization, with both good computational complexity bounds and practical performance. Moreover, we have presented algorithms for approximate submodular function minimization with convergence rate of the form $O(1/\sqrt{t})$ where t is the number of calls to the greedy algorithm; it would be interesting to obtain better rates or show that this rate is optimal. Finally, submodular functions essentially consider links between combinatorial optimization problems and linear programming, or linearly constrained quadratic programming; it would be interesting to extend submodular analysis so that more modern convex optimization tools such as semidefinite programming.

A

Review of convex analysis and optimization

In this section, we review relevant concepts from convex analysis and optimization. For more details, see [17, 13, 16, 121].

A.1   Convex analysis

In this section, we review extended-value convex functions, Fenchel conjugates and polar sets.

Extended-value convex functions. In this paper, we consider functions defined on $\mathbb{R}^p$ with values in $\mathbb{R} \cup \{+\infty\}$, and the domain of f is defined to be the set of vectors in $\mathbb{R}^p$ such that f has finite values. Such an "extended-value" function is said to be convex if its domain is convex and f restricted to its domain (which is a real-valued function) is convex.

Throughout this paper, we denote $w \mapsto I_C(w)$ the indicator function of the convex set C, defined as 0 for $w \in C$ and $+\infty$ otherwise; this defines a convex function and allows constrained optimization problems to be treated as unconstrained optimization problems. In this paper, we always assume that f is a *proper* function (i.e., has non-empty domain).

A function is said closed if for all $\alpha \in \mathbb{R}$, the set $\{w \in \mathbb{R}^p, \ f(w) \leqslant \alpha\}$ is a closed set. We only consider *closed* functions in this paper.

Fenchel conjugate. For any function $f : \mathbb{R}^p \to \mathbb{R} \cup \{+\infty\}$, we may define the *Fenchel conjugate* f^* as the extended-value function from $\mathbb{R}^p$ to $\mathbb{R} \cup \{+\infty\}$ defined as

$$f^*(s) = \sup_{w \in \mathbb{R}^p} w^\top s - f(w). \tag{A.1}$$

As a pointwise supremum of linear functions, f^* is always convex (even if f is not), and it is always closed. Moreover, if f is convex and closed, then the biconjugate of f (i.e., f^{**}) is equal to f, i.e., for all $w \in \mathbb{R}^p$,

$$f(w) = \sup_{s \in \mathbb{R}^p} w^\top s - f^*(s).$$

If f is not convex and closed, then the bi-conjugate is always a lower-bound on f, i.e., for all $w \in \mathbb{R}^p$, $f^{**}(w) \leqslant f(w)$, and it is the tightest such convex closed lower bound, often referred to as the *convex envelope* (see an example in Section 2.3).

When f is convex and closed, many properties of f may be seen from f^* and vice-versa:

- f is strictly convex if and only if f^* is differentiable in the interior of its domain,
- f is μ-strongly convex (i.e., the function $w \mapsto f(w) - \frac{\mu}{2}\|w\|_2^2$ is convex) if and only if f^* has Lipschitz-continuous gradients (with constant $1/\mu$) in the interior of its domain.

Support function. Given a convex closed set C, the support function of C is the Fenchel conjugate of I_C, defined as:

$$\forall s \in \mathbb{R}^p, \ I_C^*(s) = \sup_{w \in C} w^\top s.$$

It is always a positively homogeneous proper closed convex function. Moreover, if f is a positively homogeneous proper closed convex function, then f^* is the indicator function of a closed convex set.

Proximal problems and duality.   In this paper, we will consider minimization problems of the form

$$\min_{w\in\mathbb{R}^p} \frac{1}{2}\|w - z\|_2^2 + f(w),$$

where f is a positively homogeneous proper closed convex function (with C being a convex closed set such that $f^* = I_C$). We then have

$$
\begin{aligned}
\min_{w\in\mathbb{R}^p} \frac{1}{2}\|w - z\|_2^2 + f(w) \;&=\; \min_{w\in\mathbb{R}^p} \max_{s\in C} \frac{1}{2}\|w - z\|_2^2 + w^\top s \\
&=\; \max_{s\in C} \min_{w\in\mathbb{R}^p} \frac{1}{2}\|w - z\|_2^2 + w^\top s \\
&=\; \max_{s\in C} \frac{1}{2}\|z\|_2^2 - \frac{1}{2}\|s - z\|_2^2,
\end{aligned}
$$

where the unique minima of the two problems are related through $w = s - z$. Note that the inversion of the maximum and minimum were made possible because strong duality holds in this situation (f has domain equal to $\mathbb{R}^p$). Thus the original problem is equivalent to an orthogonal projection on C. See applications and extensions to more general separable functions (beyond quadratic) in Section 5.

Polar sets.   Given a subset C of $\mathbb{R}^p$, the polar set of C is denoted C° and defined as:

$$C^\circ = \{s \in \mathbb{R}^p, \; \forall w \in C, w^\top s \leqslant 1\}.$$

For any C, the polar set C° is a closed convex set that contains zero in its interior. If C satifies itself these properties, then $C^{\circ\circ} = C$ (more generally, $C^{\circ\circ}$ is the closure of the convex hull of $C \cup \{0\}$). Thus, the polar operation is a bijection between polar convex sets that contain zero in their interior.

Given a set C, the support function f of C (i.e., the Fenchel conjugate of I_C) is such that C° is the set $\{w \in \mathbb{R}^p, f(w) \leqslant 1\}$. In the context of norms, i.e., when f is a norm, then C is the unit ball of the dual norm (the dual norm of f, is equal to Fenchel-conjugate of the indicator function of its unit ball, to be distinguished from the Fenchel-conjugate of f); the two unit balls are then polar to each other.

A.2 Convex optimization

In this section, we consider several iterative optimization algorithms dedicated to minimizing a convex function f defined on $\mathbb{R}^p$ (potentially with infinite values). See also Section 5.1 for a quick review of proximal methods.

Subgradient descent. A subgradient of a convex function f at $x \in \mathbb{R}^p$, is any vector g such that for all $y \in \mathbb{R}^p$, $f(y) \geqslant f(x) + g^\top(y - x)$. If we assume that f is Lipschitz-continuous (with Lipschitz constant B) on the ℓ_2-ball of radius D (which is assume to be included in the interior of the domain of f), then the subgradient descent algorithm consists of (a) starting from any x_0 such that $\|x_0\|_2 \leqslant D$ and (b) iterating the recursion

$$x_t = \Pi_D(x_{t-1} - \gamma_t g_{t-1}),$$

where g_{t-1} is any subgradient of f at x_{t-1} (with our assumption, such g_t always exists), and Π_D the orthogonal projection on the ℓ_2-ball of center zero and radius D.

If we denote $f^* = \min_{\|x\|_2 \leqslant D} f(x)$, then with $\gamma_t = \frac{D}{B\sqrt{t}}$, we have for all $t > 0$, the convergence rates

$$0 \leqslant \min_{u \in \{0,\ldots,t\}} f(x_u) - f^* \leqslant \frac{4DB}{\sqrt{t}}.$$

The following proposition shows that we may also get a certificate of optimality with similar guarantee.

Proposition A.1. Let f be a convex function f defined on $\mathbb{R}^n$. We assume that f is Lipschitz-continuous on K (with diameter D), with constant B. Let x_t be the t-th iterate of subgradient descent with constants $\gamma_t = \frac{D}{B\sqrt{2t}}$, and $\bar{y}_t = \frac{1}{t}\sum_{u=0}^{t-1} g_u$. Then

$$0 \leqslant \min_{u \in \{0,\ldots,t\}} f(x_u) - f^* \leqslant f^* + f^*(\bar{y}_t) + \max_{x \in K} -\bar{y}_t^\top x \leqslant \frac{DB\sqrt{2}}{\sqrt{t}}.$$

Proof. Let f^* be the Fenchel conjugate of f, defined as $f^*(y) = \max_{x \in \mathbb{R}^n} x^\top y - f(x)$. We denote by g the support function of K, i.e.,

$g(y) = \max_{x \in K} x^\top y$. We then have

$$\min_{x \in K} f(x) \quad = \quad \min_{x \in K} \max_{y \in \mathbb{R}^n} x^\top y - f^*(y) = \max_{y \in \mathbb{R}^n} -f^*(y) - g(-y).$$

We consider the non-negative real number $\mathrm{gap}(x, y) = f(x) + f^*(y) + g(-y)$. We consider the following subgradient descent iteration

$$x_t = \Pi_K(x_{t-1} - \gamma_t y_{t-1}) \text{ with } y_{t-1} \in \partial f(x_{t-1}),$$

where Π_K is the orthogonal projection on K and $\partial f(x_{t-1})$ is the sub-differential of f at x_{t-1}.

Following standard arguments, we get for any $x \in K$ (using the contractivity of orthogonal projections):

$$\|x_t - x\|^2 \quad \leqslant \quad \|x_{t-1} - x\|^2 + \gamma_t^2 \|y_{t-1}\|^2 - 2\gamma_t(x_{t-1} - x)^\top y_{t-1},$$

leading to

$$(x_{t-1} - x)^\top y_{t-1} \leqslant \frac{\|x_{t-1} - x\|^2 - \|x_t - x\|^2 + \gamma_t^2 B^2}{2\gamma_t}.$$

Thus, summing from $t = 1$ to T, we obtain (by summing by parts):

$$\sum_{t=1}^{T}(x_{t-1} - x)^\top y_{t-1} \quad \leqslant \quad \frac{B^2}{2}\sum_{t=1}^{T}\gamma_t + \frac{1}{2}\sum_{t=1}^{T}\gamma_t^{-1}(\|x_{t-1} - x\|^2 - \|x_t - x\|^2)$$

$$= \quad \frac{B^2}{2}\sum_{t=1}^{T}\gamma_t + \frac{1}{2}\sum_{t=1}^{T-1}\|x_t - x\|^2(\gamma_{t+1}^{-1} - \gamma_t^{-1})$$

$$+ \frac{1}{2}\gamma_1^{-1}\|x_0 - x\|^2 - \frac{1}{2}\gamma_T^{-1}\|x_T - x\|^2.$$

If we further assume that γ_t is non-increasing and that $D = \mathrm{diam}(K)$, we get

$$\sum_{t=1}^{T}(x_{t-1} - x)^\top y_{t-1} \quad \leqslant \quad \frac{B^2}{2}\sum_{t=1}^{T}\gamma_t + \frac{1}{2}\sum_{t=1}^{T-1}D^2(\gamma_{t+1}^{-1} - \gamma_t^{-1}) + \frac{1}{2}\gamma_1^{-1}D^2$$

$$= \quad \frac{B^2}{2}\sum_{t=1}^{T}\gamma_t + \frac{D^2}{2\gamma_T}.$$

This leads to, using $f(x) \geqslant f(x_{t-1}) + y_{t-1}^\top(x - x_{t-1})$,

$$\frac{1}{T}\sum_{t=1}^{T}\big[f(x_{t-1}) - f(x)\big] \leqslant \frac{B^2}{2T}\sum_{t=1}^{T}\gamma_t + \frac{D^2}{2T\gamma_T}.$$

We may now apply this to x^* any minimizer of f on K, to get the two usual bounds

$$f\left(\frac{1}{T}\sum_{t=1}^{T}x_{t-1}\right) - f(x^*) \;\leqslant\; \frac{B^2}{2T}\sum_{t=1}^{T}\gamma_t + \frac{D^2}{2T\gamma_T}$$

$$\min_{t\in\{1,\dots,T\}} f(x_{t-1}) - f(x^*) \;\leqslant\; \frac{B^2}{2T}\sum_{t=1}^{T}\gamma_t + \frac{D^2}{2T\gamma_T}.$$

We now denote $\bar{x}_t = \frac{1}{t}\sum_{u=0}^{t-1}x_u$ and $\bar{y}_t = \frac{1}{t}\sum_{u=0}^{t-1}y_u$. We have:

$$f^*(\bar{y}_T) + g(-\bar{y}_T)$$

$$\leqslant\; \frac{1}{T}\sum_{t=1}^{T} f^*(y_{t-1}) + g(-\bar{y}_T) \text{ by convexity of } f$$

$$=\; \frac{1}{T}\sum_{t=1}^{T}\left[-f(x_{t-1}) + x_{t-1}^\top y_{t-1}\right] + g(-\bar{y}_T)$$

$$\text{because } x_{t-1}, y_{t-1} \text{ are Fenchel-dual}$$

$$=\; -\frac{1}{T}\sum_{t=1}^{T} f(x_{t-1}) + \frac{1}{T}\sum_{t=1}^{T} x_{t-1}^\top y_{t-1} - \bar{y}_T^\top x \text{ for a certain } x \in K$$

$$=\; -\frac{1}{T}\sum_{t=1}^{T} f(x_{t-1}) + \frac{1}{T}\sum_{t=1}^{T}(x_{t-1} - x)^\top y_{t-1}$$

$$\leqslant\; -\frac{1}{T}\sum_{t=1}^{T} f(x_{t-1}) + \frac{B^2}{2T}\sum_{t=1}^{T}\gamma_t + \frac{D^2}{2T\gamma_T}.$$

This leads to

$$\mathrm{gap}(\bar{x}_T, \bar{y}_T) = f(\bar{x}_T) + f^*(\bar{y}_T) + g(-\bar{y}_T) \leqslant \frac{B^2}{2T}\sum_{t=1}^{T}\gamma_t + \frac{D^2}{2T\gamma_T}.$$

With $\gamma_t = \frac{\alpha D}{B\sqrt{t}}$, we obtain an upper bound

$$
\begin{aligned}
\frac{B^2}{2T} \sum_{t=1}^{T} \gamma_t + \frac{D^2}{2T\gamma_T} &\leqslant \frac{DB}{2}\left[\alpha \frac{1}{T} \sum_{t=1}^{T} \frac{1}{\sqrt{t}} + \frac{1}{\alpha}\sqrt{T}\right] \\
&\leqslant \frac{DB}{2}\left[\alpha \frac{2}{T} \sum_{t=1}^{T} \left(\sqrt{t} - \sqrt{t-1}\right) + \frac{1}{\alpha}\sqrt{T}\right] \\
&\leqslant \frac{DB}{2}\left[\alpha \frac{2}{\sqrt{T}} + \frac{1}{\alpha}\sqrt{T}\right] \\
&\leqslant \frac{DB\sqrt{2}}{\sqrt{T}} \text{ with } \alpha = \frac{1}{\sqrt{2}}.
\end{aligned}
$$

$\square$

If we further assume that f is strongly convex with constant μ, (i.e., $x \mapsto f(x) - \frac{\mu}{2}\|x\|_2^2$ is convex), then by taking $\gamma_t = \frac{1}{\mu t}$, we have, for all $t > 0$ [127],

$$
0 \leqslant \min_{u \in \{0,\dots,t\}} f(x_u) - f^* \leqslant \frac{B^2}{2\mu t} \frac{1 + \log t}{t}.
$$

Conditional gradient descent. We now assume that the function f is differentiable on a compact convex set $K \subset \mathbb{R}^p$ (with diameter D), and that its gradient is Lipschitz-continuous with constant L. We consider the following conditional gradient algorithm, which is applicable when linear functions may be maximized efficiently over K.

(1) **Initialization**: Choose any $x_0 \in K$, compute a minimizer $x_1 \in K$ of $f'(x_0)^\top x$.

(2) **Iteration**: iterate until upper bound ε on duality gap is reached:

    (a) $\bar{x}_{t-1} \in \operatorname{argmin}_{x \in K} f'(x_{t-1})^\top x$,

    (b) Compute upper bound on gap: $(x_{t-1} - \bar{x}_{t-1})' f'(x_{t-1})$

    (c) Compute $\omega_{t-1} = \min\left\{1, \frac{f'(x_{t-1})^\top (x_{t-1} - \bar{x}_{t-1})}{LD^2}\right\}$

    (d) Take $x_t = x_{t-1} + \omega_{t-1}(\bar{x}_{t-1} - x_{t-1})$.

Step (2)(a) corresponds to minimizing the first order Taylor expansion at x_{t-1}, while step (2)(c) corresponds to performing approximate line-search on the segment $[x_{t-1}, \bar{x}_{t-1}]$. Combining the analysis of [40] and [72], we have the following proposition:

Proposition A.2. For the previous algorithm with have: $f(x_t) - \min_{x \in K} f(x) \leqslant \frac{LD^2}{t+1}$. Moreover, there exists at least one $k \in [2t/3, t]$ such that $\max_{x \in K}(x_k - x)^\top f'(x_k) \leqslant \frac{2LD^2}{t}$, i.e., the primal-dual pair $(x_k, f'(x_k))$ is a certificate of optimality ensuring at least an approximate optimality of $\frac{9LD^2}{2t}$.

Proof. Let $g(z) = \max_{x \in K}(z - x)^\top f'(z)$. It is a certificate of duality for $z \in K$. We denote $\Delta_t = f(x_t) - \min_{x \in K} f(x)$. We have $0 \geqslant \Delta_t \leqslant g(x_t)$. Moreover, following [40], we have $\Delta_1 \leqslant \frac{LD^2}{2}$ and

$$
\begin{aligned}
\Delta_t &\leqslant \Delta_{t-1} + f'(x_{t-1})^\top (x_t - x_{t-1}) + \frac{L}{2} \|x_t - x_{t-1}\|_2^2 \\
&= \Delta_{t-1} + \omega_{t-1} f'(x_{t-1})^\top (\bar{x}_{t-1} - x_{t-1}) + \frac{L\omega_{t-1}^2}{2} \|\bar{x}_{t-1} - x_{t-1}\|_2^2 \\
&\leqslant \Delta_{t-1} - \omega_{t-1} g(x_{t-1}) + \frac{LD^2 \omega_{t-1}^2}{2} \\
&\leqslant \Delta_{t-1} - \frac{1}{2} \min \left\{ \frac{g(x_{t-1})^2}{LD^2}, g(x_{t-1}) \right\}.
\end{aligned}
$$

This implies that Δ_t is non-increasing, and thus $\Delta_t \leqslant \Delta_1 \leqslant \frac{LD^2}{2}$. This implies, using $\Delta_{t-1} \leqslant g(x_{t-1})$:

$$
\Delta_t \leqslant \Delta_{t-1} - \frac{1}{2LD^2} \Delta_{t-1}^2.
$$

By dividing by $\Delta_t \Delta_{t-1}$, we get:

$$
\Delta_{t-1}^{-1} \leqslant \Delta_t^{-1} - \frac{1}{2LD^2},
$$

and thus $\frac{2}{LD^2} \leqslant \Delta_1^{-1} \leqslant \Delta_t^{-1} - \frac{t-1}{2LD^2}$, which implies for any $t \geqslant 1$,

$$
\Delta_t \leqslant \frac{2LD^2}{t+3} \leqslant \frac{2LD^2}{t}.
$$

Let us now assume that for all $u \in \{\alpha t, \ldots, t\}$, then $g(x_u) \geqslant \frac{\beta L D^2}{\alpha t + 3}$. We then have

$$
\begin{aligned}
\Delta_t \quad \leqslant \quad & \frac{2LD^2}{\alpha t + 3} - \frac{LD^2}{2} \sum_{u=\alpha t}^{t-1} \frac{\beta^2}{(\alpha t + 3)^2} \\
\leqslant \quad & \frac{2LD^2}{\alpha t + 3} - \frac{\beta^2 LD^2 t(1 - \alpha)}{2(\alpha t + 3)^2}
\end{aligned}
$$

With $\alpha = 2/3$ and $\beta = 3$, we obtain that $\Delta_t < 0$, which is a contradiction. This leads to the desired result. $\qquad\square$

B

Miscellaneous results on submodular functions

B.1 Conjugate functions

The next proposition computes the Fenchel conjugate of the Lovász extensions restricted to $[0,1]^p$, noting that by Prop. 4.1, the regular Fenchel conjugate of the unrestricted Lovász extension is the indicator function of the base polyhedron (for a definition of Fenchel conjugates, see [17, 16] and Appendix A). This allows a form of conjugacy between set-functions and convex functions (see more details in [49]).

Proposition B.1. (Conjugate of a submodular function) Let F be a submodular function such that $F(\varnothing) = 0$. The conjugate $\tilde{f}:$ $\mathbb{R}^p \to \mathbb{R}$ of F is defined as $\tilde{f}(s) = \max_{A \subset V} s(A) - F(A)$. Then, the conjugate function $\tilde{f}$ is convex, and is equal to the Fenchel-conjugate of the Lovász extension restricted to $[0,1]^p$. Moreover, for all $A \subset V$, $F(A) = \max_{s \in \mathbb{R}^p} s(A) - \tilde{f}(s)$.

Proof. The function $\tilde{f}$ is a maximum of linear functions and thus it is convex. We have for $s \in \mathbb{R}^p$:

$$\max_{w \in [0,1]^p} w^\top s - f(w) = \max_{A \subset V} s(A) - F(A) = \tilde{f}(s),$$

because $F - s$ is submodular and because of Prop. 2.4, which leads to first the desired result. The last assertion is a direct consequence of the fact that $F(A) = f(1_A)$. □

B.2   Operations that preserve submodularity

In this section, we present several ways of building submodular functions from existing ones. For all of these, we describe how the Lovász extensions and the submodular polyhedra are affected. Note that in many cases, operations are simpler in terms of submodular and base polyhedra. Many operations such as projections onto subspaces may be interpreted in terms of polyhedra corresponding to other submodular functions.

We have seen in Section 3.5 that given any submodular function F, we may define $G(A) = F(A) + F(V \backslash A) - F(V)$. Then G is always submodular and symmetric (and thus non-negative, see Section 7.4). This symmetrization can be applied to any submodular function and in the example of Section 3, they often lead to interesting new functions. We now present other operations that preserve submodularity.

Proposition B.2. (Restriction of a submodular function) let F be a submodular function such that $F(\varnothing) = 0$ and $A \subset V$. The restriction of F on A, denoted F_A is a set-function on A defined as $F_A(B) = F(B)$ for $B \subset A$. The function f_A is submodular. Moreover, if we can write the Lovász extension of F as $f(w) = f(w_A, w_{V \backslash A})$, then the Lovász extension of F_A is $f_A(w_A) = f(w_A, 0)$. Moreover, the submodular polyhedron $P(F_A)$ is simply the projection of $P(F)$ on the components indexed by A, i.e., $s \in P(F_A)$ if and only if $\exists t$ such that $(s, t) \in P(F)$.

Proof. Submodularity and the form of the Lovász extension are straightforward from definitions. To obtain the submodular polyhedron, notice that we have $f_A(w_A) = f(w_A, 0) = \max_{(s,t) \in P(F)} w_A^\top s + 0^\top t$, which implies the desired result, this shows that the Fenchel-conjugate of the Lovász extensions is the indicator function of a polyhedron. □

Proposition B.3. (Contraction of a submodular function) let F be a submodular function such that $F(\varnothing) = 0$ and $A \subset V$. The contraction of F on A, denoted F^A is a set-function on $V \backslash A$ defined as $F^A(B) = F(A \cup B) - F(A)$ for $B \subset V \backslash A$. The function F^A is submodular. Moreover, if we can write the Lovász extension of F as $f(w) = f(w_A, w_{V \backslash A})$, then the Lovász extension of F^A is $f^A(w_{V \backslash A}) = f(1_A, w_{V \backslash A}) - F(A)$. Moreover, the submodular polyhedron $P(F^A)$ is simply the projection of $P(F) \cap \{s(A) = F(A)\}$ on the components indexed by $V \backslash A$, i.e., $t \in P(F^A)$ if and only if $\exists s \in P(F) \cap \{s(A) = F(A)\}$, such that $s_{V \backslash A} = t$.

Proof. Submodularity and the form of the Lovász extension are straightforward from definitions. Let $t \in \mathbb{R}^{|V \backslash A|}$. If $\exists s \in P(F) \cap \{s(A) = F(A)\}$, such that $s_{V \backslash A} = t$, then we have for all $B \subset V \backslash A$, $t(B) = t(B) + s(A) - F(A) \leqslant F(A \cup B) - F(A)$, and hence $t \in P(F^A)$. If $t \in P(F^A)$, then take any $v \in B(F_A)$ and concatenate v and t into s. Then, for all subsets $C \subset V$, $s(C) = s(C \cap A) + s(C \cap (V \backslash A)) = v(C \cap A) + t(C \cap (V \backslash A)) \leqslant F(C \cap A) + F(A \cup (C \cap (V \backslash A))) - F(A) = F(C \cap A) + F(A \cup C) - F(A) \leqslant F(C)$ by submodularity. Hence $s \in P(F)$. $\square$

The next proposition shows how to build a new submodular function from an existing one, by partial minimization. Note the similarity (and the difference) between the submodular polyhedra for a partial minimum (Prop. B.4) and for the restriction defined in Prop. B.2.

Note also that contrary to convex functions, the pointwise maximum of two submodular functions is not in general submodular (as can be seen by considering functions of the cardinality from Section 3.1).

Proposition B.4. (Partial minimum of a submodular function) We consider a submodular function G on $V \cup W$, where $V \cap W = \varnothing$ (and $|W| = q$), with Lovász extension $g : \mathbb{R}^{p+q} \to \mathbb{R}$. We consider, for $A \subset V$, $F(A) = \min_{B \subset W} G(A \cup B) - \min_{B \subset W} G(B)$. The set-function F is submodular and such that $F(\varnothing) = 0$. Its Lovász extension is such that for all $w \in [0, 1]^p$, $f(w) = \min_{v \in [0,1]^q} g(w, v) - \min_{v \in [0,1]^q} g(0, v)$.

Moreover, if $\min_{B \subset W} G(B) = 0$, we have for all $w \in \mathbb{R}^p_+$, $f(w) = \min_{v \in \mathbb{R}^q_+} g(w, v)$, and the submodular polyhedron $P(F)$ is the set of $s \in \mathbb{R}^p$ such that there exists $t \in \mathbb{R}^q_+$, such that $(s, t) \in P(G)$.

Proof. Define $c = \min_{B \subset W} G(B)$, which is independent of A. We have, for $A, A' \subset V$, and any $B, B' \subset W$, by definition of F:

$$\begin{aligned}
& F(A \cup A') + F(A \cap A') \\
\leqslant\ & -2c + G([A \cup A'] \cup [B \cup B']) + G([A \cap A'] \cup [B \cap B']) \\
=\ & -2c + G([A \cup B] \cup [A' \cup B']) + G([A \cup B] \cap [A' \cup B']) \\
\leqslant\ & -2c + G(A \cup B) + G(A' \cup B') \text{ by submodularity.}
\end{aligned}$$

Minimizing with respect to B and B' leads to the submodularity of F.

Following Prop. B.1, we can get the conjugate function $\tilde{f}$ from the one $\tilde{g}$ of G. For $s \in \mathbb{R}^p$, we have, by definition, $\tilde{f}(s) = \max_{A \subset V} s(A) - F(A) = \max_{A \cup B \subset V \cup W} s(A) + c - G(A \cup B) = c + \tilde{g}(s, 0)$. We thus get from Prop. B.1 that for $w \in [0, 1]^p$,

$$\begin{aligned}
f(w) &= \max_{s \in \mathbb{R}^p} w^\top s - \tilde{f}(s) \\
&= \max_{s \in \mathbb{R}^p} w^\top s - \tilde{g}(s, 0) - c \\
&= \max_{s \in \mathbb{R}^p} \min_{(\tilde{w}, v) \in [0,1]^{p+q}} w^\top s - \tilde{w}^\top s + g(\tilde{w}, v) - c \\
&\qquad \text{by applying Prop. B.1,} \\
&= \min_{(\tilde{w}, v) \in [0,1]^{p+q}} \max_{s \in \mathbb{R}^p} w^\top s - \tilde{w}^\top s + g(\tilde{w}, v) - c \\
&= \min_{v \in [0,1]^q} g(w, v) - c \text{ by maximizing with respect to } s.
\end{aligned}$$

Note that $c = \min_{B \subset W} G(B) = \min_{v \in [0,1]^q} g(0, v)$.

For any $w \in \mathbb{R}^p_+$, for any $\lambda \geqslant \|w\|_\infty$, we have $w/\lambda \in [0, 1]^p$, and thus

$$\begin{aligned}
f(w) &= \lambda f(w/\lambda) = \min_{v \in [0,1]^q} \lambda g(w/\lambda, v) - c\lambda = \min_{v \in [0,1]^q} g(w, \lambda v) - c\lambda \\
&= \min_{v \in [0,\lambda]^q} g(w, v) - c\lambda.
\end{aligned}$$

Thus, if $c = 0$, we have $f(w) = \min_{v \in \mathbb{R}^q_+} g(w, v)$, by letting $\lambda \to +\infty$.

We then also have:

$$
\begin{aligned}
f(w) &= \min_{v\in\mathbb{R}_+^q} g(w,v) = \min_{v\in\mathbb{R}_+^q} \max_{(s,t)\in P(G)} w^\top s + v^\top t \\
&= \max_{(s,t)\in P(G),\ t\in\mathbb{R}_+^q} w^\top s.
\end{aligned}
$$

$\square$

The following propositions give an interpretation of the intersection between the submodular polyhedron and sets of the form $\{s \leqslant z\}$ and $\{s \geqslant z\}$. Prop. B.5 notably implies that for all $z \in \mathbb{R}^p$, we have: $\min_{B\subset V} F(B) + z(V\backslash B) = \max_{s\in P(F),\ s\leqslant z} s(V)$, which implies the second statement of Prop. 7.3 for $z = 0$.

Proposition B.5. (Convolution of a submodular function and a modular function) Let F be a submodular function such that $F(\varnothing) = 0$ and $z \in \mathbb{R}^p$. Define $G(A) = \min_{B\subset A} F(B) + z(A\backslash B)$. Then G is submodular, satisfies $G(\varnothing) = 0$, and the submodular polyhedron $P(G)$ is equal to $P(F)\cap\{s \leqslant z\}$. Moreover, for all $A \subset V$, $G(A) \leqslant F(A)$ and $G(A) \leqslant z(A)$.

Proof. Let $A, A' \subset V$, and B, B' the corresponding minimizers defining $G(A)$ and $G(A')$. We have:

$$G(A) + G(A')$$
$$= F(B) + z(A\backslash B) + F(B') + z(A'\backslash B')$$
$$\geqslant F(B\cup B') + F(B\cap B') + z(A\backslash B) + z(A'\backslash B') \text{ by submodularity,}$$
$$= F(B\cup B') + F(B\cap B') + z([A\cup A']\backslash[B\cup B']) + z([A\cap A']\backslash[B\cap B'])$$
$$\geqslant G(A\cup A') + G(A\cap A') \text{ by definition of } G,$$

hence the submodularity of G. If $s \in P(G)$, then $\forall B \subset A \subset V$, $s(A) \leqslant G(A) \leqslant F(B) + z(A\backslash B)$. Taking $B = A$, we get that $s \in P(F)$; from $B = \varnothing$, we get $s \leqslant z$, and hence $s \in P(F)\cap\{s \leqslant z\}$. If $s \in P(F)\cap\{s \leqslant z\}$, for all $\forall B \subset A \subset V$, $s(A) = s(A\backslash B) + s(B) \leqslant z(A\backslash B) + F(B)$; by minimizing with respect to B, we get that $s \in P(G)$.

We get $G(A) \leqslant F(A)$ by taking $B = A$ in the definition of $G(A)$, and we get $G(A) \leqslant z(A)$ by taking $B = \varnothing$. $\square$

Proposition B.6. (Monotonization of a submodular function)
Let F be a submodular function such that $F(\varnothing) = 0$. Define $G(A) = \min_{B \supset A} F(B) - \min_{B \subset V} F(B)$. Then G is submodular such that $G(\varnothing) = 0$, and the base polyhedron $B(G)$ is equal to $B(F) \cap \{s \geqslant 0\}$. Moreover, G is non-decreasing, and for all $A \subset V$, $G(A) \leqslant F(A)$.

Proof. Let $c = \min_{B \subset V} F(B)$. Let $A, A' \subset V$, and B, B' the corresponding minimizers defining $G(A)$ and $G(A')$. We have:

$$
\begin{aligned}
G(A) + G(A') &= F(B) + F(B') - 2c \\
&\geqslant F(B \cup B') + F(B \cap B') - 2c \text{ by submodularity} \\
&\geqslant G(A \cup A') + G(A \cap A') \text{ by definition of } G,
\end{aligned}
$$

hence the submodularity of G. It is obviously non-decreasing. We get $G(A) \leqslant F(A)$ by taking $B = A$ in the definition of $G(A)$. Since G is increasing, $B(G) \subset \mathbb{R}_+^p$ (because all of its extreme points, obtained by the greedy algorithm, are in $\mathbb{R}_+^p$). By definition of G, $B(G) \subset B(F)$. Thus $B(G) \subset B(F) \cap \mathbb{R}_+^p$. The opposite inclusion is trivial from the definition.

$\square$

Acknowledgements

References

[1] S. Ahmed and A. Atamtürk. Maximizing a class of submodular utility functions. *Mathematical Programming: Series A and B*, 128(1-2):149–169, 2011.

[2] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network flows: theory, algorithms, and applications*. Prentice hall, 1993.

[3] T. Ando. Concavity of certain maps on positive definite matrices and applications to Hadamard products. *Linear Algebra and its Applications*, 26:203–241, 1979.

[4] F. Bach. Consistency of the group Lasso and multiple kernel learning. *Journal of Machine Learning Research*, 9:1179–1225, 2008.

[5] F. Bach. Exploring large feature spaces with hierarchical multiple kernel learning. In *Adv. NIPS*, 2008.

[6] F. Bach. Structured sparsity-inducing norms through submodular functions. In *Adv. NIPS*, 2010.

[7] F. Bach. Shaping level sets with submodular functions. In *Adv. NIPS*, 2011.

[8] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Optimization with sparsity-inducing penalties. *Foundations and Trends® in Machine Learning*, 2011. To appear.

[9] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Structured sparsity through convex optimization. Technical Report 00621245, HAL, 2011.

[10] R. G. Baraniuk, V. Cevher, M. F. Duarte, and C. Hegde. Model-based compressive sensing. *IEEE Transactions on Information Theory*, 56:1982–2001, 2010.

[11] 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.

[12] S. Becker, J. Bobin, and E. Candes. Nesta: A fast and accurate first-order method for sparse recovery. *SIAM J. on Imaging Sciences*, 4(1):1–39, 2011.

[13] D. Bertsekas. *Nonlinear programming*. Athena Scientific, 1995.

[14] M. J. Best and N. Chakravarti. Active set algorithms for isotonic regression; a unifying framework. *Mathematical Programming*, 47(1):425–439, 1990.

[15] E. Boros and P.L. Hammer. Pseudo-Boolean optimization. *Discrete Applied Mathematics*, 123(1-3):155–225, 2002.

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

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

[18] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. PAMI*, 23(11):1222–1239, 2001.

[19] J.F. Cardoso. Dependence, correlation and gaussianity in independent component analysis. *The Journal of Machine Learning Research*, 4:1177–1203, 2003.

[20] V. Cevher, M. F. Duarte, C. Hegde, and R. G. Baraniuk. Sparse signal recovery using Markov random fields. In *Adv. NIPS*, 2008.

[21] A. Chambolle and J. Darbon. On total variation minimization and surface evolution using parametric maximum flows. *International Journal of Computer Vision*, 84(3):288–307, 2009.

[22] G. Charpiat. Exhaustive family of energies minimizable exactly by a graph cut. In *Proc. CVPR*, 2011.

[23] C. Chekuri, J. Vondrák, and R. Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. Technical Report 1105.4593, Arxiv, 2011.

[24] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998.

[25] S. Chopra. On the spanning tree polyhedron. *Operations Research Letters*, 8(1):25–29, 1989.

[26] G. Choquet. Theory of capacities. *Ann. Inst. Fourier*, 5:131–295, 1954.

[27] F.R.K. Chung. *Spectral graph theory*. Amer. Mathematical Society, 1997.

[28] P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. In *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*. Springer, 2010.

[29] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. MIT Press, 1989.

[30] G. Cornuejols, M. Fisher, and G.L. Nemhauser. On the uncapacitated location problem. *Annals of Discrete Mathematics*, 1:163–177, 1977.

[31] G. Cornuejols, M.L. Fisher, and G.L. Nemhauser. Location of bank accounts to optimize float: An analytic study of exact and approximate algorithms. *Management Science*, 23(8):789–810, 1977.

[32] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 1991.

[33] T.M. Cover, J.A. Thomas, and MyiLibrary. *Elements of information theory*, volume 6. Wiley Online Library, 1991.

[34] W.H. Cunningham. Testing membership in matroid polyhedra. *Journal of Combinatorial Theory, Series B*, 36(2):161–188, 1984.

[35] W.H. Cunningham. Minimum cuts, modular functions, and matroid polyhedra. *Networks*, 15(2):205–215, 1985.

[36] A. Das and D. Kempe. Algorithms for subset selection in linear regression. In *Proceedings of the 40th annual ACM symposium on Theory of computing*. ACM, 2008.

[37] A. Das and D. Kempe. Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. In *Proc. ICML*, 2011.

[38] B.A. Davey and H.A. Priestley. *Introduction to Lattices and Order*. Cambridge Univ. Press, 2002.

[39] 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.

[40] J. C. Dunn. Convergence rates for conditional gradient sequences generated by implicit step length rules. *SIAM Journal on Control and Optimization*, 18:473–487, 1980.

[41] J. C. Dunn and S. Harshbarger. Conditional gradient algorithms with open loop step size rules. *Journal of Mathematical Analysis and Applications*, 62(2):432–444, 1978.

[42] J. Edmonds. Submodular functions, matroids, and certain polyhedra. In *Combinatorial optimization - Eureka, you shrink!*, pages 11–26. Springer, 2003.

[43] V.V. Fedorov. *Theory of optimal experiments*. Academic press, 1972.

[44] U. Feige. A threshold of ln$n$ for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998.

[45] U. Feige. On maximizing welfare when utility functions are subadditive. In *Proc. ACM symposium on Theory of computing*, pages 41–50, 2006.

[46] U. Feige, V.S. Mirrokni, and J. Vondrak. Maximizing non-monotone submodular functions. In *Proc. Symposium on Foundations of Computer Science*, pages 461–471. IEEE Computer Society, 2007.

[47] S. Foldes and P. L. Hammer. Submodularity, supermodularity, and higher-order monotonicities of pseudo-Boolean functions. *Mathematics of Operations Research*, 30(2):453–461, 2005.

[48] J. Friedman, T. Hastie, and R. Tibshirani. A note on the group lasso and a sparse group lasso. *preprint*, 2010.

[49] S. Fujishige. *Submodular Functions and Optimization*. Elsevier, 2005.

[50] S. Fujishige and S. Isotani. A submodular function minimization algorithm based on the minimum-norm base. *Pacific Journal of Optimization*, 7:3–17, 2011.

[51] G. Gallo, M.D. Grigoriadis, and R.E. Tarjan. A fast parametric maximum flow algorithm and applications. *SIAM Journal on Computing*, 18(1):30–55, 1989.

[52] A. Gelman. *Bayesian data analysis*. CRC press, 2004.

[53] B. Goldengorin, G. Sierksma, G.A. Tijssen, and M. Tso. The data-correcting algorithm for the minimization of supermodular functions. *Management Science*, pages 1539–1551, 1999.

[54] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, 1996.

[55] H. Groenevelt. Two algorithms for maximizing a separable concave function over a polymatroid feasible region. *European Journal of Operational Research*, 54(2):227–236, 1991.

[56] B. Grünbaum. *Convex polytopes*, volume 221. Springer Verlag, 2003.

[57] Z. Harchaoui and C. Lévy-Leduc. Catching change-points with Lasso. *Adv. NIPS*, 20, 2008.

[58] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer-Verlag, 2001.

[59] J. Haupt and R. Nowak. Signal reconstruction from noisy random projections. *IEEE Transactions on Information Theory*, 52(9):4036–4048, 2006.

[60] E. Hazan and S. Kale. Online submodular minimization. In *Adv. NIPS*, 2009.

[61] D. Heckerman, D. Geiger, and D.M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20(3):197–243, 1995.

[62] D. S. Hochbaum. An efficient algorithm for image segmentation, Markov random fields and related problems. *Journal of the ACM*, 48(4):686–701, 2001.

[63] D. S. Hochbaum and S.P. Hong. About strongly polynomial time algorithms for quadratic optimization over submodular constraints. *Mathematical Programming*, 69(1):269–309, 1995.

[64] T. Hocking, A. Joulin, F. Bach, and J.-P. Vert. Clusterpath: an algorithm for clustering using convex fusion penalties. In *Proc. ICML*, 2011.

[65] H. Hoefling. A path algorithm for the fused Lasso signal approximator. *Journal of Computational and Graphical Statistics*, 19(4):984–1006, 2010.

[66] R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge Univ. Press, 1990.

[67] R. Horst and N.V. Thoai. Dc programming: overview. *Journal of Optimization Theory and Applications*, 103(1):1–43, 1999.

[68] J. Huang, T. Zhang, and D. Metaxas. Learning with structured sparsity. In *Proc. ICML*, 2009.

[69] D.R. Hunter and K. Lange. A tutorial on MM algorithms. *The American Statistician*, 58(1):30–37, 2004.

[70] S. Iwata, L. Fleischer, and S. Fujishige. A combinatorial strongly polynomial algorithm for minimizing submodular functions. *Journal of the ACM*, 48(4):761–777, 2001.

[71] L. Jacob, G. Obozinski, and J.-P. Vert. Group Lasso with overlaps and graph Lasso. In *Proc. ICML*, 2009.

[72] M. Jaggi. Convex optimization without projection steps. Technical Report 1108.1170, Arxiv, 2011.

[73] S. Jegelka, H. Lin, and J. A. Bilmes. Fast approximate submodular minimization. In *Adv. NIPS*, 2011.

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

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

[76] R. Jenatton, J. Mairal, G. Obozinski, and F. Bach. Proximal methods for sparse hierarchical dictionary learning. In *Proc. ICML*, 2010.

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

[78] R. Jenatton, G. Obozinski, and F. Bach. Structured sparse principal component analysis. In *Proc. AISTATS*, 2009.

[79] K. Kavukcuoglu, M. A. Ranzato, R. Fergus, and Y. Le-Cun. Learning invariant features through topographic filter maps. In *Proc. CVPR*, 2009.

[80] Y. Kawahara, K. Nagano, K. Tsuda, and J.A. Bilmes. Submodularity cuts and applications. In *Adv. NIPS 22*, 2009.

[81] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proc. SIGKDD*, 2003.

[82] S. Kim and E. Xing. Tree-guided group Lasso for multi-task regression with structured sparsity. In *Proc. ICML*, 2010.

[83] V. Kolmogorov. Minimizing a sum of submodular functions. Technical Report 1006.1990, Arxiv, 2010.

[84] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004.

[85] A. Krause and V. Cevher. Submodular dictionary selection for sparse representation. In *Proc. ICML*, 2010.

[86] A. Krause and C. Guestrin. Near-optimal nonmyopic value of information in graphical models. In *Proc. UAI*, 2005.

[87] A. Krause and C. Guestrin. Beyond convexity: Submodularity in machine learning, 2008. Tutorial at ICML.

[88] Andreas Krause and Carlos Guestrin. Submodularity and its applications in optimized information gathering. *ACM Transactions on Intelligent Systems and Technology*, 2(4), 2011.

[89] S. L. Lauritzen. *Graphical Models (Oxford Statistical Science Series)*. Oxford University Press, USA, July 1996.

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

[91] H. Lin and J. Bilmes. A class of submodular functions for document summarization. In *North American chapter of the Association for Computational Linguistics/Human Language Technology Conference (NAACL/HLT-2011)*, Portland, OR, June 2011. (long paper).

[92] L. Lovász. Submodular functions and convexity. *Mathematical programming: The state of the art, Bonn*, pages 235–257, 1982.

[93] R. Luss, S. Rosset, and M. Shahar. Decomposing isotonic regression for efficiently solving large problems. In *Adv. NIPS*, volume 23, 2010.

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

[95] J. Mairal, R. Jenatton, G. Obozinski, and F. Bach. Network flow algorithms for structured sparsity. In *Adv. NIPS*, 2010.

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

[97] C. L. Mallows. Some comments on C_p. *Technometrics*, 15:661–675, 1973.

[98] N. Megiddo. Optimal flows in networks with multiple sources and sinks. *Mathematical Programming*, 7(1):97–107, 1974.

[99] M. Minoux. Accelerated greedy algorithms for maximizing submodular set functions. *Optimization Techniques*, pages 234–243, 1978.

[100] J. J. Moreau. Fonctions convexes duales et points proximaux dans un espace Hilbertien. *C. R. Acad. Sci. Paris Sér. A Math.*, 255:2897–2899, 1962.

[101] J.R. Munkres. *Elements of algebraic topology*, volume 2. Addison-Wesley Reading, MA, 1984.

[102] H. Nagamochi and T. Ibaraki. A note on minimizing submodular functions. *Information Processing Letters*, 67(5):239–244, 1998.

[103] K. Nagano. A strongly polynomial algorithm for line search in submodular polyhedra. *Discrete Optimization*, 4(3-4):349–359, 2007.

[104] K. Nagano, Y. Kawahara, and K. Aihara. Size-constrained submodular minimization through minimum norm base. In *Proc. ICML*, 2011.

[105] M. Narasimhan and J. Bilmes. PAC-learning bounded tree-width graphical models. In *Proc. UAI*, 2004.

[106] M. Narasimhan and J. Bilmes. A submodular-supermodular procedure with applications to discriminative structure learning. In *Adv. NIPS*, volume 19, 2006.

[107] M. Narasimhan and J. Bilmes. Local search for balanced submodular clusterings. In *Proc. IJCAI*, 2007.

[108] M. Narasimhan, N. Jojic, and J. Bilmes. Q-clustering. *Adv. NIPS*, 18, 2006.

[109] H. Narayanan. A rounding technique for the polymatroid membership problem. *Linear algebra and its applications*, 221:41–57, 1995.

[110] H. Narayanan. *Submodular Functions and Electrical Networks*. North-Holland, 2009. Second edition.

[111] G.L. Nemhauser, L.A. Wolsey, and M.L. Fisher. An analysis of approximations for maximizing submodular set functions–i. *Mathematical Programming*, 14(1):265–294, 1978.

[112] Y. Nesterov. *Introductory lectures on convex optimization: a basic course.* Kluwer Academic Publishers, 2004.

[113] Y. Nesterov. Gradient methods for minimizing composite objective function. Technical report, Center for Operations Research and Econometrics (CORE), Catholic University of Louvain, 2007.

[114] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 2nd edition, 2006.

[115] G. Obozinski and F. Bach. Convex relaxation of combinatorial penalties. Technical report, HAL, 2011.

[116] J.B. Orlin. A faster strongly polynomial time algorithm for submodular function minimization. *Mathematical Programming*, 118(2):237–251, 2009.

[117] M. Queyranne. Minimizing symmetric submodular functions. *Mathematical Programming*, 82(1):3–12, 1998.

[118] M. Queyranne and A. Schulz. Scheduling unit jobs with compatible release dates on parallel machines with nonstationary speeds. *Integer Programming and Combinatorial Optimization*, 920:307–320, 1995.

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

[120] C. E. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

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

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

[123] A. Schrijver. A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory, Series B*, 80(2):346–355, 2000.

[124] A. Schrijver. *Combinatorial optimization: Polyhedra and efficiency*. Springer, 2004.

[125] M. Seeger. On the submodularity of linear experimental design, 2009. `http://lapmal.epfl.ch/papers/subm_lindesign.pdf`.

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

[127] S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub-gradient solver for svm. In *Proc. ICML*, 2007.

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

[129] P. Sprechmann, I. Ramirez, G. Sapiro, and Y. Eldar. Collaborative hierarchical sparse modeling. In *Conf. Information Sciences and Systems (CISS)*, 2010.

[130] P. Stobbe and A. Krause. Efficient minimization of decomposable submodular functions. In *Adv. NIPS*, 2010.

[131] R. Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B*, pages 267–288, 1996.

[132] 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*, pages 91–108, 2005.

[133] A Toshev. Submodular function minimization. Technical report, University of Pennsylvania, 2010. Written Preliminary Examination.

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

[135] P. Wolfe. Finding the nearest point in a polytope. *Math. Progr.*, 11(1):128–149, 1976.

[136] Laurence A. Wolsey. Maximising real-valued submodular functions: Primal and dual heuristics for location problems. *Mathematics of Operations Research*, 7(3):pp. 410–425, 1982.

[137] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo. Sparse reconstruction by separable approximation. *IEEE Transactions on Signal Processing*, 57(7):2479–2493, 2009.

[138] A.L. Yuille and A. Rangarajan. The concave-convex procedure. *Neural Computation*, 15(4):915–936, 2003.

[139] Z. Zhang and R.W. Yeung. On characterization of entropy function via information inequalities. *IEEE Transactions on Information Theory*, 44(4):1440–1452, 1998.

[140] P. Zhao, G. Rocha, and B. Yu. Grouped and hierarchical model selection through composite absolute penalties. *Annals of Statistics*, 37(6A):3468–3497, 2009.

[141] S. Zivnı, D.A. Cohen, and P.G. Jeavons. The expressive power of binary submodular functions. *Discrete Applied Mathematics*, 157(15):3347–3358, 2009.