Cut Pursuit: fast algorithms to learn piecewise constant functions

Loic Landrieu[1,2]
[1]Inria - Sierra Project-Team
Ecole Normale Supérieure
Paris, France

Guillaume Obozinski[2]
[2]Université Paris-Est, LIGM
Ecole des Ponts ParisTech
Champs-sur-Marne, France

Abstract

We propose working-set/greedy algorithms to efficiently find the solutions to convex optimization problems penalized respectively by the total variation and the Mumford Shah boundary size. Our algorithms exploit the piecewise constant structure of the level-sets of the solutions by recursively splitting them using graph cuts. We obtain significant speed up on images that can be approximated with few level-sets compared to state-of-the-art algorithms.

1 Introduction

Estimation or approximation with piecewise constant functions has many applications in image and signal processing, machine learning and statistics. In particular, the assumption that natural images are well modeled by functions with bounded total variation motivates the use of the latter as a regularizer, which leads to piecewise constant images for discrete approximations. Moreover a number of models used in medical imaging (El-Zehiry and Elmaghraby, 2007) assume more directly piecewise constant images. More generally piecewise constant models can be used for compression, for their interpretability and finally because they are typically adaptive to the local regularity of the function approximated (Wang et al., 2014). Piecewise constant functions display a form of structured sparsity since their gradient is sparse.

Both convex and non-convex formulations have been proposed to learn function with sparse gradients, the most famous being (a) the formulation of Rudin et al. (1992), hereafter referred to as ROF, who proposed to regularize with the total variation, and (b) the formulation of Mumford and Shah (Mumford and Shah, 1989) who proposed to penalize with the total length of discontinuities of piecewise smooth functions. A fairly large literature is devoted to these formulations mainly in the image processing and optimization literature. Although the connection between the total variation, the Mumford-Shah energy and graph cuts is today well established, algorithms that leverage this connection are relatively recent. In particular, for ROF, Chambolle and Darbon (2009); Goldfarb and Yin (2009) use the fact that the problem can be formulated as a parametric max-flow. El-Zehiry and Grady (2011) use graph cuts to solve the formulation of Mumford and Shah for the case of two components.

The literature on sparsity in computational statistics and machine learning has shown how the sparsity of the solutions sought can be exploited to design algorithms which use parsimonious computations to solve the corresponding large scale optimization problem with significant speed-ups (Bach et al., 2012). Our work is motivated by the fact that this has to the best of our knowledge not been fully leveraged to estimate and optimize with piecewise constant functions. In the convex case, the algorithm proposed to exploit sparsity are working set algorithms and the related Frank-Wolfe. In the non-convex case forward selection algorithms such as OMP, OLS, CoSamp, FoBa and others have been proposed (Mallat and Zhang, 1992; Needell and Tropp, 2009; Zhang, 2009)[1].

It is well understood that algorithms for the convex and non-convex case are in fact fairly related. In particular, for a given type of sparsity the forward step of working set methods, Frank-Wolfe or greedy algorithms is typically the same, and followed by the resolution of a reduced problem.

Appearing in Proceedings of the 19^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2016, Cadiz, Spain. JMLR: W&CP volume 41.

[1]Proximal methods that perform soft-thresholding or the non-convex IHT methods maintain sparse solutions as well, but typically need to update a full dimensional vector at each iteration, which is why we do not include them here. They blend however very well with active set algorithms.

Given their similarity, we explore in this paper both greedy and working set strategies to solve respectively the problems penalized by the Mumford-Shah penalty for piecewise constant functions (aka minimal partition problem) and the optimization problem regularized by the total variation. In the convex case, our algorithms do not apply only to the case where the data fitting term is the MSE or a separable smooth convex function, for which some efficient algorithms implicitly exploiting sparsity exist (Chambolle and Darbon, 2009; Bach, 2013; Kumar and Bach, 2015), but also to a general smooth convex term.

Our algorithms are very competitive for deblurring and are applicable for the estimation of piecewise constant functions on general weighted graphs.

1.1 Notations

We will denote $G = (V, E, w)$ an unoriented weighted graph whose edge set is of cardinality m and $V = [1, \cdots, n]$. For convenience of notation and proofs we encode the undirected graph G as a directed graph with oriented edges in both directions in place of unoriented edges. For a set of nodes $A \subset V$ we denote $\mathbf{1}_A$ the vector of $\{0,1\}^n$ such that $[\mathbf{1}_A]_i = 1$ if and only if $i \in A$. For $F \subset E$ a subset of edges we denote $w(F) = \sum_{(i,j)\in F} w_{ij}$.By extension for two subsets A and B of V we denote $w(A, B) = w(A \times B \cap E)$ the weight of the boundary between those two subsets. Finally we denote $\mathcal{C} \subsetneq 2^V$ the set of all partition of V into connected components in the graph G.

1.2 General problem considered

1.2.1 Problem formulation

We consider in this work the problem of minimizing functions Q of the form $Q(x) := f(x) + \lambda\Phi(x)$ with f differentiable and $\Phi(x)$ a penalty function that decomposes as : $\Phi(x) = \sum_{ij\in E} w_{ij}\phi(x_i - x_j)$ with $\phi : \mathbb{R} \to \mathbb{R}^+$ a sparsity inducing function such that $\phi(0) = 0$. The general problem writes

$$\min_{x \in \mathbb{R}^n} \; f(x) + \frac{\lambda}{2} \sum_{(i,j)\in E} w_{ij}\phi(x_i - x_j). \qquad (1)$$

The function ϕ is typically the absolute value, which corresponds to the total variation, or one minus the Kronecker delta at 0, which leads to the Mumford-Shah penalty for piecewise constant functions. More generally, for functions ϕ that have a non-differentiability at 0, the solution of (1) has sparse gradient and is thus constant on the elements of a certain coarse partition of V. We therefore reformulate the problem for candidate solution that have that property.

1.2.2 Decomposition on a partition

Any $x \in \mathbb{R}^n$ can be written as $x = \sum_{i=1}^k c_i \mathbf{1}_{A_i}$ with $\Pi = \{A_1, \cdots, A_k\} \in \mathcal{C}$ a partition of V into k connected components and $c \in \mathbb{R}^k$. Conversely we say that x can be expressed by partition $\Pi = (A_1, \cdots, A_k)$ if it is in the set $\mathrm{span}(\Pi) := \mathrm{span}(\mathbf{1}_{A_1}, \cdots, \mathbf{1}_{A_k}) = \{\sum_{i=1}^k c_i \mathbf{1}_{A_i} \mid c \in \mathbb{R}^k\}$. We denote $x_\Pi = \arg\min_{z\in\mathrm{span}(\Pi)} Q(z)$ the solution of (1) when x is constrained to be in $\mathrm{span}(\Pi)$. With this notation, we can rewrite problem (1) as the problem of finding an optimal partition $\Pi^\star$:

$$\Pi^\star = \arg\min_{\Pi\in\mathcal{C}} \; Q(x_\Pi) \qquad (2)$$

We say that a partition Π is *coarse* if its cardinality k is such that $k \ll n$.

Before presenting our approach, we review some of the main ideas relevant in the related literature.

1.3 Related work

Mumford and Shah (1989) describe an image as *simple* if it can be expressed as a piecewise-smooth function, i.e. if the domain of the image can be partitioned into regions with short contours and such that the image varies smoothly inside of each region. The so-called *minimal partition problem* refers to the particular case where the functions are taken to be constant inside of each region. The setting in which the number of regions is known prior to the optimization, typically two, is known as the Chan-Vese problem and was first solved using active contour methods (Kass et al., 1988). Chan and Vese (2001) propose a level-set based method for the binary case, which has the advantage of foregoing edges and gradient completely, as they are typically very sensitive to noise. This method has since been extended to the so called *multiphase* setting where the number of *phases*, that is of level-sets of the function is a power of two (Vese and Chan, 2002). The resolution of those problems was considerably sped up by the introduction of graph-cut methods, for binary phase (El-Zehiry and Elmaghraby, 2007) and in the multiphase setting (El-Zehiry and Grady, 2011).

A formulation which can be considered a posteriori very related to the one of Mumford and Shah was proposed by Rudin, Osher and Fatemi (Rudin et al., 1992), and consists in regularizing the estimated image with the total variation. Their celebrated approach boasts numerous applications in various fields such as vision and signal processing, and is solved in the recent literature using proximal splitting (Chambolle and Pock, 2011; Raguet et al., 2013).

The relation between graph-cuts and the total variation goes back to Picard and Ratliff (1975) but

has been only fully exploited recently, when (Chambolle and Darbon, 2009) and Goldfarb and Yin (2009) among others, have exploited the fact that the ROF model can be reformulated as a parametric maximum flow problem; the latter entails that it can be solved by a divide-and-conquer strategy which requires to solve a sequence of max-flow problems on the same graph, thus allowing for efficient reuse of partial computation with a push-relabel algorithm. These results on the total variation are actually an instance of results that apply more generally to submodular functions (Bach, 2013). Indeed, the intimate relation existing between the total variation and graph-cuts is due fundamentally to the fact that the total variation is the Lovász extension of the value of the cut which is a submodular function. Recent progress made on efficient optimization of submodular function produced new fast algorithm for the total variation (Kumar and Bach, 2015; Jegelka et al., 2013).

Problems regularized by the total variation or the Mumford-Shah energy are both related to Potts model. Indeed, if the values of the level-sets are quantized, the corresponding energy to minimize is that of a discrete valued conditional random field (CRF), with as many values as there are quantization levels (Ishikawa, 2003; Chambolle et al., 2010). A number of optimization techniques exist for CRFs (Szeliski et al., 2006). One of the fastest is the α-expansion algorithm of Boykov et al. (2001), which is relying on graph-cut algorithms of Boykov and Kolmogorov (2004).

In the sparsity literature a number of algorithms have been proposed to take advantage computationally of the sparsity of the solution. In the convex setting notable examples are the LARS homotopy algorithms (Efron et al., 2004) to compute the regularization path of the Lasso problem or working set algorithms such Glmnet (Friedman et al., 2010) that computes efficiently the solutions to ℓ_1 regularized problems. The Frank-Wolfe algorithm has also be used to exploit the sparsity of solution of optimization problem in the constrained setting (Jaggi, 2013) as well as in the regularized setting (Harchaoui et al., 2015). In the non-convex setting greedy approaches that are used to compute sparse partial solutions count forward selection algorithms such as orthogonal matching pursuit (Mallat and Zhang, 1992), orthogonal least squares (Chen et al., 1991) and related algorithms (Needell and Tropp, 2009), and algorithms such as SBR (Soussen et al., 2011) and FoBa (Zhang, 2009), which use backwards steps to remove previously introduced variables that are no longer relevant. See Bach et al. (2012) for a review.

2 A working set algorithm for total variation regularization

We propose to solve the minimization of a differentiable function f regularized by a weighted total variation of the form $\mathrm{TV}(x) = \frac{1}{2}\sum_{ij\in E} w_{ij}\,|x_i - x_j|$. Our working set algorithm alternates between solving a reduced problem of the form $\min_{x\in\mathrm{span}(\Pi)} Q(x)$ for $Q(x) = f(x) + \lambda \mathrm{TV}(x)$, and refining the partition Π. We will discuss in Section 2.3 how to solve the reduced problem efficiently, but first present a criterion to refine the partition Π. The propositions of the next section are proved in the appendix.

2.1 Steepest binary cut

To obtain a new partition, and given the current solution $x_\Pi = \arg\min_{x\in\mathrm{span}(\Pi)} Q(x)$ of the reduced problem we consider updates of x of the form $x_\Pi + h\,u_B$ with $u_B = \gamma_B 1_B - \gamma_{B^c} 1_{B^c}$ for some set $B \subset V$ and some scalars h, γ_B and γ_{B^c} such that $\|u_B\|_2 = 1$. We discuss later (in section 2.2) how the choice of B leads to a new partition and focus first on a rationale to choose B. A natural criterion is to choose the set B such that Q decreases the most in the direction of u_B. If we let $Q'(x,v) = \lim_{h\to 0} h^{-1}(Q(x+hv) - Q(x))$ so that, when $d \in \mathbb{R}^n$ is a unit vector, $Q'(x,d)$ denotes the directional derivative of Q at $x \in \mathbb{R}^n$ in the direction d, then this criterion requires to solve $\min_{B\subset V} Q'(x_\Pi, u_B)$. Note that since $Q'(x, \mathbf{1}_\varnothing) = 0$ then $\min_{B\subset V} Q'(x, \mathbf{1}_B) \leq 0$.

To further characterize Q' we decompose the objective function. For a current value of x, we denote by $S(x) := \{(i,j) \in E \mid x_i \neq x_j\}$ the set of edges connecting nodes with different values, which is arguably the appropriate notion of support for our setting, and $S^c(x) := E \backslash S(x)$. Setting $S := S(x_\Pi)$ and given that the absolute value is differentiable everywhere except at zero, we can split Q into two parts, Q_S and $\mathrm{TV}_{|S^c}$, which are respectively differentiable and non-differentiable at x_Π

$$\begin{cases} Q_S(x) & = f(x) + \frac{\lambda}{2}\sum_{(i,j)\in S} w_{ij}\,|x_i - x_j| \\ \mathrm{TV}_{|S^c}(x) & = \frac{\lambda}{2}\sum_{(i,j)\in S^c} w_{ij}\,|x_i - x_j| \end{cases}$$

$\mathrm{TV}_{|S^c}$ is a weighted total variation on the graph G but with weights w_{S^c} such that $[w_{S^c}]_{i,j} = w_{ij}$ for $(i,j) \in S^c$ and 0 else. We extend the previous notations and define $w_{S^c}(A,B) = w_{S^c}(A \times B) = w((A \times B) \cap S^c)$.

Proposition 1. *For $x \in \mathbb{R}^n$, if we set $S = S(x)$ then*

$$Q'(x, 1_B) = \langle \nabla Q_S(x), \mathbf{1}_B \rangle + \lambda w_{S^c}(B, B^c).$$

Moreover if $\langle \nabla f(x), 1_B \rangle = 0$ then

$$Q'(x, u_B) = (\gamma_B + \gamma_{B^c})\, Q'(x, 1_B).$$

Considering the case of $x = x_\Pi$ then for $S = S(x_\Pi)$ clearly $\nabla Q_S(x_\Pi)$ is orthogonal to span(Π) and thus to $\mathbf{1}$. Therefore, by the previous lemma, finding the steepest descent direction of the form u_B requires to solve

$$\min_{B\subset V} (\gamma_B + \gamma_{B^c})\, Q'(x_\Pi, \mathbf{1}_B).$$

To keep a formulation which remains amenable to efficient computations, we will assume that $\gamma_B + \gamma_{B^c}$ is constant or ignore this factor[2]. This leads us to define a *steepest binary cut* as any cut (B_Π, B_Π^c) such that

$$B_\Pi \in \arg\min_{B\subset V} \langle \nabla Q_S(x_\Pi), \mathbf{1}_B\rangle + \lambda w_{S^c}(B, B^c).$$

If $\varnothing$ is a solution, we set $B_\Pi = \varnothing$. As formulated, this problem can be interpreted as a minimum cut problem in a suitably defined flow graph. Indeed consider $G_{flow} = (V \cup \{s,t\}, E_{flow})$ where s and t are respectively a source node and a sink node, and where the edge set E_{flow} and the associated nonzero (undirected) capacities $c \in \mathbb{R}^{|S^c|+n}$ are defined as follows:

$$E_{flow} = \begin{cases} (s,i), \forall i \in \nabla_+ & c_{si} = \nabla_i Q_S(x) \\ (i,t), \forall i \in \nabla_- & c_{it} = -\nabla_i Q_S(x) \\ (i,j), \forall (i,j) \in S^c & c_{ij} = \lambda w_{ij}, \end{cases} \quad (3)$$

with $\nabla_+ = \{i \in V \mid \nabla_i Q_S(x) > 0\}$ and $\nabla_- = V \backslash \nabla_+$.

Proposition 2. *Let $S = S(x)$ then $(C, V_{flow}\backslash C)$ is a minimal cut in G_{flow} if and only if $C\backslash\{s\}$, and its complement in V are minimizers of $B \mapsto Q'(x, \mathbf{1}_B)$.*

We can now characterize the optimality of the partition Π with its steepest binary partition B_Π.

Proposition 3. *We have $x = \arg\min_{z\in\mathbb{R}^n} Q(z)$ if and only if $\min_{B\subset V} Q'(x, \mathbf{1}_B) = 0$ and $Q'(x, \mathbf{1}_V) = 0$.*

This guarantees that if we fail to find a steep cut, the algorithm has reached its optimum.

2.2 Induced new partition in connected sets

For $\Pi = (A_1, \cdots, A_k)$ we motivated the choice of B_Π by the best addition of a term for the form hu_B to $x = \sum_{i=1}^k c_i \mathbf{1}_{A_i}$. At the next iteration, we could thus consider minimizing Q under the constraint that $x \in \text{span}(\mathbf{1}_{A_1}, \ldots, \mathbf{1}_{A_k}, \mathbf{1}_B)$ with $B = B_\Pi$. But, on this set, the values $x_{i_1}, x_{i_2}, x_{i_3}$ and x_{i_4} with $i_1 \in A_j \cap B$, $i_2 \in A_j \cap B^c$, $i_3 \in A_{j'} \cap B$ and $i_4 \in A_{j'} \cap B^c$ are a priori coupled; also, if $A_j \cap B$ has several connected components, $i \mapsto x_i$ must take the same value on these components. These constraints are unnecessarily restrictive. Indeed, since the right notion of sparsity of the model is arguably the set of edges that can be non zeros $S_{new} = S(x) \cup (B \times B^c)$, it makes sense to only constrain the variables to stay in the set $\mathcal{X}_{S_{new}} = \{x' \mid S(x') = S_{new}\}$. If we define Π_{new} as the collection of all connected components in G of all sets $A_j \cap B_\Pi$ and $A_j \cap B_\Pi^c$ for $A_j \in \Pi$ then it can be easily shown that $\text{span}(\Pi_{\text{new}}) = \mathcal{X}_{S_{\text{new}}}$.

Proposition 4. *If $B_\Pi \neq \varnothing$, $Q(x_{\Pi_{\text{new}}}) < Q(x_\Pi)$.*

Algorithm 1: Working set scheme

Initialize $\Pi \leftarrow \{V\}$, $x_\Pi \in \arg\min_{z=c\mathbf{1}_V, c\in\mathbb{R}} Q(z)$
while $\min_{B\subset V} Q'(x_\Pi, 1_B) < 0$ **do**
  Pick $B_\Pi \in \arg\min_{B\subset V} Q'(x_\Pi, 1_B)$
  $\Pi \leftarrow \{B_\Pi \cap A\}_{A\in\Pi} \cup \{B_\Pi^c \cap A\}_{A\in\Pi}$
  $\Pi \leftarrow$ connected components of elements of Π
  Pick $x_\Pi \in \arg\min_{z\in\text{span}(\Pi)} Q(z)$
return (Π, x_Π)

We thus obtain Algorithm 1, illustrated on Figure 1. Based on the previous propositions, this algorithm guarantees monotonic convergence to the optimum $\Pi^\star$. In terms of complexity, at each iteration Π_{new} has at least one more component than Π, so that the algorithm converges in at most n steps. We now discuss how to exploit the sparse structure of x_Π to solve the reduced problem efficiently.

2.3 A reduced graph for the reduced problem

Let Π be a coarse[3] partition of V into connected components. We argue that $\min_{z\in\text{span}(\Pi)} Q(z)$ can be solved on a weighted graph whose nodes are associated with the elements of the partition Π. Indeed, consider the graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $\mathcal{V} = \Pi$ and $\mathcal{E} = \{(A,B) \in \mathcal{V}^2 \mid \exists (i,j) \in (A\times B) \cap E\}$. For $x \in \text{span}(\Pi)$ we can indeed express $\text{TV}(x)$ simply.

Proposition 5. *For $x = \sum_{A\in\Pi} c_A \mathbf{1}_A$ and $2\text{TV}_\mathcal{G}(c) := \sum_{(A,B)\in\mathcal{E}} w(A,B)\,|c_A - c_B|$ we have $\text{TV}(x) = \text{TV}_\mathcal{G}(c)$.*

Note that if TV is the total variation associated with the weighted graph G with weights $(w_{ij})_{(i,j)\in E}$ then $\text{TV}_\mathcal{G}$ is the total variation associated with the weighted graph $\mathcal{G}$ and the weights $\big(w(A,B)\big)_{(A,B)\in\mathcal{E}}$. Denoting $\tilde{f} : c \mapsto f(\sum_{A\in\Pi} c_A 1_A)$, the reduced problem is equivalent to solving $\min_{c\in\mathbb{R}^k} \tilde{f}(c) + \lambda \text{TV}_\mathcal{G}(c)$ on $\mathcal{G}$. Since Π is coarse, we have $|\mathcal{E}| \ll m$ and hence computations involving $\text{TV}_\mathcal{G}$ are much cheaper than those involving TV. As discussed in Section 2.4 the structure of $\tilde{f}$ can often be exploited as well. $\mathcal{G}$ is cheap to compute

[2] γ_B and γ_{B^c} could otherwise be determined by requiring that $\langle 1, u_B\rangle = 0$. More rigorously, descent directions considered could be required to be orthogonal to span(Π), but this leads to even less tractable formulations, that we therefore do not consider here.

[3] A coarse partition corresponds to a sparse solution.

(a) (b) (c) (d) (e)

Figure 1: Two first iterations of cut pursuit for the ROF problem on image (a). Image (b) and (d) represent the new cut at iterations 1 and 2, (c) and (e) the partial solution, with the current set of contours S in red.

comparatively to the speed ups allowed, as it is obtained by computing the connected components of the graph $(V, E\backslash S(x))$, which can be done in linear time by depth-first search.

2.4 Solving linear inverse problems with TV

A number of classical problems in image processing such as deblurring, blind deconvolution and inpainting are formulated as ill-posed linear inverse problems (Chan et al., 2005), where a low TV prior on the image provides appropriate regularization. Typically if x_0 is the original image, H a blurring linear operator typically computed as a convolution on the image, ϵ additive noise and $y = Hx_0 + \epsilon$ the degraded image, this leads to problems of the form

$$x^\star = \operatorname*{arg\,min}_{x\in\mathbb{R}^n} \frac{1}{2}\|Hx - y\|^2 + \lambda \mathrm{TV}(x). \tag{4}$$

Since n is large, manipulating directly the matrices H or $H^\intercal$ should be avoided, but if the forward operator $x \mapsto Hx$ is a convolution, it can be computed quickly using e.g. the fast Fourier transform. In the case of a blurring operator with adequate symmetry $H^\intercal Hx$ is also a blurring operator. These fast computations of $x \mapsto Hx$ can be exploited in our algorithm to compute very efficiently the Hessian and loading vector of the reduced problem. Indeed, For a k-partition Π of V we denote by $K \in \{0,1\}^{n,k}$ the components matrix whose columns are the vectors $\mathbf{1}_A$ for $A \in \Pi$. Any $x \in \mathrm{span}(\Pi)$ can be rewritten as Kc with $c \in \mathbb{R}^k$. The gradient of the discrepancy function with respect to c writes: $\nabla_c 1/2\|HKc - y\|^2 = K^\intercal H^\intercal HKc - K^\intercal Hy$ As a result, the reduced problem can be solved by a similar forward backward scheme of much smaller size, with parameter $K^\intercal H^\intercal HK$ and $K^\intercal Hy$, which are of size $k \times k$ and k respectively.

3 Minimal partition problems

We consider now a generalization of the minimal partition problem of the form $\min_{x\in\mathbb{R}^n} Q(x)$ with $Q(x) = f(x) + \lambda\Gamma(x)$ where $\Gamma(x) = \frac{1}{2}\sum_{(i,j)\in S(x)} w_{ij}$ the Mumford-Shah penalty. This non-convex non-differentiable problem being significantly harder than the previous one, we restrict the functions f we consider to be separable functions of the form $f(x) = \sum_{i\in V} f_i(x_i)$ with $f_i : \mathbb{R} \mapsto \mathbb{R}$ continuously differentiable and convex. Inspired by greedy algorithms in the sparsity literature such as OMP, and by the working set algorithm we presented for TV regularization, we propose to exploit that $|\Pi^*|$ is not too large to construct an algorithm that greedily optimizes the objective by adding and removing cuts in the graph.

3.1 A greedy algorithm

As for the working set algorithm, we propose to gradually build an expansion of x of the form $x = \sum_{i=1}^k c_i 1_{A_i}$ for $\Pi = (A_1, \cdots, A_k)$ a partition of V, by recursively splitting some of the existing sets $A \in \Pi$. As a first remark, given the separability assumption on f the choice of a global binary cut of a current partition Π reduces to cut individually all elements of Π, and the optimal cut for each $A \in \Pi$ considered does furthermore not depend on x_{A^c} or $\Pi\backslash\{A\}$. We should thus focus on cutting a single set A at a time. Without loss of generality, we consider the case $\Pi = \{V\}$.

3.1.1 Optimal binary cut

The optimal binary partition (B, B^c) of V is such that Q optimized over $\mathrm{span}(\mathbf{1}_B, \mathbf{1}_{B^c})$ is as small as possible. Since $\Gamma(h\mathbf{1}_B + h'\mathbf{1}_{B^c}) = \Gamma(\mathbf{1}_B) = w(B, B^c)$ the corresponding optimization problem is of the form

$$\min_{B\subset V,\ h,h'\in\mathbb{R}} \sum_{i\in B} f_i(h) + \sum_{i\in B^c} f_i(h') + \lambda w(B, B^c). \tag{5}$$

This problem is a priori hard to solve in general ($B \mapsto \min_{h,h'\in\mathbb{R}} f(h\mathbf{1}_B + h'\mathbf{1}_{B^c})$ is not submodular). However, when h, h' are fixed the assumption that f is separable entails that $B \mapsto f(h\mathbf{1}_B + h'\mathbf{1}_{B^c})$ is a modular function, so that the objective can be optimized with respect to B by solving a max-flow problem similar to (3) where $f_i(x_i)$ is substituted for $\nabla_i Q_S(x)$. The smoothness and convexity of f w.r.t. h and h' guarantee that the objective can be minimized efficiently

w.r.t. these variables. A local minimum of the objective can thus be obtained efficiently by alternatively minimizing with respect to B and (h, h') as suggested by Bresson et al. (2007) or El-Zehiry et al. (2011). One issue is again the fact that the binary partition B obtained as a solution of (5) and its complement are not necessarily connected sets, and it would be intractable to optimize only over pairs of sets (B, B^c) that are connected. However, splitting B and B^c into their connected components will not increase the contour length Γ and can only decrease f. Consequently, given the collection of connected components $A_1, \ldots, A_k$ of B and B^c we set $x = h_1 1_{A_1} + \ldots + h_k 1_{A_k}$ with h_j the minimizer of $h \mapsto \sum_{i \in A_j} f_i(h)$.

3.1.2 Recursive splitting and merging

Like for the working set algorithm, we recursively split the components of the current partition Π, however, in this case, as discussed earlier, only one connected component is cut at a time and the new boundary introduced is within that component. Our algorithm is structured by the following observations:

Optimal cuts of existing components are independents. Given the separability assumption on f the problem of the choice of an optimal cut in each component are independent and need to be recomputed only for the newly created components.

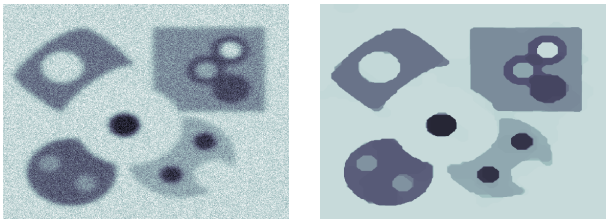
Merging components. We consider a form of backward step, which merges a pair (A, B) of elements of Π and therefore removes the boundary between A and B. The step is taken if the increase of f induced is smaller than $\lambda w(A, B)$. This step is similar to the backward step of the Single Best Replacement algorithm by Soussen et al. (2011) which considers at each iteration the addition or removal of a single variable, whichever reduces most the value of the objective. Our approach differs in the sense that the merge we consider has no reason to correspond to a previous cut, since we consider all pairs of adjacent components.

Saturated components. If the optimal binary partition of a component is the empty set, this component is said to be *saturated*. The separability of the fidelity function ensures that this component will not longer be split and can be ignored for the rest of the computation, unless it is involved in a mergeing step. The resulting algorithm, ℓ_0 Cut Pursuit is detailed in the supplementary material.

4 Experiments

4.1 Deblurring experiments with TV

To assess the performance in terms of speed of our working set algorithm we compare it with several state-of-the-art algorithms on a deblurring task of the form presented in section 2.4. Specifically, given an image x, we compute $y = Hx + \epsilon$, where H is a Gaussian blur matrix, and ϵ is some Gaussian additive noise, and we solve (1) with a total variation regularization based on the 8-neighborhood graph built on image pixels. We use three 512×512 images of increasing complexities to benchmark the algorithms: the Shepp-Logan phantom, a simulated example and Lena represented in Figure 2 and 5 (all figures are represented in full size in the appendix). For all images the standard deviation of the blur is set at 5 pixels.



(a) PSNR : 15.9 (b) PSNR : 27.2

Figure 2: (a) Simulated example with Gaussian blur (b) deblurred image obtained by Cut Pursuit (CP).

4.1.1 Competing methods

Preconditioned Generalized Forward Backward (PGFB). As a general baseline, we consider a recent preconditioned generalized forward-backward splitting algorithm by Raguet and Landrieu (2015) whose prior non-preconditioned version was shown to outperform state-of-the art convex optimization on deblurring tasks in Raguet et al. (2013).

Accelerated FB with parametric max-flows (FB+). Since efficient algorithms that solve the ROF problem have been the focus of recent work, and given that the ROF problem corresponds to the computation of the proximal operator of the total variation, we also make a comparison with an implementation of the accelerated forward-backward algorithm of Nesterov (2007). To compute the proximal operator, we use an efficient solver of the ROF problem based on a reformulation as a parametric max-flow proposed by Chambolle and Darbon (2009). The solver we used is the one made publicly available by the authors.

Cut Pursuit with Frank-Wolfe descent directions (CP-FW). We consider an alternative to the steepest binary partition to split the existing components of the partial solution. Inspired by a version of the Frank-Wolfe algorithm for regularized problems proposed by Harchaoui et al. (2015), we add the cut corresponding to the point of the set $\{x \in \mathbb{R}^n \mid \Phi(x) \leq 1\}$ with minimal dot product with the gradient direction, which is itself computed as the solution of a maximum flow problem.

Cut Pursuit. To implement our algorithm (CP), we solve the min-cut problems with the solver of Kohli and Torr (2005) which itself is based on Boykov et al. (2001) and Kolmogorov and Zabih (2004). The problems on the reduced graph are solved using the PGFB algorithm. This last choice is motivated by the fact that the preconditioning is quite useful as it compensates for the fact that the weights on the reduced graph can be quite imbalanced.

4.1.2 Results

Figure 3 shows the speed of the different algorithms on the three test images on a quad core CPU at 2.4 Ghz. We represent the relative primal suboptimality gap $(Q_t - Q_\infty)/Q_\infty$ where Q_∞ is the lowest value obtained by CP in 100 seconds. We can see that our algorithm speeds up significantly the direct optimization approach (PGFB) when the solution is sparse, and that it remains competitive in the case of a natural image with strong regularization. Indeed since the reduced problems are of much smaller size than the original, our algorithm can perform many more forward-backward iterations in the same allotted time. The variant of Cut Pursuit using Frank-Wolfe directions (CP-FW) is as efficient over the first few iterations but then stagnates. The computation of a new Frank-Wolfe direction does not take into account the current support $S(x)$, that provides a set of edges that are "free"; this entails that it overestimates the cost of adding new boundaries, resulting in too conservative updates. Accelerated forward-backward with parametric max-flow FB+ is slower in this setting, which is explained by the small number of updates the algorithm can afford in the allotted time given the cost of solving parametric max-flows; PGFB can make more than 50 updates during each iteration of FB+.

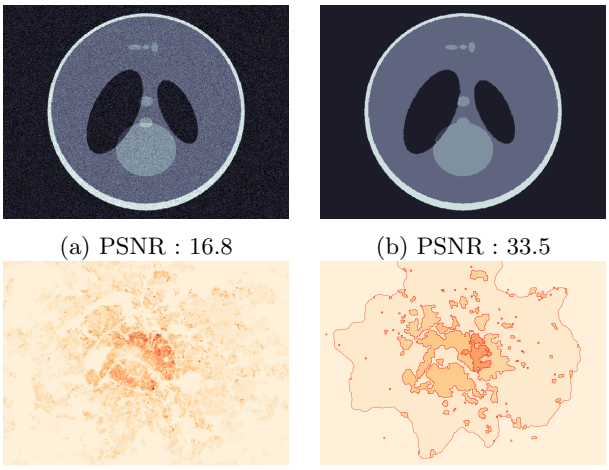
We report and discuss in the appendix the breakdown of computation time for each algorithm. Most notably the actual time spent by Cut Pursuit solving the reduced problem (with PGFB) takes comparatively very little time (around 3%) when this is the only step actually decreasing the objective function.

4.2 Experiments on minimal partitions

4.2.1 Denoising experiment

We now present experiments demonstrating the efficiency of the ℓ_0-Cut Pursuit presented in section 3. We assess its performance against two state-of-the-art algorithms to minimize the Mumford-Shah energy of two noisy 512×512 images: the phantom of Shepp and Logan (1974) and another simulated example. In order to illustrate the advantage of our algorithm over alternatives that discretize the value range, we make a small random shift of grey values to both images.

We also test the algorithms on a spatial statistic aggregation problem which corresponds to the design of a simplified map of the population density in the Paris area. We use raster open-source data[4] producing the map of Figure 5. The raster is partitionned using constrained Delaunay triangulation (Chew, 1989) to obtain a graph with 252,183 nodes and 378,258 edges. We use the squared loss weighted by the surface of each triangle as a fidelity term.



(a) PSNR : 16.8 (b) PSNR : 33.5

(c) Paris population (d) Simplified map

Figure 5: (a) Noisy Shepp-Logan phantom, (b) the result of ℓ_0-CP, (c) triangulated population density in Paris area and (d) the simplified map obtained by ℓ_0-CP (69% of the variance explained with 1.2% of the contour length).

4.2.2 Competing methods

α-expansions on quantized models. Following Ishikawa (2003), we use the fact that if the range of values of x_i is discretized, the MPP and TV problems reduce to maximum a posterior inference in a Potts model. More precisely it is a Potts model in which each value taken by the class variable c_i is associated with a (non necessarily connected) level-set, and that has pairwise terms of the form $1_{\{c_i \neq c_j\}} w_{ij}$. We use α-expansions (Boykov et al., 2001) to minimize the corresponding energy. More precisely we use the α-expansions implementation of Fulkerson et al. (2009), which uses the same max-flow code (Boykov and Kolmogorov, 2004) as our algorithm. We denote the resulting algorithm (CRFi) where i is the number of quantization levels.

Non-convex relaxation. We implemented a non-convex relaxation of the Mumford-Shah functional which is a "concave" version of the total variation, such as the adaptive Lasso (Zou, 2006) with $t \mapsto (\epsilon + t)^{\frac{1}{2}}$

[4] https://www.data.gouv.fr/fr/datasets/donnees-carroyees-a-200m-sur-la-population

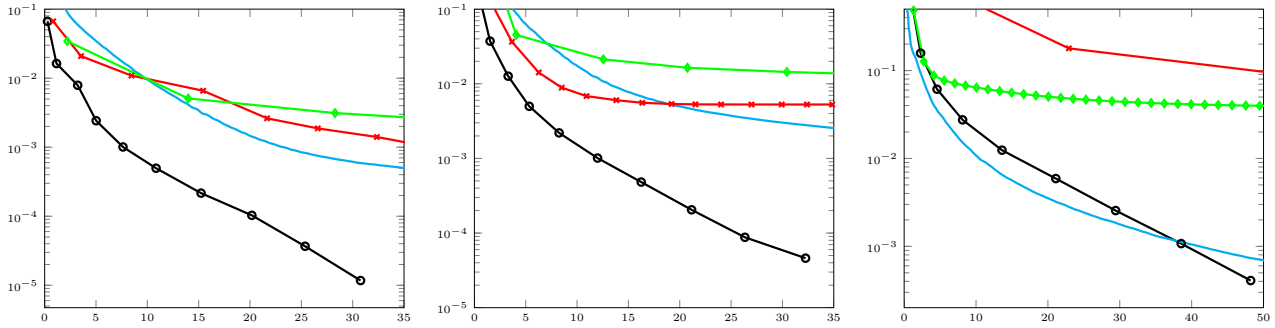


Figure 3: Relative primal suboptimality gap $Q_t/Q_\infty - 1$ at time t (in seconds) for different algorithms on the deblurring task: FB+: , PGFB: , CP: CPFW: , for three 512×512 images and different regularization values: Shepp-Logan phantom (left), our simulated example (middle) and Lena (right). Each marker corresponds to a cut computation for CP, CPFW, and a proximal operator computation for FB+.

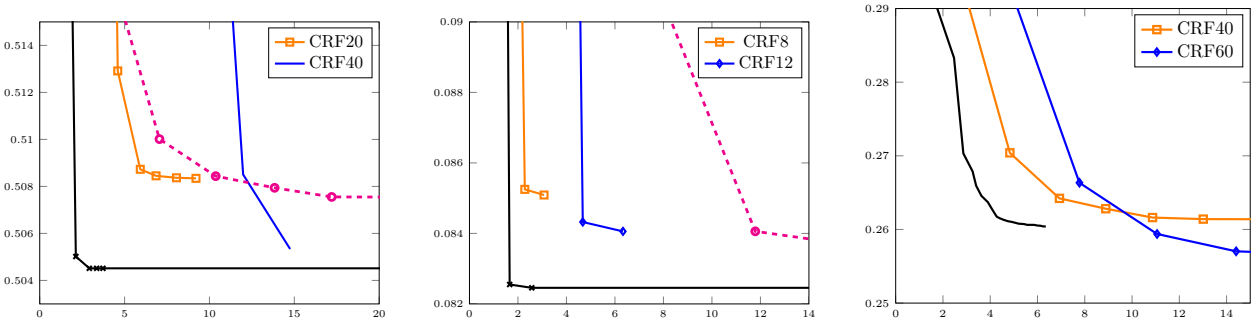


Figure 4: Mumford-Shah energy at time t (in seconds) divided by the same energy for the best constant image obtained with: Non-convex relaxation ($\text{TV}_{0.5}$) , ℓ_0 cut pursuit (ℓ_0-CP) and α-expansions (CRFi), for the Shepp-Logan phantom (left), our simulated example (middle) and the map simplification (right). Markers corresponds respectively to one reweighting, one α-expansion cycle and one cut for ($\text{TV}_{0.5}$), (CRF) and (ℓ_0-CP).

in lieu of $t \mapsto |t|$. The resulting functional can be minimized locally using a reweighted TV (Ochs et al., 2015). We use our Cut Pursuit algorithm to solve each reweighted TV problems, since it was the fastest implementation. This algorithm is denoted $\text{TV}_{0.5}$.

4.2.3 Results

We report on Figure 4 the energy obtained by the different algorithms normalized by the energy of the best constant approximation. We can see that our algorithms find faster local optima that are essentially as good or better than the ones obtained using α-expansions on the discretized problem, as long as the solutions are sufficiently sparse. For the population density data our algorithm performs roughly as well as CRF40 but is outperformed by CRF60.

5 Conclusion

We proposed two algorithms to minimize functions penalized respectively by the total variation and by the Mumford-Shah boundary size. They exploit that the solution typically has a sparse gradients, which entails that it has a small number of connected level-sets. By constructing a sequence of approximate solutions that have the same property, they operate on reduced problems that can be solved efficiently, and require only to perform a number of graph cuts on the original graph, which are the bottleneck for further speed-ups. Like other working set schemes, our algorithms are not competitive if the solution has too many connected level-sets. In future work, we intend to extend the approach presented in this paper to approximate efficiently the total variation regularization path. It would also be interesting to find guarantees similar toS those existing for α-expansions to our greedy algorithm.

Experiment	Phantom		Simulated	
Algorithm	PSNR	time	PSNR	time
Noisy image	16.8	-	16.8	-
ℓ_0-CP	**33.5**	**4.3**	**37.0**	4.6
CRF20/CRF8	32.6	8.6	34.2	**4.0**
CRF40/CRF12	33.3	25.3	34.8	11.4
$TV_{0.5}$	32.2	16.4	33.6	18.0

Figure 6: PSNR at convergence and time to converge in seconds for the four algorithms as well as the noisy image for the first two denoising experiments.

References

Bach, F. (2013). Learning with submodular functions: a convex optimization perspective. *Foundations and Trends in Machine Learning*, 6(2-3):145–373.

Bach, F., Jenatton, R., Mairal, J., and Obozinski, G. (2012). Optimization with sparsity-inducing penalties. *Foundations and Trends in Machine Learning*, 4(1):1–106.

Boykov, Y. and Kolmogorov, V. (2004). An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1124–1137.

Boykov, Y., Veksler, O., and Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239.

Bresson, X., Esedoḡlu, S., Vandergheynst, P., Thiran, J.-P., and Osher, S. (2007). Fast global minimization of the active contour/snake model. *Journal of Mathematical Imaging and Vision*, 28(2):151–167.

Chambolle, A., Caselles, V., Cremers, D., Novaga, M., and Pock, T. (2010). An introduction to total variation for image analysis. In *Theoretical foundations and numerical methods for sparse recovery*, pages 263–340. De Gruyter.

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

Chambolle, A. and Pock, T. (2011). A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145.

Chan, T., Esedoḡlu, S., Park, F., and Yip, A. (2005). Recent developments in total variation image restoration. In *Mathematical Models of Computer Vision*, pages 17–31. Springer Verlag.

Chan, T. F. and Vese, L. A. (2001). Active contours without edges. *IEEE Transactions on Image Processing*, 10(2):266–277.

Chen, S., Cowan, C. F., and Grant, P. M. (1991). Orthogonal least squares learning algorithm for radial basis function networks. *IEEE Transactions on Neural Networks*, 2(2):302–309.

Chew, L. P. (1989). Constrained Delaunay triangulations. *Algorithmica*, 4(1):97–108.

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., et al. (2004). Least angle regression. *The Annals of statistics*, 32(2):407–499.

El-Zehiry, N. and Grady, L. (2011). Discrete optimization of the multiphase piecewise constant Mumford-Shah functional. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 233–246. Springer.

El-Zehiry, N., Sahoo, P., and Elmaghraby, A. (2011). Combinatorial optimization of the piecewise constant Mumford-Shah functional with application to scalar/vector valued and volumetric image segmentation. *Image and Vision Computing*, 29(6):365–381.

El-Zehiry, N. Y. and Elmaghraby, A. (2007). Brain MRI tissue classification using graph cut optimization of the Mumford–Shah functional. In *Proceedings of the International Vision Conference of New Zealand*, pages 321–326.

Friedman, J., Hastie, T., and Tibshirani, R. (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22.

Fulkerson, B., Vedaldi, A., and Soatto, S. (2009). Class segmentation and object localization with superpixel neighborhoods. In *Proceedings of the International Conference on Computer Vision*, pages 670–677. IEEE.

Goldfarb, D. and Yin, W. (2009). Parametric maximum flow algorithms for fast total variation minimization. *SIAM Journal on Scientific Computing*, 31(5):3712–3743.

Harchaoui, Z., Juditsky, A., and Nemirovski, A. (2015). Conditional gradient algorithms for norm-regularized smooth convex optimization. *Mathematical Programming*, 152(1–2):75–112.

Ishikawa, H. (2003). Exact optimization for Markov random fields with convex priors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1333–1336.

Jaggi, M. (2013). Revisiting Frank-Wolfe: projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning*, pages 427–435.

Jegelka, S., Bach, F., and Sra, S. (2013). Reflection methods for user-friendly submodular optimization. In *Advances in Neural Information Processing Systems*, pages 1313–1321.

Kass, M., Witkin, A., and Terzopoulos, D. (1988). Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331.

Kohli, P. and Torr, P. H. (2005). Efficiently solving dynamic Markov random fields using graph cuts. In *International Conference on Computer Vision (ICCV)*, volume 2, pages 922–929. IEEE.

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

Kumar, K. and Bach, F. (2015). Active-set methods for submodular optimization. *arXiv preprint arXiv:1506.02852*.

Mallat, S. and Zhang, Z. (1992). Adaptive time-frequency decomposition with matching pursuits. In *Time-Frequency and Time-Scale Analysis, Proceedings of the IEEE-SP International Symposium*, pages 7–10. IEEE.

Mumford, D. and Shah, J. (1989). Optimal approximations by piecewise smooth functions and associated variational problems. *Communications on pure and applied mathematics*, 42(5):577–685.

Needell, D. and Tropp, J. A. (2009). CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *Applied and Computational Harmonic Analysis*, 26(3):301–321.

Nesterov, Y. (2007). Gradient methods for minimizing composite objective function. Technical report, Université catholique de Louvain, Center for Operations Research and Econometrics (CORE).

Ochs, P., Dosovitskiy, A., Brox, T., and Pock, T. (2015). On iteratively reweighted algorithms for nonsmooth nonconvex optimization in computer vision. *SIAM Journal on Imaging Sciences*, 8(1):331–372.

Picard, J.-C. and Ratliff, H. D. (1975). Minimum cuts and related problems. *Networks*, 5(4):357–370.

Raguet, H., Fadili, J., and Peyré, G. (2013). A generalized forward-backward splitting. *SIAM Journal on Imaging Sciences*, 6(3):1199–1226.

Raguet, H. and Landrieu, L. (2015). Preconditioning of a generalized forward-backward splitting and application to optimization on graphs. *SIAM Journal on Imaging Sciences*, 8(4):2706–2739.

Rudin, L. I., Osher, S., and Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60:259 – 268.

Shepp, L. A. and Logan, B. F. (1974). The Fourier reconstruction of a head section. *IEEE Transactions on Nuclear Science*, 21(3):21–43.

Soussen, C., Idier, J., Brie, D., and Duan, J. (2011). From Bernoulli–Gaussian deconvolution to sparse signal restoration. *IEEE Transactions on Signal Processing*, 59(10):4572–4584.

Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M., and Rother, C. (2006). A comparative study of energy minimization methods for Markov random fields. In *Proceeding of the European Conference in Computer Vision (ECCV)*, pages 16–29. Springer.

Vese, L. A. and Chan, T. F. (2002). A multiphase level set framework for image segmentation using the Mumford and Shah model. *International Journal of Computer Vision*, 50(3):271–293.

Wang, Y.-X., Sharpnack, J., Smola, A., and Tibshirani, R. J. (2014). Trend filtering on graphs. *arXiv preprint arXiv:1410.7690*. To appear in JMLR.

Zhang, T. (2009). Adaptive forward-backward greedy algorithm for sparse learning with linear models. In *Advances in Neural Information Processing Systems*, pages 1921–1928.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429.