Active-set Methods for Submodular Optimization

K. S. Sesh Kumar, Francis Bach

Active-set Methods for Submodular Optimization

K. S. Sesh Kumar
INRIA-Sierra project-team
Département d'Informatique
de l'Ecole Normale Supérieure
Paris, France
sesh-kumar.karri@inria.fr

Francis Bach
INRIA-Sierra project-team
Département d'Informatique
de l'Ecole Normale Supérieure
Paris, France
francis.bach@inria.fr

June 9, 2015

Abstract

We consider submodular optimization problems such as submodular function minimization (SFM) and quadratic problems regularized by the Lovász extension; for cut functions, this corresponds respectively to graph cuts and total variation (TV) denoising. Given a submodular function with an SFM oracle, we propose a new active-set algorithm for total variation denoising, which is more flexible than existing ones; the algorithm may be seen as a local descent algorithm over ordered partitions with explicit convergence guarantees. For functions that decompose into the sum of two functions F_1 and F_2 with efficient SFM oracles, we propose a new active-set algorithm for total variation denoising (and hence for SFM by thresholding the solution at zero). This algorithm also optimizes over ordered partitions and improves over existing ones based on TV or SFM oracles for F_1 and F_2.

1 Introduction

Submodular optimization problems such as total variation denoising and submodular function minimization are convex optimization problems which are common in machine learning, signal processing and computer vision [1], with notably application to graph cut-based image segmentation [9], sensor placement [18], or document summarization [20].

In this paper, we consider a submodular function F defined on $V = \{1, \dots, n\}$ as well as a vector $u \in \mathbb{R}^n$. We aim at minimizing with respect to $w \in \mathbb{R}^n$:

$$f(w) - u^\top w + \tfrac{1}{2}\|w\|_2^2, \tag{1}$$

where f is the Lovász extension of f. If F is a cut function in a weighted undirected graph, then f is the total variation, hence the denomination of total variation denoising problem which we use in this paper—since it is equivalent to minimizing $\frac{1}{2}\|u - w\|_2^2 + f(w)$.

We also consider the submodular function minimization (SFM) problem:

$$\min_{w \in [0,1]^n} f(w) - u^\top w = \min_{A \subseteq V} F(A) - u(A), \tag{2}$$

where we use the convention $u(A) = u^\top 1_A$, where $1_A \in \mathbb{R}^n$ is the indicator vector of the set A. Our goal in this paper is to propose iterative algorithms to solve these two problems given certain oracles on the submodular function F.

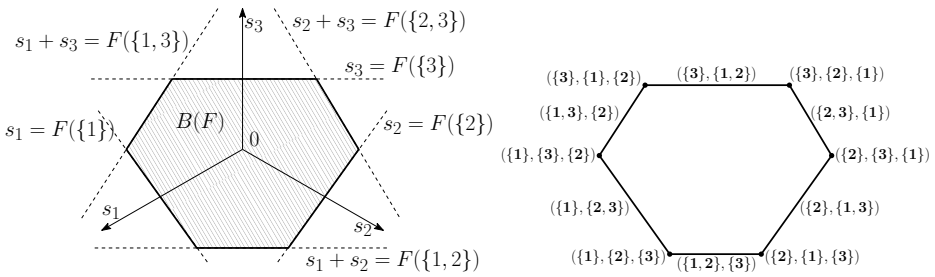


Figure 1: Base polytope for $n=3$. Left: definition from its supporting hyperplanes $\{s(A) = F(A)\}$. Right: each face (point or segment) of $B(F)$ is associated with an ordered partition.

Relationship with existing work. Generic algorithms which only access F through function values (e.g., subgradient descent or min-norm-point algorithm) are too slow without any assumptions [1], as for signal processing applications, high precision is typically required (and often the exact solution).

Generic algorithms have reasonable running-time for decomposable problems, i.e., when $F = F_1 + \cdots + F_r$, where each F_j is "simple" and called with more powerful oracles. When only SFM oracles are used for each function F_j [24], they remain significantly slower than existing algorithms. However, when total variation oracles for each F_j (which are significantly more expensive than SFM oracles) are used, they become competitive [17, 19, 16].

In this paper, we exploit the polytope structure of these non-smooth optimization problems, where each face is indexed by a partition of the underlying set $V = \{1, \dots, n\}$. The main insight of this paper is that once given a face of the base polytope $B(F)$ and its tangent cone, orthogonal projections may be done in linear time by isotonic regressions. We will only need SFM oracles, i.e., the minimization of $F(A) - s(A)$ with respect to $A \subseteq V$ for all possible $s \in \mathbb{R}^n$, to check optimality of this partition and/or generate a new partition.

Contributions. We make two main contributions:

- Given a submodular function F with an SFM oracle, we propose a new active-set algorithm for total variation denoising, which is more efficient and flexible than existing ones (i.e., it allows warm restarts). This algorithm may be seen as a local descent algorithm over ordered partitions.
- Given a decomposition of $F = F_1 + \cdots + F_r$, with available SFM oracles for each F_j, we propose a new active-set algorithm for total variation denoising for F (and hence for SFM by thresholding the solution at zero). This is also an algorithm that optimizes over ordered partitions (one per function F_j). Following [16, 19], this algorithm is naturally parallelizable. Given that only SFM oracles are needed, it is much more flexible.

2 Review of Submodular Analysis

A set-function $F : 2^V \to \mathbb{R}$ is said submodular if and only if $F(A) + F(B) \geqslant F(A \cup B) + F(A \cap B)$ for any subsets A, B of V. Our main motivating examples in this paper are cuts in a weighted undirected graph with weight function $a : V \times V \to \mathbb{R}_+$, which will be our running example. We now review the relevant concepts from submodular analysis (for more details, see [1, 13]).

Lovász extension and convexity. The power set 2^V is naturally identified with the vertices $\{0,1\}^n$ of the hypercube in n dimensions (going from $A \subseteq V$ to $1_A \in \{0,1\}^n$). Thus, any set-function may be seen

as a function f on $\{0,1\}^n$. It turns out that f may be extended to the full hypercube $[0,1]^n$ by piecewise-linear interpolation, and then to the whole vector space $\mathbb{R}^n$. Given a vector $w \in \mathbb{R}^n$, and given its level-set representation as $w = \sum_{i=1}^m v_i 1_{A_i}$, with $(A_1, \dots, A_m)$ a partition of V and $v_1 > \cdots > v_m$, $f(w)$ is equal to $f(w) = \sum_{i=1}^m v_i \big[F(B_i) - F(B_{i-1})\big]$. For cut functions, the Lovász extension is the *total variation* equal to $f(w) = \sum_{i \in V, j \in V} a(i,j)|w_i - w_j|$, hence our denomination total variation denoising for the problem in Eq. (1).

The extension is piecewise linear for any set-function F. It turns out that it is convex if and only if F is submodular [21]. Any piecewise linear convex function may be represented as the support function of a certain polytope K, i.e., as $f(w) = \max_{s \in K} w^\top s$ [22]. For the Lovász extension of a submodular function, we have an explicit description, which we now review.

Base polytope. We define the *base polytope* as

$$B(F) = \big\{ s \in \mathbb{R}^n, \ s(V) = F(V), \ \forall A \subset V, s(A) \leqslant F(A) \big\}.$$

Given that it is included in the affine hyperplane $\{s(V) = F(V)\}$, it is traditionally represented projected on that hyperplane (see Figure 1, left). A key result in submodular analysis is that the Lovász extension is the support function of $B(F)$, that is, for any $w \in \mathbb{R}^n$, $f(w) = \sup_{s \in B(F)} w^\top s$; maximizers may be computed in closed form from an ordered level-set representation of w.

SFM as a convex optimization problem. Another key result of submodular analysis is that minimizing a submodular function F (i.e., minimizing the Lovász extension f on $\{0,1\}^n$), is equivalent to minimizing the Lovász extension f on the full hypercube $[0,1]^n$ (a convex optimization problem).

Total variation denoising as projection onto the base polytope. A consequence of the representation of f as a support function leads to the following primal/dual pair [1, Sec. 8]:

$$\min_{w \in \mathbb{R}^n} f(w) - u^\top w + \tfrac{1}{2}\|w\|_2^2 = \max_{s \in B(F)} -\tfrac{1}{2}\|s - u\|_2^2, \tag{3}$$

with $w = u - s$ at optimality. Thus the TV problem is equivalent to the orthogonal projection of u onto $B(F)$.

From TV denoising to SFM. The SFM problem in Eq. (2) and the TV problem in Eq. (1) are tightly connected. Indeed, given the unique solution w of the TV problem, then we obtain a solution of $\min_{A \subseteq V} F(A) - u(A)$ by thresholding w at 0, i.e., by taking $A = \{i \in V, w_i \geqslant 0\}$.

Conversely, one may solve the TV problem by an appropriate sequence of SFM problems. The original divide-and-conquer algorithm may involve $O(n)$ SFM problems [15]. The extended algorithm of [16] can reach a precision ε in $O(\log \frac{1}{\varepsilon})$ but can only get the exact solution in $O(n)$ oracles.

3 Ordered Partitions and Isotonic Regression

The main insight of this paper is (a) to consider the detailed face structure of the base polytope $B(F)$ and (b) to notice that for the outer approximation of $B(F)$ based on the tangent cone to a certain face, the orthogonal projection problem (which is equivalent to constrained TV denoising) may be solved efficiently in $O(n)$ by a simple isotonic regression problem. This allows an explicit efficient local search over ordered partitions.

3.1 Faces, ordered partitions, and outer approximations of $B(F)$

Supporting hyperplanes. The base polytope is defined as the intersection of half-spaces $\{s(A) \leqslant F(A)\}$, for $A \subseteq V$. Therefore, faces of $B(F)$ are indexed by subsets of the power set. As a consequence of

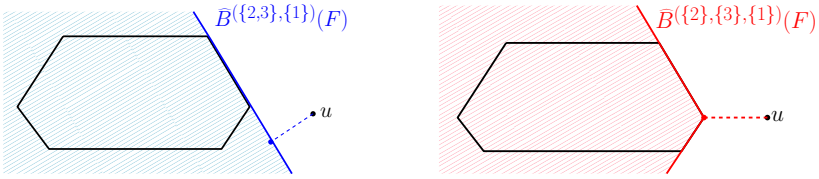


Figure 2: Projection algorithm for a single polytope: first projecting on the outer approximation $\widehat{B}^{(\{2,3\},\{1\})}(F)$, with a projected element which is not in $B(F)$ (blue), then on $\widehat{B}^{(\{2\},\{3\},\{1\})}(F)$, with a projected element being the projection of s onto $B(F)$ (red).

submodularity [1, 13], the faces of the base polytope $B(F)$ are characterized by "ordered partitions" $\mathcal{A} = (A_1, \dots, A_m)$ with $V = A_1 \cup \cdots \cup A_m$. Then, a face of $B(F)$ is such that $s(B_i) = F(B_i)$ for all $B_i = A_1 \cup \cdots \cup A_i$, $i = 1, \dots, m$. See the right plot of Figure 1 for the enumeration of faces for $n = 3$ based on an enumeration of all ordered partitions.

From a face of $B(F)$ defined by the ordered partition $\mathcal{A}$, we may define its tangent cone $\widehat{B}^{\mathcal{A}}(F)$ at this face as the set

$$\widehat{B}^{\mathcal{A}}(F) = \big\{ s \in \mathbb{R}^n, \ s(V) = F(V), \ \forall i \in \{1, \dots, m-1\}, \ s(B_i) \leqslant F(B_i) \big\}.$$

These are outer approximations of $B(F)$, as illustrated in Figure 2 for two ordered partitions.

Support function. We may compute the support function of $\widehat{B}^{\mathcal{A}}(F)$, which should be an upper bound on $f(w)$ since this set is an outer approximation of $B(F)$. As shown in the appendix, it is equal to the Lovász extension $f(w)$ when w is *compatible* with $\mathcal{A}$, that is, for w having ordered level sets corresponding to the ordered partition $\mathcal{A}$, and $+\infty$ otherwise.

3.2 Isotonic regression for restricted problems

Given an ordered partition $\mathcal{A} = (A_1, \dots, A_m)$ of V, we consider the original TV problem restricted to w being compatible with $\mathcal{A}$. Since on this constraint set $f(w) = \sum_{i=1}^m v_i \big[F(B_i) - F(B_{i-1}) \big]$, this is equivalent to

$$\min_{v \in \mathbb{R}^m} \sum_{i=1}^m v_i \big[F(B_i) - F(B_{i-1}) - u(A_i) \big] + \tfrac{1}{2} \sum_{i=1}^m |A_i| v_i^2 \text{ such that } v_1 \geqslant \cdots \geqslant v_m. \qquad (4)$$

This may be done by isotonic regression in complexity $O(m)$ by the weighted pool-adjacent-violator algorithm [7]. Typically the solution v will have some values which are equal to each other, which corresponds to merging some sets A_i. If these merges are made, we now obtained an ordered partition such that our optimal w has strictly decreasing values. These values are equal to $v_i = u(A_i)/|A_i| - (F(B_i) - F(B_{i-1}))/|A_i|$, i.e., given $\mathcal{A}$, the exact solution of the TV problem may be obtained in closed form.

Basic ordered partition. Given a submodular function F and an ordered partition $\mathcal{A}$, when the unique solution problem in Eq. (6) is such that $v_1 > \cdots > v_m$, we say that we $\mathcal{A}$ is a *basic ordered partition* for $F - u$. Given any ordered partition, isotonic regression allows to compute a coarser partition (obtained by merging some sets) which is a basic.

Dual interpretation. The dual of the problem in Eq. (6) is, with the relationship $w = u - s$, $\max_{s \in \widehat{B}^{\mathcal{A}}(F)} -\frac{1}{2}\|s - u\|_2^2$. Thus, this corresponds to projecting u on the outer approximation $\widehat{B}^{\mathcal{A}}(F)$ of $B(F)$ which only keeps m constraints instead of the $2^n - 1$ constraints defining $B(F)$. See an illustration in Figure 2.

3.3 Checking optimality of a basic ordered partition

Given a basic ordered partition $\mathcal{A}$, the associated $w \in \mathbb{R}^n$ is optimal for the TV problem in Eq. (1) if and only if $s = u - w \in B(F)$, which can be checked by minimizing the submodular function $F - s$. For a basic partition, a more efficient algorithm is available.

By repeated application of submodularity, we have for all sets $C \subseteq V$, if $C_i = C \cap A_i$:

$$F(C) - s(C) \geqslant \sum_{i=1}^m \big[F(B_{i-1} \cup C_i) - F(B_{i-1}) - s(C_i)\big].$$

Since, moreover, we have $s(A_i) = F(B_i) - F(B_{i-1})$, which implies $s(B_i) = F(B_i)$ for all $i \in \{1, \dots, m\}$, all subproblems $\min_{C_i \subseteq A_i} F(B_{i-1} \cup C_i) - F(B_{i-1}) - s(C_i)$ have non-positive values. This implies that we may check optimality by solving these m subproblems: s is optimal if and only if all of them have zero values. This leads to smaller subproblems whose overall complexity is less than a single SFM oracle calls. Moreover, for cut functions, it may be solved by a single oracle call on a graph where some edges have been removed [25].

Given all sets C_i, we may then define a new ordered partition by splitting all A_i for which $F(B_{i-1} \cup C_i) - F(B_{i-1}) - s(C_i) < 0$. If no split is possible, the pair (w, s) is optimal for Eq. (1). Otherwise, this new strictly finer partition may not be basic, but as shown in the appendix, the value of the optimization problem in Eq. (6) is strictly lower (and leads to another basic ordered partition), which ensures finite convergence of the algorithm. This leads to the active set algorithm below. In appendix, we show how we can obtain certificates of optimality.

- **Input**: Submodular function F with SFM oracle, $u \in \mathbb{R}^n$, ordered partition $\mathcal{A}$
- **Algorithm**: iterate until convergence
  (a) Solve Eq. (6) by isotonic regression.
  (b) Merge the sets with equal values of v_i to define a new ordered partition $\mathcal{A}$. Define $w = \sum_{i=1}^m v_i 1_{A_i}$ and $s = u - w$.
  (c) Check optimality by solving $\min_{C_i \subseteq A_i} F(B_{i-1} \cup C_i) - F(B_{i-1}) - s(C_i)$ for $i \in \{1, \dots, m\}$.
  (d) If s not optimal, for all C_i which are different from $\varnothing$ and A_i, add the new set $B_{i-1} \cup C_i$ in the ordered partition $\mathcal{A}$.
- **Output**: $w \in \mathbb{R}^n$ and $s \in B(F)$.

Relationship with divide-and-conquer algorithm. When starting from the trivial ordered partition $\mathcal{A} = (V)$, then we exactly obtain a parallel version of the divide-and-conquer algorithm [15], that is, the isotonic regression problem in (a) is always solved without using the constraints of monotonicity, i.e., there are no merges in (b), and thus in (c), it is not necessary to re-solve the problems where nothing has changed. This shows that the number of iterations is then less than n. The key added benefits in our formulation is the possibility of warm-starting, which can be very useful for building paths of solutions with different weights on the total variation. See experiments in Section 5.

4 Decomposable Problems

Many interesting problems in signal processing and computer vision naturally involve submodular functions F that decompose into $F = F_1 + \cdots + F_r$, with r "simple" submodular functions. For example, a cut function in a 2D grid decomposes into a function F_1 composed of cuts along vertical lines and a function F_2 composed of cuts along horizontal lines. For both of these functions, SFM oracles may be solved in $O(n)$ by message passing. For simplicity, in this paper, we consider the case $r = 2$ functions, but following [17, 16], our framework easily extends to $r > 2$.

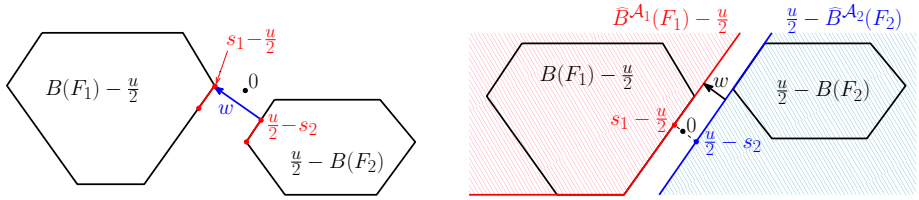


Figure 3: Closest point between two polytopes. Left: output of Dykstra's algorithm for the TV problem, the pair (s_1, s_2) may not be unique while $w = s_1 + s_2 - u$ is. Right: Dykstra's output for outer approximations.

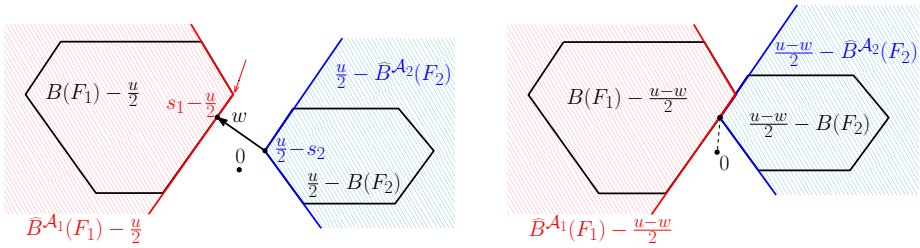


Figure 4: Translated intersecting polytopes. Left: output of Dykstra's algorithm before translation. Right: Translated formulation.

4.1 Reformulation as finding the distance between two polytopes

Following [16], we have the primal/dual problems (see proof in appendix):

$$\min_{w \in \mathbb{R}^n} f_1(w) + f_2(w) - u^\top w + \tfrac{1}{2}\|w\|_2^2 = \max_{s_1 \in B(F_1),\ s_2 \in B(F_2)} -\tfrac{1}{2}\|s_1 + s_2 - u\|_2^2, \tag{5}$$

with $w = u - s_1 - s_2$ at optimality. This is the projection of u on the sum of the base polytopes $B(F_1) + B(F_2) = B(F)$. This may be interpreted as finding the distance between two polytopes $B(F_1) - u/2$ and $u/2 - B(F_2)$. Note that these two polytopes typically do not intersect (they will if and only if $w = 0$ is the optimal solution of the TV problem, which is an uninteresting situation).

Assuming that TV oracles are available for F_1 and F_2, [16, 19] consider alternating projection [4] and alternating reflection [5] algorithms. However, none of these algorithms can be cast explicitly as descent algorithms for the primal TV problem (alternating projections is equivalent to block *dual* coordinate ascent), which we require for our local search over partitions. Dykstra's algorithm [3] can also be used and has a form of primal descent interpretation, i.e., as coordinate descent for a well-formulated primal problem [14]. We have implemented it and it behaves similarly to alternating projections, but it still requires TV oracles (see experiments in Section 5). There is however a key difference: while alternating projections and alternating reflections always converge to a pair of closest points, Dykstra's algorithm converges to a *specific* pair of points, namely the pair closest to the initialization of the algorithm [3]; see an illustration in Figure 3 (left). This insight will be key in our algorithm to avoid cycling.

4.2 First attempt at an active-set method

Given our algorithm for a single function, it is natural to perform a local search over two partitions $\mathcal{A}_1$ and $\mathcal{A}_2$, one for each function F_1 and F_2, and consider in the primal formulation a weight vector w compatible with both $\mathcal{A}_1$ and $\mathcal{A}_2$; or, equivalently, in the dual formulation, two outer approximations $\widehat{B}^{\mathcal{A}_1}(F_1)$ and $\widehat{B}^{\mathcal{A}_2}(F_2)$.

That is, given the ordered partitions $\mathcal{A}_1$ and $\mathcal{A}_2$, we consider the primal/dual pairs of optimization problems

$$\max_{s_1 \in \widehat{B}^{\mathcal{A}_1}(F_1),\ s_2 \in \widehat{B}^{\mathcal{A}_2}(F_2)} -\tfrac{1}{2}\|u - s_1 - s_2\|_2^2 = \min_{w \in \mathbb{R}^n} f_1(w) + f_2(w) - u^\top w + \tfrac{1}{2}\|w\|_2^2,$$

such that w is compatible with $\mathcal{A}_1$ and $\mathcal{A}_2$, with the relationship $w = u - s_1 - s_2$ at optimality.

Primal solution by isotonic regression. The primal solution w is unique by strong convexity. Moreover, it has to be compatible with both $\mathcal{A}_1$ and $\mathcal{A}_2$, which is equivalent to being compatible with the *coalesced* ordered partition $\mathcal{A} = \text{coalesce}(\mathcal{A}_1, \mathcal{A}_2)$ defined as the coarsest ordered partition compatible by both. As shown in the appendix, $\mathcal{A}$ may be found in time $O(\min(m_1, m_2)n)$.

Given $\mathcal{A}$, the primal solution w of the subproblem may be found by isotonic regression like in Section 3.2 in time $O(m)$ where m is the number of sets in $\mathcal{A}$. Finding optimal dual variables s_1 and s_2 turns out to be more problematic. We know that $s_1 + s_2 = u - w$ and that $s_1 + s_2 \in \widehat{B}^{\mathcal{A}}(F)$, but the split in (s_1, s_2) is unknown.

Obtaining dual solutions by convex feasibility algorithms. In order to obtain (s_1, s_2), we could use any convex feasibility algorithm such as alternating projections [4] or alternating reflections [5]. However, the result would depend in non understood ways on the initialization, and we have observed cycling of the active-set algorithm. Using Dykstra's algorithm allows us to converge to a unique well-defined pair (s_1, s_2) that will lead to a provably non-cycling algorithm.

4.3 Dykstra's algorithm for outer approximations

When running Dykstra's algorithm starting from 0 on the polytopes $\widehat{B}^{\mathcal{A}_1}(F_1) - u/2$ and $u/2 - \widehat{B}^{\mathcal{A}_2}(F_2)$, if w is the unique distance vector between the two polytopes, then the iterates converge to the projection of 0 onto the convex sets of elements in the two polytopes that achieve the minimum distance [3]. See Figure 3 (right) for an illustration. This algorithm is however slow to converge when the polytopes do not intersect (they will not here for the most interesting situations when $w \neq 0$) and convergence is hard to monitor because primal iterates diverge [3].

Translated intersecting polytopes. In our situation, we want to reach the Dykstra solution *while knowing the vector* w (as mentioned earlier, it is obtained cheaply from isotonic regression). Indeed, from Lemma 2.2 and Theorem 3.8 from [3], given this vector w, we may translate the two polytopes and now obtain a formulation where the two polytopes do intersect; that is we aim at projecting 0 on the (non-empty) intersection of $\widehat{B}^{\mathcal{A}_1}(F_1) - u/2 + w/2$ and $u/2 - w/2 - \widehat{B}^{\mathcal{A}_2}(F_2)$. See Figure 4. The key added benefit is that the two sets now intersect and Dykstra's algorithm is then linearly convergent [23].

Dykstra's iteration is as follows, and may be warm-started by the value of the auxiliary variable w_2:

$$\begin{aligned} s_{1,t} &= \Pi_{\widehat{B}^{\mathcal{A}_1}(F_1)}(u/2 - w/2 + w_{2,t-1}), \quad w_{1,t} = u/2 - w/2 + w_{2,t-1} - s_{1,t}, \\ s_{2,t} &= \Pi_{\widehat{B}^{\mathcal{A}_2}(F_2)}(u/2 - w/2 + w_{1,t}), \qquad w_{2,t} = u/2 - w/2 + w_{1,t} - s_{2,t}, \end{aligned}$$

with Π_C denoting the orthogonal projection onto the sets C, solved here by isotonic regression.

In our simulations, we have used the recent accelerated version of [12], which led to faster convergence. In order to monitor convergence, we compute the value of $\|u - w - s_{1,t} - s_{2,t}\|_1$ which is equal to zero at convergence. The optimization problem can also be decoupled into smaller optimization problems by using the knowledge of the face of the base polytopes on which s_1 and s_2 lie. See details in the appendix.

4.4 Algorithm

Our active-set algorithm is presented below.

– **Input**: Submodular function F_1 and F_2 with SFM oracles, $u \in \mathbb{R}^n$, ordered partitions $\mathcal{A}_1, \mathcal{A}_2$

– **Algorithm**: iterate until convergence (i.e., $\varepsilon_1 + \varepsilon_2$ small enough)

(a) Find $\mathcal{A} = \text{coalesce}(\mathcal{A}_1, \mathcal{A}_2)$ and run isotonic regression to minimize $f(w) - u^\top w + \frac{1}{2}\|w\|_2^2$ such that w is compatible with $\mathcal{A}$.

(b) Run accelerated Dykstra's algorithm ro find the projection of 0 onto the intersection of $\widehat{B}^{\mathcal{A}_1}(F_1) - u/2 + w/2$ and $u/2 - w/2 - \widehat{B}^{\mathcal{A}_2}(F_2)$.

(c) Merge the sets in $\mathcal{A}_j$ which are tight for s_j, $j \in \{1, 2\}$.

(c) Check optimality by solving $\min_{C_{j,i_j} \subseteq A_{j,i_j}} F_j(B_{j,i_j-1} \cup C_{j,i_j}) - F_j(B_{j,i+j-1}) - s_j(C_{j,i_j})$ for $i_j \in \{1, \dots, m_j\}$, Monitor ε_1 and ε_2 such that $F_j(C_j) - s_j(C_j) \geqslant -\varepsilon_j$, $j = 1, 2$.

(d) If both s_1 and s_2 not optimal, for all C_{j,i_j} which are different from $\varnothing$ and A_{j,i_j}, split partitions.

– **Output**: $w \in \mathbb{R}^n$ and $s_1 \in B(F_1)$, $s_2 \in B(F_2)$.

Given two ordered partitions $\mathcal{A}_1$ and $\mathcal{A}_2$, we obtain $s_1 \in \hat{B}^{\mathcal{A}_1}(F_1)$ and $s_2 \in \hat{B}^{\mathcal{A}_2}(F_2)$ as described in Section 4.3. The solution $w = u - s_1 - s_2$ is optimal if and only if both $s_1 \in B(F_1)$ and $s_2 \in B(F_2)$. When running the optimality described in Section 3.3, we split the partition. As shown in appendix, either (a) $\|w\|_2^2$ strictly increases at each iteration, or (b) $\|w\|_2^2$ remains constant but $\|s_1 - s_2\|_2^2$ strictly increases (this is true only for the Dykstra solution). This implies that the algorithm is finitely convergent.

5 Experiments

5.1 Non decomposable total variation denoising

Our experiments consider images, which are 2-dimensional grids with 4-neighborhood. The dataset comprises of 6 different images of varying sizes. In this section, we restrict to anistropic uniform-weighted total variation to compare with Chambolle et al. [11]. Therefore, the total variation is $f(w) = \sum_{i \sim j} \lambda |w_i - w_j|$, where λ is a regularizing constant for solving the total variation problem in Eq. (1). Note that we restrict to uniform weights only to be able to perform a fair comparison with their method [11].

Maxflow [8] is used as the SFM oracle for checking the optimality of the ordered partitions. Figure 5(a) shows the number of oracle calls to solve the TV denoising problem with increasing numbers of pixels in the image. Figure 5(b) shows the time required for each of the methods to solve the TV problem to convergence. We have an optimized code and only use the oracle as plugin which takes about 80-85 percent of the running time. This is primarily the reason our approach takes more time than [11] inspite of having lesser oracle calls. Note that the horizontal axis of the plots are not scaled linearly with the numbers of pixels. We always attain an optimal solution when compared to [11], which always obtains a slightly sub-optimal solution.

Figure 5(c) also shows the ability to warm start by using the output of a related problem, i.e., when computing the solution for several values of λ (which is typical in practice). In this case, we use optimal ordered partitions of the problem with larger λ to warmstart the problem with smaller λ. It can be observed that warm start of the algorithm requires lesser number of oracle calls to converge than using trivial ordered partition, as initialized. Warm start also largely helps in reducing the burden on the SFM oracle. With warm starts the number of ordered partitions does not change much over iterations. Hence, it suffices to query only ordered partitions that have changed. To analyze this we define *oracle complexity* as the ratio of pixels in the elements of the partitions that need to be queried with the full set. Oracle complexity is averaged over iterations to understand the average burden on the oracle per iteration. With warm starts this reduces drastically, which can be observed in Figure 5(d).

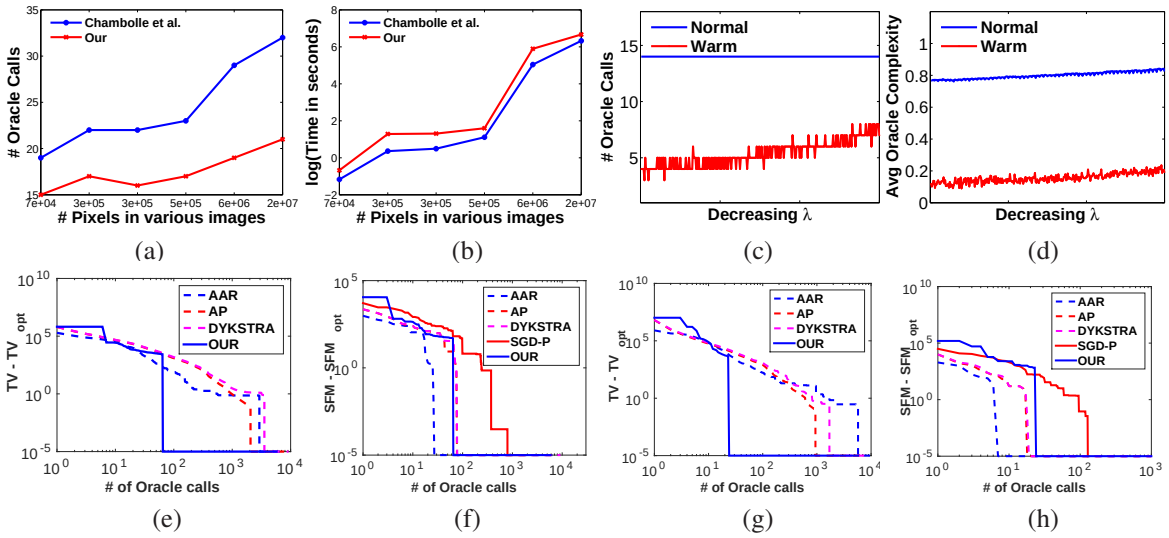


Figure 5: (a) 2D SFM calls for images of various sizes, (b) Time taken for images of various sizes, (c) Number of iterations with and without warm start, (d) Average complexity of the oracle with and without warm start, (e) TV convergence for the 640×427 image, (f) SFM convergence for the 640×427 image, (g) TV for 300×427 image, (g) SFM for 300×427 image.

5.2 Decomposable total variation denoising and SFM

In the decomposable case, we consider a 2D-grid that decomposes into a function F_1 composed of cuts along vertical lines and a function F_2 composed of cuts along horizontal lines. For each of these functions, the corresponding SFM oracle is a message passing algorithm that can be solved in $O(n)$ time. We compare our algorithm with other methods like alternating projection (AP), averaged alternating reflection (AAR) [5], Dykstra's alternating projection [3] and dual subgradient based method (SGD-P) [17] modified with Polyak's [6] rule. SGD-P is the only other algorithm, which requires 1D *SFM* oracle, while the others require a 1D *TV* oracle [2].

In our experiments, we consider two images of sizes 300×427 and 640×427. We consider the segmentation problem solved using maxflow algorithms following the set up of [16]. The unary potentials of each pixel is calculated using the Gaussian mixture model of the color features. The edge weight $a(i,j) = \exp(-\|y_i - y_j\|^2)$, where y_i denotes the RGB values of the pixel i. Figure 5(e) and (g) shows that our algorithm converges for solving TV quickly by using only simple oracles and relatively less number of oracle calls. However, for solving the SFM problem, we use similar number of calls as methods using 1D *TV* oracles but much lesser than SGD-P, which also uses 1D *SFM* oracle.

The oracle time for solving TV for SGD-P on 640×427 image is about 3.53 seconds while we only spend about 1.74 seconds. For the same image AAR, which uses *1D TV oracle* consumes around 3.35 second because it uses a complex oracle. However, our implementation spends much time in solving step-(b) of the Algorithm in Section 4.4, which we refer to as the Dykstra step. Dykstra's algorithm is linearly convergent but not finitely convergent. Therefore, it suffices to solve this approximately. We thus introduce a parameter α to approximately solve the Dykstra step such that $\|s_1 + s_2 - u + w\|_1 \leq \alpha(\epsilon_1 + \epsilon_2)$. We observed a trade-off between approximation of the Dykstra step and number of oracle calls. See appendix for more details.

6 Conclusion

We have presented an active-set method to solve submodular function minization (SFM) and total variation (TV) denoising problem. For decomposable problems, we have showed that we can solve both TV denoising and SFM problems by using cheaper SFM oracles of the individual functions, while other competitive methods use expensive TV oracles, which restricts the function decomposition, as for cut functions, they are efficient only for chains and trees. This provides us flexibility to decompose into functions for which we have efficient SFM oracles. In this paper, we consider only decomposition into 2 functions but this can be generalized to r functions following the formulations in [16, 19]. Due to inherent parallelism, this approach can be very useful in solving large scale optimization problems. As future work, it could be interesting to improve the Dykstra step in the decomposable case and extend this formulation to solve constrained submodular optimization.

A Support function for outer approximation

In this section, we compute the support function of the outer approximation $\widehat{B}^{\mathcal{A}}(F)$. We have, by Lagrangian duality:

$$\begin{aligned}
& \sup_{s \in \widehat{B}^{\mathcal{A}}(F)} w^\top s \\
= & \sup_{s \in \mathbb{R}^n} \inf_{\lambda \in \mathbb{R}_+^{m-1} \times \mathbb{R}} w^\top s - \sum_{i=1}^m \lambda_i (s(B_i) - F(B_i)) \\
= & \inf_{\lambda \in \mathbb{R}_+^{m-1} \times \mathbb{R}} \sup_{s \in \mathbb{R}^n} s^\top \Big(w - \sum_{i=1}^m (\lambda_i + \cdots + \lambda_m) 1_{A_i} \Big) + \sum_{i=1}^m (\lambda_i + \cdots + \lambda_m) \big[F(B_i) - F(B_{i-1}) \big] \\
= & \inf_{\lambda \in \mathbb{R}_+^{m-1} \times \mathbb{R}} \sum_{i=1}^m (\lambda_i + \cdots + \lambda_m) \big[F(B_i) - F(B_{i-1}) \big] \text{ such that } w = \sum_{i=1}^m (\lambda_i + \cdots + \lambda_m) 1_{A_i}.
\end{aligned}$$

Thus, by defining $v_i = \lambda_i + \cdots + \lambda_m$, then the support function is finite for w having ordered level sets corresponding to the ordered partition $\mathcal{A}$ (we then say that w is *compatible* with $\mathcal{A}$); it is then equal to the Lovász extension w. Otherwise, when w is not compatible with $\mathcal{A}$, the support function is infinite.

B Proof of convergence for Algorithm proposed in Section - 3

In order to prove the convergence of algorithm, we only need to show that if the optimality check fails, then we obtain a partition on which the isotonic problem

$$\min_{v \in \mathbb{R}^m} \sum_{i=1}^m v_i \big[F(B_i) - F(B_{i-1}) - u(A_i) \big] + \frac{1}{2} \sum_{i=1}^m |A_i| v_i^2 \text{ such that } v_1 \geqslant \cdots \geqslant v_m. \tag{6}$$

has a strictly lower value. Before checking optimality, the merging in step (b) ensures that the sets A_i are exactly the constant sets of w (i.e., $\mathcal{A}$ is a basic partition). Around w, then the components of w_{A_i} are strictly greater than the one of $w_{A_{i-1}}$. This implies that the Lovász extension decouples as

$$f(w) = \sum_{i=1}^m f_{A_i | B_{i-1}}(w_{A_i}),$$

where $F_{A_i|B_{i-1}}(C_i) = F(B_{i-1} \cup C_i) - F(B_{i-1})$ for $C_i \subset A_i$. Thus the optimality check indeed decouples as independent checks on the set A_i (as shown in the main paper directly). If for a index $i \in \{1, \dots, m\}$, there is C_i such that $F(B_{i-1} \cup C_i) - F(B_{i-1}) - s(C_i) < 0$, then by taking a direction Δ_{A_i} with values 1 in C_i and -1 otherwise, for $a_i, b_i > 0$, and $t > 0$ small enough,

$$\begin{aligned}
& f_{A_i|B_{i-1}}(w_{A_i} + t\Delta_{A_i}) - u^\top(w_{A_i} + t\Delta_{A_i}) + w^\top \\
= \; & f_{C_i|B_{i-1}}\big((v_i + t)1_{C_i}\big) + f_{A_i \setminus C_i|B_{i-1} \cup C_i}\big((v_i - t)1_{A_i \setminus C_i}\big) \\
= \; & (v_i + t)\big(F(C_i \cup B_{i-1}) - F(B_{i-1})\big) + (v_i - t\big(F(B_i) - F(C_i \cup B_{i-1})\big) \\
= \; & v_i\big[F(B_i) - F(B_{i-1})\big] + t\Big[2F(C_i \cup B_{i-1}) - F(B_{i-1}) - F(B_i)\Big] \\
= \; & f_{A_i|B_{i-1}}(w_{A_i}) + t\Big[2F(C_i \cup B_{i-1}) - F(B_{i-1}) - F(B_i)\Big].
\end{aligned}$$

If we add $u^\top(w_{A_i} + t\Delta_{A_i}) + \frac{1}{2}\|w_{A_i} + t\Delta_{A_i}\|_2^2 - u^\top w_{A_i} + \frac{1}{2}\|w_{A_i}\|_2^2$, the directional derivative in direction Δ_{A_i} of the cost function minimized for the TV problem restricted to A_i is equal to

$$\begin{aligned}
& 2F(C_i \cup B_{i-1}) - F(B_{i-1}) - F(B_i) - u(C_i) + u(A_i \setminus C_i) + w_{A_i}^\top \Delta_{A_i} \\
= \; & 2F(C_i \cup B_{i-1}) - F(B_{i-1}) - F(B_i) - u(C_i) + u(A_i \setminus C_i) + v_i|A_i| - v_i|A_i \setminus C_i| \\
= \; & 2F(C_i \cup B_{i-1}) - F(B_{i-1}) - u(C_i) + v_i|A_i| - F(B_i) + u(A_i) - u(C_i) - v_i|A_i| + v_i|C_i|
\end{aligned}$$

We have $v_i|A_i| = u(A_i) - F(B_i) + F(B_{i-1})$ and $s(C_i) = u(C_i) - v_i|C_i| = u(C_i) - u(A_i) + \frac{|C_i|}{|A_i|}(F(B_i) - F(B_{i-1}))$, from optimality conditions for isotonic regression, which implies that the quantity above is equal to

$$2F(C_i \cup B_{i-1}) - 2F(B_{i-1}) - 2s(C_i) < 0.$$

Hence we have a direction of strict descent by splitting the two partitions.

C Certificates of optimality for Algorithm proposed in Section - 3

Certificates of optimality. The new algorithm has dual-infeasible iterates s (they only belong to $B(F)$ at convergence). However, after step (c), we have that for all $C \subset V$, $F(C) - s(C) \geqslant -\varepsilon$. This implies that $s \in B(F + \varepsilon 1_{\mathrm{Card} \in (0,n)})$, i.e., $s \in B(F_\varepsilon)$ with $F_\varepsilon = F + \varepsilon 1_{\mathrm{Card} \in (0,n)}$. Since by construction $w = u - s$, we have:

$$\begin{aligned}
f_\varepsilon(w) - u^\top w + \frac{1}{2}\|w\|_2^2 + \frac{1}{2}\|s - u\|_2^2 &= \varepsilon\big|\max_{j \in V} w_j - \min_{j \in V} w_j\big| + f(w) - u^\top w + \|w\|^2 \\
&= \varepsilon\big|\max_{j \in V} w_j - \min_{j \in V} w_j\big| \\
&\quad + \sum_{i=1}^m v_i\big[F(B_i) - F(B_{i-1}) - u(A_i)\big] + \sum_{i=1}^m |A_i| v_i^2 \\
&= \varepsilon\big|\max_{j \in V} w_j - \min_{j \in V} w_j\big| = \varepsilon \operatorname{range}(w),
\end{aligned}$$

where $\operatorname{range}(w) = \max_{k \in V} w_k - \min_{k \in V} w_k$. This means that w is approximately optimal for $f(w) - u^\top w + \frac{1}{2}\|w\|_2^2$ with *certified gap* less than $\varepsilon \operatorname{range}(w) + \varepsilon \operatorname{range}(w^*)$.

Maximal range of an active-set solution. For any ordered partition $\mathcal{A}$, and the optimal value of w (which we know in closed form), we have $\operatorname{range}(w) \leqslant \operatorname{range}(u) + \max_{i \in V}\big\{F(\{i\}) + F(V \setminus \{i\}) - F(V)\big\}$. Indeed, for the u part of the expression, this is because values of w are averages of values of u; for the F part of the expression, we always have $F(B_i) - F(B_{i-1}) \leqslant \sum_{k \in A_i} F(\{k\})$ and $F(B_i) - F(B_{i-1}) \geqslant -\sum_{k \in A_i} F(V) - F(V \setminus \{k\})$.

Exact solution. If we have an approximate solution of the TV problemwith gap ε, then (a) if the submodular function only takes integer values and (b) if $\varepsilon \leqslant (2n^2)^{-1}$, then we have the optimal solution [10].

D Dual of decomposable TV problem

Following [16], we have the primal/dual problems :

$$\begin{aligned}
& \min_{w \in \mathbb{R}^n} f_1(w) + f_2(w) - u^\top w + \frac{1}{2}\|w\|_2^2 \\
= & \min_{w \in \mathbb{R}^n} \max_{s_1 \in B(F_1),\ s_2 \in B(F_2)} w^\top (s_1 + s_2) - u^\top w + \frac{1}{2}\|w\|_2^2 \\
= & \max_{s_1 \in B(F_1),\ s_2 \in B(F_2)} \min_{w \in \mathbb{R}^n} (s_1 + s_2 - u)^\top w + \frac{1}{2}\|w\|_2^2 \\
= & \max_{s_1 \in B(F_1),\ s_2 \in B(F_2)} -\frac{1}{2}\|s_1 + s_2 - u\|_2^2,
\end{aligned}$$

with $w = u - s_1 - s_2$ at optimality.

E Algorithms for coalescing partitions

The basic interpretation in coalescing two ordered partitions is as follows. Given an ordered partition $\mathcal{A}_1$ and $\mathcal{A}_2$ with m_1 and m_2 elements in the partitions respectively, we define for each $j = 1, 2, \forall i_j = (1, \ldots, m_j)$,

$$B_{j,i_j} = (A_{j,1} \cup \ldots \cup A_{j,i_j}).$$

The inqualities defining the outer approximation of the base polytopes are given by hyperplanes defined by $\forall i_j = (1, \ldots, m_j), s_j(B_{j,i_j}) \leq F_j(B_{j,i_j})$. The hyperplanes defined by common sets of both these partitions, defines the coalesced ordered partitions. The following algorithm performs coalescing between these partitions.

- **Input**: Ordered partitions $\mathcal{A}_1$ and $\mathcal{A}_2$.
- **Initialize**: $x = 1$, $y = 1$, $z = 1$ and $C = \varnothing$.
- **Algorithm**: Iterate until $x = m_1$ and $y = m_2$ with $m = z$
  - (a) If $|B_{1,x}| > |B_{2,y}|$ then $y := y + 1$.
  - (b) If $|B_{1,x}| < |B_{2,y}|$ then $x := x + 1$.
  - (c) If $|B_{1,x}| == |B_{2,y}|$ then
    - If $B_{1,x} == B_{2,y}$ then
      - $A_z = (B_{1,x} \setminus C)$,
      - $C = B_{1,x}$, and
      - $z := z + 1$.
- **Output**: $m = z$, ordered partitions $\mathcal{A} = (A_1, \ldots, A_m)$.

F Optimality of algorithm for decomposable problems

In step (d) of the algorithms, when we split partitions, the value of the primal/dual pair of optimization algorithms

$$\begin{aligned} & \max_{s_1 \in \widehat{B}^{\mathcal{A}_1}(F_1),\ s_2 \in \widehat{B}^{\mathcal{A}_2}(F_2)} -\tfrac{1}{2}\|u - s_1 - s_2\|_2^2, \\ = & \min_{w \in \mathbb{R}^n} f_1(w) + f_2(w) - u^\top w + \tfrac{1}{2}\|w\|_2^2 \text{ such that } w \text{ compatible with } \mathcal{A}_1 \text{ and } \mathcal{A}_2, \end{aligned}$$

cannot increase. This because, when splitting, the constraint set for the minimization problem only gets bigger. Since at optimality, we have $w = u - s_1 - s_2$, $\|w\|_2$ cannot decrease, which shows the first statement.

Now, if $\|w\|_2$ remains constant after an iteration, then it has to be the same (and not only have the same norm), because the optimal s_1 and s_2 can only move in the direction orthogonal to w.

In step (b) of the algorithm, we project 0 on the (non-empty) intersection of $\widehat{B}^{\mathcal{A}_1}(F_1) - u/2 + w/2$ and $u/2 - w/2 - \widehat{B}^{\mathcal{A}_2}(F_2)$. This corresponds to minimizing $\frac{1}{2}\|s_1 - u/2 + w/2\|^2$ such that $s_1 \in \widehat{B}^{\mathcal{A}_1}(F_1)$ and $s_2 = u - w - s_1 \in \widehat{B}^{\mathcal{A}_2}(F_2)$. This is equivalent to minimizing $\frac{1}{8}\|s_1 - s_2\|^2$. We have:

$$\begin{aligned} & \max_{s_1 \in \widehat{B}^{\mathcal{A}_1}(F_1),\ s_2 \in \widehat{B}^{\mathcal{A}_2}(F_2)} -\frac{1}{8}\|s_1 - s_2\|_2^2 \text{ such that } s_1 + s_2 = u - w \\ = & \min_{w_1 \in \mathbb{R}^n,\ w_2 \in \mathbb{R}^n} \max_{s_1 \in \mathbb{R}^n,\ s_2 \in \mathbb{R}^n} -\frac{1}{8}\|s_1 - s_2\|_2^2 + f_1(w_1) + f_2(w_2) - w_1^\top s_1 - w_2^\top s_2 \text{ such that } s_1 + s_2 = u - w \\ = & \min_{w_1 \in \mathbb{R}^n,\ w_2 \in \mathbb{R}^n} \max_{s_2 \in \mathbb{R}^n} -\frac{1}{8}\|u - w - 2s_2\|_2^2 + f_1(w_1) + f_2(w_2) - w_1^\top (u - w - s_2) - w_2^\top s_2 \\ = & \min_{w_1 \in \mathbb{R}^n,\ w_2 \in \mathbb{R}^n} \max_{s_2 \in \mathbb{R}^n} -\frac{1}{8}\|u - w\|_2^2 - \frac{1}{2}\|s_2\|_2^2 + \frac{1}{2}s_2^\top (u - w) + f_1(w_1) + f_2(w_2) - w_1^\top (u - w - s_2) - w_2^\top s_2 \\ = & \min_{w_1 \in \mathbb{R}^n,\ w_2 \in \mathbb{R}^n} -w_1^\top (u - w) + f_1(w_1) + f_2(w_2) - \frac{1}{8}\|u - w\|_2^2 + \max_{s_2 \in \mathbb{R}^n} -\frac{1}{2}\|s_2\|_2^2 + s_2^\top (\frac{u}{2} - \frac{w}{2} + w_1 - w_2) \\ = & \min_{w_1 \in \mathbb{R}^n,\ w_2 \in \mathbb{R}^n} -w_1^\top (u - w) + f_1(w_1) + f_2(w_2) - \frac{1}{8}\|u - w\|_2^2 + \frac{1}{2}\|\frac{u}{2} - \frac{w}{2} + w_1 - w_2\|_2^2 \\ = & \min_{w_1 \in \mathbb{R}^n,\ w_2 \in \mathbb{R}^n} -w_1^\top (u - w) + f_1(w_1) + f_2(w_2) + \frac{1}{2}\|w_1 - w_2\|_2^2 + \frac{1}{2}(u - w)^\top (w_1 - w_2) \\ = & \min_{w_1 \in \mathbb{R}^n,\ w_2 \in \mathbb{R}^n} f_1(w_1) + f_2(w_2) - \frac{1}{2}(u - w)^\top (w_1 + w_2) + \frac{1}{2}\|w_1 - w_2\|_2^2, \end{aligned}$$

with the constraint that w_1 is compatible with $\mathcal{A}_1$ and w_2 is compatible with $\mathcal{A}_2$.

Thus s_1 and s_2 are dual to certain vectors w_1 and w_2, which minimize a decoupled formulation in f_1 and f_2. To check optimality, like in the single function case, it decouples over the constant sets of w_1 and w_2, which is exactly what step (c) is performing.

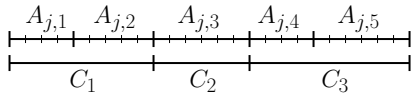
If the check is satisfied, it means that w_1 and w_2 are in fact optimal for the problem above without the restriction in compatibilies, which implies that they are the Dykstra solutions for the TV problem.

If the check is not satisfied, then the same reasoning as for the one function case, leads directions of descent for the new primal problem above. Hence it decreases; since its value is equal to $-\frac{1}{8}\|s_1 - s_2\|_2^2$, the value of $\|s_1 - s_2\|_2^2$ must increase, hence the second statement.

G Decoupled problems.

Given that we deal with polytopes, knowning w implies that we know the faces on which we have to looked for. It turns outs that for base polytopes, these faces are products of base polytopes for modified functions (a similar fact holds for their outer approximations).

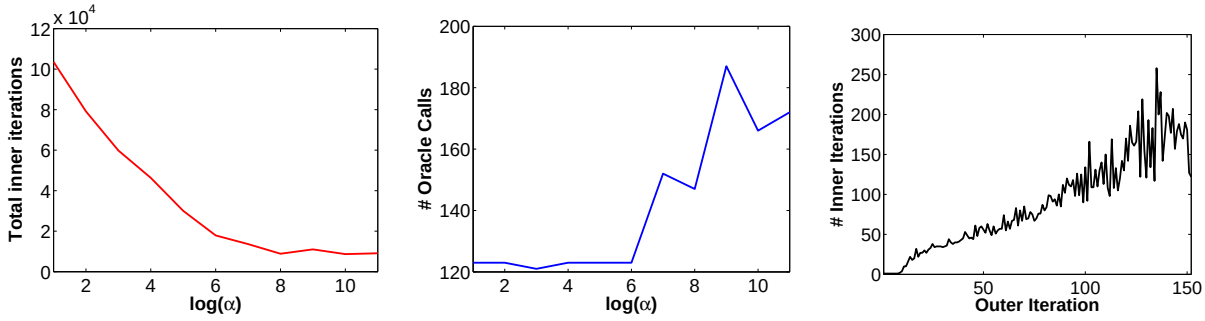
Given the ordered partition $\mathcal{A}'$ defined by the level sets of w (which have to be finer than $\mathcal{A}_1$ and $\mathcal{A}_2$), we know that we may restrict $\widehat{B}^{\mathcal{A}_j}(F_j)$ to elements s such that $s(B) = F(B)$ for all sup-level sets B of w (which have to be unions of contiguous elements of A_j); see an illustration below.



More precisely, if $C_1, \dots, C_{m'}$ are constant sets of w ordered with decreasing values. Then, we may search for s_j independently for each subvector $(s_j)_{C_k} \in \mathbb{R}^{C_k}$, $k \in \{1, \dots, m'\}$ and with the constraint that

$$(s_j)_{C_k} \in \hat{B}^{\mathcal{A}_j \cap C_k}\big[(F_j)_{C_k | C_1 \cup \cdots \cup C_{k-1}}\big],$$

where $\mathcal{A}_j \cap C_k$ is the ordered partition obtained from $\mathcal{A}_j$ once restricted onto C_k and the submodular function is the so-called contraction of F on C_k given $C_1 \cup \cdots \cup C_{k-1}$, defined as $S \mapsto F_j(S \cup C_1 \cup \cdots \cup C_{k-1}) - F(C_1 \cup \cdots \cup C_{k-1})$. Thus this corresponds to solving m different smaller subproblems.



(a) (b) (c)

Figure 6: (a) Total number of inner iterations for varying α. (b) Total number of outer iterations for varying α. and (c) Number of inner iterations per each outer iteration for the $\alpha = 10^1$

H Choice of α

The Dykstra step, i.e., step (b) of the algorithm proposed in Section - 4.4 is not finitely convergent. Therefore, it needs to be solved approxmately. For this purpose, we introduce a parameter α to approximately solve the Dysktra step such that $\|s_1 + s_2 - u + w\|_1 \leq \alpha(\epsilon_1 + \epsilon_2)$. Let ϵ be defined as $\alpha(\epsilon_1 + \epsilon_2)$. This shows that the s_1 and s_2 are ϵ-accurate. Therefore, α must be chosen in such a way that we avoid cycling in our algorithm. However, another alternative is to warm start the dykstra step with w_1 and w_2 of the previous iteration. This ensures we dont go back to the same w_1 and w_2, which we have already encountered and avoid cycling. Figure 6 shows the performance of our algorithm for a simple problem of 100×100 2D-grid with 4-neighborhood and uniform weights on the edges with varying α. Figure 6-(a) shows the total number of inner iterations required to solve the TV problem. Figure 6-(b) gives the total number of SFM orace calls required to solve the TV problem. In Figure 6-(c), we show the number of inner iterations in every outer iteration for the best α we have encountered.

References

[1] F. Bach. *Learning with Submodular Functions: A Convex Optimization Perspective*, volume 6 of *Foundations and Trends in Machine Learning*. NOW, 2013.

[2] A. Barbero and S. Sra. Modular proximal optimization for multidimensional total-variation regularization. Technical Report 1411.0589, ArXiv, 2014.

[3] H. H. Bauschke and J. M. Borwein. Dykstra's alternating projection algorithm for two sets. *Journal of Approximation Theory*, 79(3):418–443, 1994.

[4] H. H. Bauschke, J. M. Borwein, and A. S. Lewis. The method of cyclic projections for closed convex sets in Hilbert space. *Contemporary Mathematics*, 204:1–38, 1997.

[5] H. H. Bauschke, P. L. Combettes, and D. Luke. Finding best approximation pairs relative to two closed convex sets in Hilbert spaces. *J. Approx. Theo.*, 127(2):178–192, 2004.

[6] D. P. Bertsekas. *Nonlinear programming*. Athena Scientific, 1999.

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

[8] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE TPAMI*, 26(9):1124–1137, 2004.

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

[10] D. Chakrabarty, P. Jain, and P. Kothari. Provable submodular minimization using Wolfe's algorithm. In *Adv. NIPS*. 2014.

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

[12] A. Chambolle and T. Pock. A remark on accelerated block coordinate descent for computing the proximity operators of a sum of convex functions. Technical Report 01099182, HAL, 2015.

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

[14] N. Gaffke and R. Mathar. A cyclic projection algorithm via duality. *Metrika*, 36(1):29–54, 1989.

[15] 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.

[16] S. Jegelka, F. Bach, and S. Sra. Reflection methods for user-friendly submodular optimization. In *Adv. NIPS*, 2013.

[17] N. Komodakis, N. Paragios, and G. Tziritas. MRF energy minimization and beyond via dual decomposition. *IEEE TPAMI*, 33(3):531–552, 2011.

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

[19] K. S. S. Kumar, A. Barbero, S. Jegelka, S. Sra, and F. Bach. Convex optimization for parallel energy minimization. Technical Report 01123492, HAL, 2015.

[20] H. Lin and J. Bilmes. A class of submodular functions for document summarization. In *NAACL/HLT*, 2011.

[21] L. Lovász. Submodular functions and convexity. *Mathematical programming: the state of the art, Bonn*, pages 235–257, 1982.

[22] R. T. Rockafellar. *Convex Analysis*. Princeton U. P., 1997.

[23] X. Shusheng. Estimation of the convergence rate of Dykstras cyclic projections algorithm in polyhedral case. *Acta Mathematicae Applicatae Sinica (English Series)*, 16(2):217–220, 2000.

[24] P. Stobbe and A. Krause. Efficient minimization of decomposable submodular functions. In *Adv. NIPS*, 2010.

[25] R. Tarjan, J. Ward, B. Zhang, Y. Zhou, and J. Mao. Balancing applied to maximum network flow problems. In *European Symp. on Algorithms (ESA)*, pages 612–623, 2006.