Mathematical Foundations of Data Sciences

Gabriel Peyré
CNRS & DMA
École Normale Supérieure
gabriel.peyre@ens.fr
https://mathematical-tours.github.io
www.numerical-tours.com

November 18, 2020

Chapter 17

Non-smooth Convex Optimization

The main references for this chapter are [7, 8, 3], see also [19, 2, 1].

We consider a general convex optimization problem

$$\min_{x \in \mathcal{H}} f(x) \tag{17.1}$$

where $\mathcal{H} = \mathbb{R}^p$ is a finite dimensional Hilbertian (i.e. Euclidean) space, and try to devise "cheap" algorithms with a low computational cost per iterations. The class of algorithms considered are first order, i.e. they make use of gradient information.

17.1 Descent Methods

The gradient descent method is covered in detailed in Section 13.4. We explain here how to extend it to non-smooth and constrained problems.

17.1.1 Gradient Descent

The optimization program (8.26) is an example of unconstrained convex optimization of the form (17.1) where $f : \mathcal{H} \to \mathbb{R}$ is a $\mathcal{C}^1$ function with Lipschitz gradient (so-called "smooth" function). Recall that the gradient $\nabla f : \mathcal{H} \mapsto \mathcal{H}$ of this functional (not to be confound with the discretized gradient $\nabla x \in \mathcal{H}$ of f) is defined by the following first order relation

$$f(x + r) = f(x) + \langle f, r \rangle_{\mathcal{H}} + O(\|r\|_{\mathcal{H}}^2)$$

where we used $O(\|r\|_{\mathcal{H}}^2)$ in place of $o(\|r\|_{\mathcal{H}})$ (for differentiable function) because we assume here f is of class $\mathcal{C}^1$ (i.e. the gradient is continuous). Section 8.4.3 shows typical examples of gradient computation.

For such a function, the gradient descent algorithm is defined as

$$x^{(\ell+1)} \overset{\text{def.}}{=} x^{(\ell)} - \tau_\ell \nabla f(x^{(\ell)}), \tag{17.2}$$

where the step size $\tau_\ell > 0$ should be small enough to guarantee convergence, but large enough for this algorithm to be fast.

17.1.2 Sub-gradient Descent

The gradient descent (17.2) cannot be applied on a non-smooth function f. One can use in place of a gradient a sub-gradient, which defines the sub-gradient descent

$$x^{(\ell+1)} \overset{\text{def.}}{=} x^{(\ell)} - \tau_\ell g^{(\ell)} \quad \text{where} \quad g^{(\ell)} \in \partial f(x^{(\ell)}). \tag{17.3}$$

The main issue with this scheme is that to ensure convergence, the iterate should go to zero. One can easily convince oneself why by looking at the iterates on a function $f(x) = |x|$. This means that in practice this method is very slow, and should be avoided if the functions has some structure that can be used (as it will be explained in the remaining part of the chapter).

Theorem 29. *If $\sum_\ell \tau_\ell = +\infty$ and $\sum_\ell \tau_\ell^2 < +\infty$, then $x^{(\ell)}$ converges to a minimizer of f.*

17.1.3 Projected Gradient Descent

We consider a generic constraint optimization problem as

$$\min_{x \in \mathcal{C}} f(x) \tag{17.4}$$

where $\mathcal{C} \subset \mathbb{R}^S$ is a closed convex set and $f : \mathbb{R}^S \to \mathbb{R}$ is a smooth convex function (at least of class $\mathcal{C}^1$).

The gradient descent algorithm (17.2) is generalized to solve a constrained problem using the projected gradient descent

$$x^{(\ell+1)} \stackrel{\text{def.}}{=} \mathrm{Proj}_{\mathcal{C}}\left(x^{(\ell)} - \tau_\ell \nabla f(x^{(\ell)})\right), \tag{17.5}$$

where $\mathrm{Proj}_{\mathcal{C}}$ is the orthogonal projector on $\mathcal{C}$

$$\mathrm{Proj}_{\mathcal{C}}(x) = \operatorname*{argmin}_{x' \in \mathcal{C}} \|x - x'\|$$

which is always uniquely defined because $\mathcal{C}$ is closed and convex. The following proposition shows that all the convergence properties of the classical gradient descent caries over to this projected algorithm.

Theorem 30. *Theorems ?? and 23 still holds when replacing iterations (17.2) by (17.5).*

Proof. The proof of Theorem ?? extends because the projector is contractant, $\|\mathrm{Proj}_{\mathcal{C}}(x) - \mathrm{Proj}_{\mathcal{C}}(x')\| \leqslant \|x - x'\|$ so that the strict contraction properties of the gradient descent is maintained by this projection. □

The main bottleneck that often prevents to use (17.5) is that the projector is often complicated to compute. We are however lucky since for ℓ^1 mininization, one can apply in a straightforward manner this method.

17.2 Interior Point Methods

We give here an informal description of a class of schemes which converge extremely fast in term of number of iterations, but where each iteration can be very costly. These methods are extensions of the Newton method for unconstrained minimization, but can deal with a specific class of non-smoothness which can be encoded using some specific type of constraints (such as for instance positivity of vector or matrices).

To illustrate the main idea, we consider the following problem

$$\min_{x \in \mathbb{R}^d, Ax \leqslant y} f(x) \tag{17.6}$$

for $A \in \mathbb{R}^{m \times d}$. This can be generalized for instance by replacing Ax by a matrix and $\leqslant$ by PSD matrix inequalities.

Example 3 (Lasso). The Lasso problem (applied to the regression problem $Bw \approx b$ for $B \in \mathbb{R}^{n \times p}$)

$$\min_{w \in \mathbb{R}^p} \frac{1}{2}\|Bw - b\|^2 + \lambda \|w\|_1 \tag{17.7}$$

can be re-casted as (17.6) by introducing the positive/negative decomposition $x = (x_-, x_+) \in \mathbb{R}^{2p}$ (so that $d = 2p$) and $w = x_+ - x_-$ with $(x_+, x_-) \geqslant 0$, so that $f(x) = \frac{1}{2}\|B(x_+ - x_-) - b\|^2 + \lambda \langle x, \mathbb{1} \rangle$ and $A = -\mathrm{Id}_{2p}$ (so that $m = 2p$).

Example 4 (Dual of the Lasso). One can also solve the dual problem to (17.7)

$$\min_{\|B^\top q\|_\infty \leqslant 1} f(q) = \frac{\lambda}{2}\|q\|^2 - \langle q, y\rangle \tag{17.8}$$

so that one can use $A = \begin{pmatrix} B^\top \\ -B^\top \end{pmatrix}$ and $b = \mathbb{1}_{2n}$ (so that $d = p$ and $m = 2n$).

The main idea of interior point methods is to approximate (17.6) using a logarithmic barrier function

$$\min_{x\in\mathbb{R}^p,} f_t(x) \overset{\text{def.}}{=} f(x) - \frac{1}{t}\mathrm{Log}(y - Ax) \tag{17.9}$$

where

$$\mathrm{Log}(u) \overset{\text{def.}}{=} \sum_i \log(u_i)$$

so that -Log is a strictly concave function which acts as a barrier for positivity. On recovers the initial problem in the limit, i.e. in some sense $f + \iota_{A\cdot\leqslant y} = f_\infty$.

For a fixed t, one can solve (17.9) using a Newton method with some line search procedure (the simplest being Armijo backtracking (13.14)) to select the step size $0 < \tau_\ell \leqslant 1$

$$x^{(\ell+1)} \overset{\text{def.}}{=} x^{(\ell)} - \tau_\ell[\partial^2 f_t(x)]^{-1}\nabla f_t(x^{(\ell)}) \tag{17.10}$$

where here

$$\nabla f_t(x) = \nabla f(x) + \frac{1}{t}A^\top \frac{1}{y - Ax} \quad \text{and} \quad \partial^2 f_t(x) = \partial^2 f(x) + \frac{1}{t}A^\top \mathrm{diag}\left(\frac{1}{(y - Ax)^2}\right) A.$$

The natural stopping criterion for this descent step is

$$\langle [\partial^2 f_t(x)]^{-1}\nabla f_t(x^{(\ell)}),\, \nabla f_t(x^{(\ell)})\rangle < \frac{\varepsilon}{2}.$$

The interior point barrier method proceed by f_t using (17.10) (defining an approximated "central" path $t \mapsto x(t)$) for a series of increasing step sizes $t = t_k = \mu^k t_0$ for some $\mu > 1$. The crucial point here is to use a "warm restart" strategy: in order to solve for $x(t_k)$, one should initialize the Newton steps (17.10) using $x(t_{k-1})$. This is what makes interior point methods efficient. Thanks to the use of a logarithmic barrier, one can show the upper bound $f(x(t_k)) - f(x^\star) \leqslant m/t_k$ (m being the number of scalar constraints), so that in order to a final error of ε, one needs to use $k = 0, \ldots, K$ such that

$$\frac{m}{t_K} = \frac{m}{t_0\mu^K} \leqslant \varepsilon.$$

This shows that only $O(|\log(\varepsilon)|)$ steps are necessary to reach ε precision. The important question to bound the complexity of the method is thus to bound the number of Newton steps (17.10). This requires additional hypotheses on f.

If the function f has a so-called self-concordance property, namely that for any (x, y), $\varphi(s) \overset{\text{def.}}{=} f(sx + (1 - s)y)$ satisfies

$$|\varphi'''(s)| \leqslant 2\varphi''(s)^{3/2},$$

one can then that only a constant number of Newton steps are required per iterations (note that $-\log$ being self-concordant, f_t is also self-concordant) when using the warm-restart initialization to computes the succession of $x(t_k)$. This result might looks surprising, but is possible because of the combination of the warm restart strategy with the self-concordance property of f_t: although problems become more and more difficult (f_t is becoming less regular) as t increases, the number of iterations of Newton stays constant. This fundamental result supports the claim that interior point methods solve linear programs (and more general types of problems including SDP problems) in polynomial times (where polynomial refers to polynomial in $\log(\varepsilon)$).

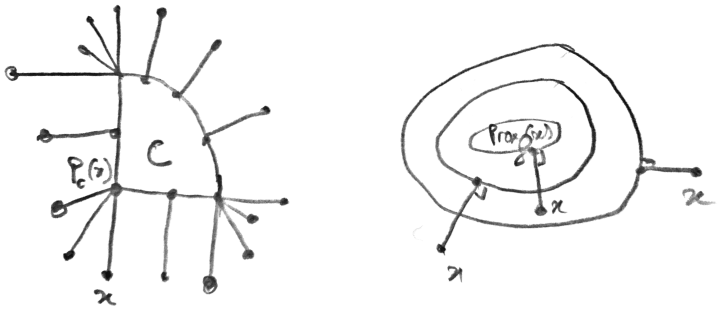

Figure 17.1: Proximal map and projection map.

17.3 Proximal Algorithm

For non-smooth functions f, it is not possible to perform an "explicit" gradient descent step because the gradient is not even defined. One thus needs to replace this "explicit" step by an "implicit" one, which is possible even if f is non-smooth.

17.3.1 Proximal Map

The implicit stepping of amplitude $\tau > 0$ is defined as

$$\forall\, x, \quad \mathrm{Prox}_{\tau f}(x) \overset{\text{def.}}{=} \operatorname*{argmin}_{x'} \frac{1}{2}\|x - x'\|^2 + f(x'). \tag{17.11}$$

It amounts to minimize function f locally around x, in a ball of radius controlled by τ. This the involved function $\frac{1}{2}\|x - \cdot\|^2 + f$ is strongly convex, this operator $\mathrm{Prox}_{\tau f}$ is well defined and single-valued.

When $f = \iota_{\mathcal{C}}$ is an indicator, the proximal map boils down to a projection $\mathrm{Prox}_{\iota_{\mathcal{C}}} = \mathrm{Proj}_{\mathcal{C}}$, it is thus in some sense a generalization of the projection to arbitrary function. And can also be interpreted as a projector on a level set of f. An interesting feature of the proximal map is that it is a contraction, thus generalizing the well-known property of projectors.

Proposition 57. *One has* $\|\mathrm{prox}_f(x) - \mathrm{prox}_f(y)\| \leqslant \|x - y\|$.

Examples The following proposition states a few simples examples.

Proposition 58. *One has*

$$\mathrm{Prox}_{\frac{\tau}{2}\|\cdot\|^2}(x) = \frac{x}{1+\tau}, \quad \textit{and} \quad \mathrm{Prox}_{\tau\|\cdot\|_1} = \mathcal{S}^1_\tau(x), \tag{17.12}$$

where the soft-thresholding is defined as

$$\mathcal{S}^1_\tau(x) \overset{\text{def.}}{=} (S_\tau(x_i))_{i=1}^p \quad \textit{where} \quad S_\tau(r) \overset{\text{def.}}{=} \mathrm{sign}(r)(|r| - \lambda)_+,$$

(see also (9.5)). For $A \in \mathbb{R}^{P\times N}$, one has

$$\mathrm{Prox}_{\frac{\tau}{2}\|A\cdot - y\|^2}(x) = (\mathrm{Id}_N + \tau A^* A)^{-1}(x + \tau A^* y). \tag{17.13}$$

Proof. The proximal map of $\|\cdot\|_1$ was derived in Proposition 26. For the quadratic case

$$z = \mathrm{Prox}_{\frac{\tau}{2}\|A\cdot - y\|^2}(x) \quad \Leftrightarrow \quad z - x + \tau A^*(Az - y) = 0 \quad \Leftrightarrow \quad (\mathrm{Id}_N + \tau A^* A)z = x + \tau A^* y.$$

□

Note that in some case, the proximal map of a non-convex function is well defined, for instance $\mathrm{Prox}_{\tau\|\cdot\|_0}$ is the hard thresholding associated to the threshold $\sqrt{2\tau}$, see Proposition 26.

17.3.2 Basic Properties

We recap some useful proximal-calculus.

Proposition 59. *One has*

$$\mathrm{Prox}_{f+\langle y,\cdot\rangle} = y + \mathrm{Prox}_f, \quad \mathrm{Prox}_{f(\cdot-y)} = y + \mathrm{Prox}_f(\cdot - y).$$

If $f(x) = \sum_{k=1}^K f(x_k)$ *for* $x = (x_1, \ldots, x_K)$ *is separable, then*

$$\mathrm{Prox}_{\tau f}(x) = (\mathrm{Prox}_{\tau f_k}(x_k))_{k=1}^K. \tag{17.14}$$

Proof. One has

$$z = \mathrm{Prox}_{f+\langle y,\cdot\rangle}(x) \quad \Leftrightarrow \quad 0 \in x - z + (\partial f(x) + y) \quad \Leftrightarrow \quad 0 \in x - (z-y) + \partial f(x)$$

which is the optimality condition for $z - y = \mathrm{Prox}_f(x)$.

One has

$$z = \mathrm{Prox}_{f(\cdot-y)}(x) \quad \Leftrightarrow \quad 0 \in x - z + \lambda \partial f(x-y) \quad \Leftrightarrow \quad 0 \in x' - (z-y) + \partial f(x')$$

where we defined $x' \overset{\text{def.}}{=} x - y$, and this is the optimality condition for $z - y = \mathrm{Prox}_f(x')$ □

The following proposition is very useful.

Proposition 60. *If* $A \in \mathbb{R}^{P\times N}$ *is a tight frame, i.e.* $AA^* = \mathrm{Id}_P$*, then*

$$\mathrm{Prox}_{f\circ A} = A^* \circ \mathrm{Prox}_f \circ A + \mathrm{Id}_N - A^*A.$$

In particular, if A *is orthogonal, then* $\mathrm{Prox}_{f\circ A} = A^* \circ \mathrm{Prox}_f \circ A$.

17.3.3 Related Concepts

Link with sub-differential. For a set-valued map $U : \mathcal{H} \hookrightarrow \mathcal{G}$, we define the inverse set-valued map $U^{-1} : \mathcal{G} \hookrightarrow \mathcal{H}$ by

$$h \in U^{-1}(g) \quad \Longleftrightarrow \quad g \in U(h) \tag{17.15}$$

[ToDo: add picture] The following proposition shows that the proximal map is related to a regularized inverse of the sub-differential.

Proposition 61. *One has* $\mathrm{Prox}_{\tau f} = (\mathrm{Id} + \tau\partial f)^{-1}$.

Proof. One has the following equivalence

$$z = \mathrm{Prox}_{\tau f}(x) \Leftrightarrow 0 \in z - x + \tau\partial f(z) \Leftrightarrow x \in (\mathrm{Id} + \tau\partial f)(z) \Leftrightarrow z = (\mathrm{Id} + \tau\partial f)^{-1}(x)$$

where for the last equivalence, we have replace "$\in$" by "$=$" because the proximal map is single valued. □

The proximal operator is hence often referred to the "resolvent" $\mathrm{Prox}_{\tau f} = (\mathrm{Id} + \tau\partial f)^{-1}$ of the maximal monotone operator ∂f.

Link with duality. One has the following fundamental relation between the proximal operator of a function and of its Legendre-Fenchel transform

Theorem 31 (Moreau decomposition)**.** *One has*

$$\mathrm{Prox}_{\tau f} = \mathrm{Id} - \tau\,\mathrm{Prox}_{f^*/\tau}(\cdot/\tau).$$

This theorem shows that the proximal operator of f is simple to compute if and only the proximal operator of f^* is also simple. As a particular instantiation, since according to , one can re-write the soft thresholding as follow

$$\mathrm{Prox}_{\tau\|\cdot\|_1}(x) = x - \tau\,\mathrm{Proj}_{\|\cdot\|_\infty\leqslant 1}(x/\tau) = x - \mathrm{Proj}_{\|\cdot\|_\infty\leqslant\tau}(x) \quad \text{where} \quad \mathrm{Proj}_{\|\cdot\|_\infty\leqslant\tau}(x) = \min(\max(x,-\tau),\tau).$$

In the special case where $f = \iota_{\mathcal{C}}$ where $\mathcal{C}$ is a closed convex cone, then

$$(\iota_{\mathcal{C}})^* = \iota_{\mathcal{C}^\circ} \quad \text{where} \quad \mathcal{C}^\circ \overset{\text{def.}}{=} \{y\,;\,\forall\, x\in\mathcal{C}, \langle x,\,y\rangle \leqslant 0\} \tag{17.16}$$

and $\mathcal{C}^\circ$ is the so-called polar cone. Cones are fundament object in convex optimization because they are invariant by duality, in the sense of (17.16) (if $\mathcal{C}$ is not a cone, its Legendre transform would not be an indicator). Using (17.16), one obtains the celebrated Moreau polar decomposition

$$x = \mathrm{Proj}_{\mathcal{C}}(x) +^{\perp} \mathrm{Proj}_{\mathcal{C}^\circ}(x)$$

where "$+^{\perp}$" denotes an orthogonal sum (the terms in the sum are mutually orthogonal). **[ToDo: add drawing]** In the case where $\mathcal{C} = V$ is a linear space, this corresponds to the usual decomposition $\mathbb{R}^p = V \oplus^{\perp} V^{\perp}$.

Link with Moreau-Yosida regularization. The following proposition shows that the proximal operator can be interpreted as performing a gradient descent step on the Moreau-Yosida smoothed version f_μ of f, defined in (16.11).

Proposition 62. *One has*

$$\mathrm{Prox}_{\mu f} = \mathrm{Id} - \mu\nabla f_\mu.$$

17.4 Primal Algorithms

We now describe some important algorithm which assumes some structure (a so-called "splitting") of the minimized functional to be able to apply proximal maps on sub-functions. Note that there is obviously many ways to split or structure a given initial problem, so there are many non-equivalent ways to apply a given proximal-based method to solve the problem. Finding the "best" way to split a problem is a bit like black magic, and there is no definite answer. Also all there algorithm comes with step size and related parameters, and there is no obvious way to tune these parameters automatically (although some insight might be gained by studying convergence rate).

17.4.1 Proximal Point Algorithm

One has the following equivalence

$$x^\star \in \mathrm{argmin}\, f \quad \Leftrightarrow \quad 0 \in \partial f(x^\star) \quad \Leftrightarrow \quad x^\star \in (\mathrm{Id} + \tau\partial f)(x^\star) \tag{17.17}$$

$$\Leftrightarrow \quad x^\star = (\mathrm{Id} + \tau\partial f)^{-1}(x^\star) = \mathrm{Prox}_{\tau f}(x^\star). \tag{17.18}$$

This shows that being a minimizer of f is equivalent to being a fixed point of $\mathrm{Prox}_{\tau f}$. This suggest the following fixed point iterations, which are called the proximal point algorithm

$$x^{(\ell+1)} \overset{\text{def.}}{=} \mathrm{Prox}_{\tau_\ell f}(x^{(\ell)}). \tag{17.19}$$

On contrast to the gradient descent fixed point scheme, the proximal point method is converging for any sequence of steps.

Theorem 32. *If $0 < \tau_{\min} \leqslant \tau_\ell \leqslant \gamma_{\max} < +\infty$, then $x^{(\ell)} \to x^\star$ a minimizer of f.*

This implicit step (17.19) should be compared with a gradient descent step (17.2)

$$x^{(\ell+1)} \stackrel{\text{def.}}{=} (\mathrm{Id} + \tau_\ell \nabla f)(x^{(\ell)}).$$

One sees that the implicit resolvent $(\mathrm{Id} - \tau_\ell \partial f)^{-1}$ replaces the explicit step $\mathrm{Id} + \tau_\ell \nabla f$. For small τ_ℓ and smooth f, they are equivalent at first order. But the implicit step is well defined even for non-smooth function, and the scheme (the proximal point) is always convergent (whereas the explicit step size should be small enough for the gradient descent to converge). This is inline with the general idea the implicit stepping (e.g. implicit Euler for integrating ODE, which is very similar to the proximal point method) is more stable. Of course, the drawback is that explicit step are very easy to implement whereas in general proximal map are hard to solve (most of the time as hard as solving the initial problem).

17.4.2 Forward-Backward

It is in general impossible to compute $\mathrm{Prox}_{\gamma f}$ so that the proximal point algorithm is not implementable. In oder to derive more practical algorithms, it is important to restrict the class of considered function, by imposing some structure on the function to be minimized. We consider functions of the form

$$\min_x \mathcal{E}(x) \stackrel{\text{def.}}{=} f(x) + g(x) \tag{17.20}$$

where $g \in \Gamma_0(\mathcal{H})$ can be an arbitrary, but f needs to be smooth.

One can modify the fixe point derivation (17.17) to account for this special structure

$$\begin{aligned} x^\star \in \operatorname{argmin} f + g \quad &\Leftrightarrow \quad 0 \in \nabla f(x^\star) + \partial g(x^\star) \quad \Leftrightarrow \quad x^\star - \tau \nabla f(x^\star) \in (\mathrm{Id} + \tau \partial g)(x^\star) \\ &\Leftrightarrow \quad x^\star = (\mathrm{Id} + \tau \partial g)^{-1} \circ (\mathrm{Id} - \tau \nabla f)(x^\star). \end{aligned}$$

This fixed point suggests the following algorithm, with the celebrated Forward-Backward

$$x^{(\ell+1)} \stackrel{\text{def.}}{=} \mathrm{Prox}_{\tau_\ell g}\left(x^{(\ell)} - \tau_\ell \nabla f(x^{(\ell)})\right). \tag{17.21}$$

Derivation using surrogate functionals. An intuitive way to derive this algorithm, and also a way to prove its convergence, it using the concept of surrogate functional.

To derive an iterative algorithm, we modify the energy $\mathcal{E}(x)$ to obtain a surrogate functional $\mathcal{E}(x, x^{(\ell)})$ whose minimization corresponds to a simpler optimization problem, and define the iterations as

$$x^{(\ell+1)} \stackrel{\text{def.}}{=} \operatorname*{argmin}_x \mathcal{E}(x, x^{(\ell)}). \tag{17.22}$$

In order to ensure convergence, this function should satisfy the following property

$$\mathcal{E}(x) \leqslant \mathcal{E}(x, x') \quad \text{and} \quad \mathcal{E}(x, x) = \mathcal{E}(x) \tag{17.23}$$

and $\mathcal{E}(x) - \mathcal{E}(x, x')$ should be a smooth function. Property (17.23) guarantees that f is decaying by the iterations

$$\mathcal{E}(x^{(\ell+1)}) \leqslant \mathcal{E}(x^{(\ell)})$$

and it simple to check that actually all accumulation points of $(x^{(\ell)})_\ell$ are stationary points of f.

In order to derive a valid surrogate $\mathcal{E}(x, x')$ for our functional (17.20), since we assume f is L-smooth (i.e. satisfies $(\mathcal{R}_L)$), let us recall the quadratic majorant (13.19)

$$f(x) \leqslant f(x') + \langle \nabla f(x'), x' - x \rangle + \frac{L}{2}\|x - x'\|^2,$$

so that for $0 < \tau < \frac{1}{L}$, the function

$$\mathcal{E}(x, x') \stackrel{\text{def.}}{=} f(x') + \langle \nabla f(x'), x' - x \rangle + \frac{1}{2\tau}\|x - x'\|^2 + g(x) \tag{17.24}$$

satisfies the surrogate conditions (17.23). The following proposition shows that minimizing the surrogate functional corresponds to the computation of a so-called proximal operator.

Proposition 63. *The update* (17.22) *for the surrogate* (17.24) *is exactly* (17.21).

Proof. This follows from the fact that

$$\langle \nabla f(x'), x' - x \rangle + \frac{1}{2\tau}\|x - x'\|^2 = \frac{1}{2\tau}\|x - (x' - \tau \nabla f(x'))\|^2 + \text{cst}.$$

□

Convergence of FB. Although we impose $\tau < 1/L$ to ensure majorization property, one can actually show convergence under the same hypothesis as for the gradient descent, i.e. $0 < \tau < 2/L$, with the same convergence rates. This means that Theorem 30 for the projected gradient descent extend to FB.

Theorem 33. *Theorems* **??** *and 23 still holds when replacing iterations* (17.2) *by* (17.21).

Note furthermore that the projected gradient descent algorithm (17.5) is recovered as a special case of (17.21) when setting $J = \iota_{\mathcal{C}}$ the indicator of the constraint set, since $\text{Prox}_{\rho J} = \text{Proj}_{\mathcal{C}}$ in this case.

Of course the difficult point is to be able to compute in closed form $\text{Prox}_{\tau g}$ in (17.21), and this is usually possible only for very simple function. We have already seen such an example in Section 14.1.4 for the resolution of ℓ^1-regularized inverse problems (the Lasso).

17.4.3 Douglas-Rachford

We consider here the structured minimization problem

$$\min_{x \in \mathbb{R}^p} f(x) + g(x), \tag{17.25}$$

but on contrary to the Forward-Backward setting studied in Section 17.4.2, no smoothness is imposed on f. We here suppose that we can compute easily the proximal map of f and g.

Example 5 (Constrained Lasso). An example of a problem of the form (17.25) where one can apply Douglas-Rachford is the noiseless constrained Lasso problem (9.11)

$$\min_{Ax = y} \|x\|_1$$

where one can use $f = \iota_{\mathcal{C}_y}$ where $\mathcal{C}_y \stackrel{\text{def.}}{=} \{x \,;\, Ax = y\}$ and $g = \|\cdot\|_1$. As noted in Section 9.3.1, this problem is equivalent to a linear program. The proximal operator of g is the soft thresholding as stated in (17.12), while the proximal operator of g is the orthogonal projector on the affine space $\mathcal{C}_y$, which can be computed by solving a linear system as stated in (16.9) (this is especially convenient for inpainting problems or deconvolution problem where this is achieved efficiently).

The Douglas-Rachford iterations read

$$\tilde{x}^{(\ell+1)} \stackrel{\text{def.}}{=} \left(1 - \frac{\mu}{2}\right)\tilde{x}^{(\ell)} + \frac{\mu}{2}\,\text{rProx}_{\tau g}(\text{rProx}_{\tau f}(\tilde{x}^{(\ell)})) \quad \text{and} \quad x^{(\ell+1)} \stackrel{\text{def.}}{=} \text{Prox}_{\tau f}(\tilde{x}^{(\ell+1)}), \tag{17.26}$$

where we have used the following shortcuts

$$\text{rProx}_{\tau f}(x) = 2\,\text{Prox}_{\tau f}(x) - x.$$

One can show that for any value of $\tau > 0$, any $0 < \mu < 2$, and any $\tilde{x}_0$, $x^{(\ell)} \to x^\star$ which is a minimizer of $f + g$.

Note that it is of course possible to inter-change the roles of f and g, which defines another set of iterations.

More than two functions. Another sets of iterations can be obtained by "symetrizing" the algorithm. More generally, if we have K functions $(f_k)_k$, we re-write

$$\min_x \sum_k f_k(x) = \min_{X=(x_1,\ldots,x_k)} f(X)+g(X) \quad \text{where} \quad f(X)=\sum_k f_k(x_k) \quad \text{and} \quad g(X)=\iota_\Delta(X)$$

where $\Delta=\{X\ ;\ x_1=\ldots=x_k\}$ is the diagonal. The proximal operator of f is

$$\mathrm{Prox}_{\tau f}(X)=\mathrm{Proj}_\Delta(X)=(\bar{x},\ldots,\bar{x}) \quad \text{where} \quad \bar{x}=\frac{1}{K}\sum_k x_k$$

while the proximal operator of f is easily computed from those of the $(f_k)_k$ using (17.14). One can thus apply DR iterations (17.26).

Handling a linear operator. One can handle a minimization of the form (17.28) by introducing extra variables

$$\inf_x f_1(x)+f_2(Ax)=\inf_{z=(x,y)} f(z)+g(z) \quad \text{where} \quad \left\{\begin{array}{l} f(z)=f_1(x)+f_2(y) \\ g(z)=\iota_{\mathcal{C}}(x,y), \end{array}\right.$$

where $\mathcal{C}=\{(x,y)\ ;\ Ax=y\}$. This problem can be handled using DR iterations (17.26), since the proximal operator of f is obtained from those of (f_1,f_2) using (17.14), while the proximal operator of g is the projector on $\mathcal{C}$, which can be computed in two alternative ways as the following proposition shows.

Proposition 64. *One has*

$$\mathrm{Proj}_{\mathcal{C}}(x,y)=(x+A^*\tilde{y},y-\tilde{y})=(\tilde{x},A\tilde{x}) \quad \textit{where} \quad \left\{\begin{array}{l} \tilde{y}\overset{\text{def.}}{=}(\mathrm{Id}_P+AA^*)^{-1}(Ax-y) \\ \tilde{x}\overset{\text{def.}}{=}(\mathrm{Id}_N+A^*A)^{-1}(A^*y+x). \end{array}\right. \tag{17.27}$$

Proof. **[ToDo: todo]** □

Remark 10 (Inversion of linear operator). At many places (typically to compute some sort of projector) one has to invert matrices of the form AA^*, A^*A, Id_P+AA^* or Id_N+A^*A (see for instance (17.27)). There are some case where this can be done efficiently. Typical examples where this is simple are inpainting inverse problem where AA^* is diagonal, and deconvolution or partial Fourier measurement (e.g. fMRI) for which A^*A is diagonalized using the FFT. If this inversion is too costly, one needs to use more advanced methods, based on duality, which allows to avoid trading the inverse A by the application of A^*. They are however typically converging more slowly.

17.5 Dual and Primal-Dual Algorithms

Convex duality, detailed in Section 16.2 (either from the Lagrange or the Fenchel-Rockafellar point of view – which are essentially equivalent), is very fruitful to derive new optimization algorithm or to apply existing algorithm on a dual reformulation.

17.5.1 Forward-backward on the Dual

Let us illustrate first the idea of applying a known algorithm to the dual problem. We consider here the structured minimization problem associated to Fenchel-Rockafellar duality (16.12)

$$p^\star=\inf_x f(x)+g(Ax), \tag{17.28}$$

but furthermore assume that f is μ-strongly convex, and we assume for simplicity that both (f,g) are continuous. If f were also smooth (but it needs to be!), one could think about using the Forward-Backward algorithm (17.21). But the main issue is that in general $\mathrm{Prox}_{\tau g\circ A}$ cannot be computed easily even if one can compute $\mathrm{Prox}_{\tau g\circ A}$. An exception to this is when A is a tight frame, as exposed in Proposition 60, but in practice it is rarely the case.

Example 6 (TV denoising). A typical example, which was the one used by Antonin Chambolle [6] to develop this class of method, is the total variation denoising

$$\min_x \frac{1}{2}\|y-x\|^2 + \lambda\|\nabla x\|_{1,2} \tag{17.29}$$

where $\nabla x \in \mathbb{R}^{N\times d}$ is the gradient (a vector field) of a signal ($d=1$)or image ($d=2$) x, and $\|\cdot\|_{1,2}$ is the vectorial-ℓ^1 norm (also called $\ell^1-\ell^2$ norm), such that for a d-dimensional vector field $(v_i)_{i=1}^p$

$$\|v\|_{1,2} \overset{\text{def.}}{=} \sum_i \|v_i\|.$$

Here

$$f = \frac{1}{2}\|\cdot - y\|^2 \quad \text{and} \quad g = \lambda\|\cdot\|_{1,2}$$

so that f is $\mu=1$ strongly convex, and one sets $A=\nabla$ the linear operator.

Example 7 (Support Vector Machine). We consider a supervised classification problems with data $(a_i \in \mathbb{R}^p, y_i \in \{-1,+1\})_{i=1}^n$. The support vector machine classification method, with a ridge penalty $\|x\|^2$ reads

$$\min_{x\in\mathbb{R}^p} \ell(-y_i\langle x,\, a_i\rangle) + \frac{\lambda}{2}\|x\|^2 = \frac{\lambda}{2}\|x\|^2 + g(Ax) \tag{17.30}$$

where $\ell(s)=\max(0,s+1)$ is the hinge loss function, where we denoted $A_{i,j}=-y_i a_i[j]$ and $L(u)=\sum_i \ell(u_i)$. It corresponds to the split (17.28) using $f(x)=\frac{\lambda}{2}\|x\|^2$.

Applying Fenchel-Rockafellar Theorem 28 (since strong duality holds, all involved functions being continuous), one has that

$$p^\star = \sup_u \; -g^*(u) - f^*(-A^*u). \tag{17.31}$$

But more importantly, since f is μ-strongly convex, one has that f^* is smooth with a $1/\mu$-Lipschitz gradient. One can thus use the Forward-Backward algorithm (17.21) on (minus the energy of) this problem, which reads

$$u^{(\ell+1)} = \mathrm{Prox}_{\tau_k g^*}\left(u^{(\ell)} + \tau_k A\nabla f^*(A^*u^{(\ell)})\right).$$

To guarantee convergence, the step size τ_k should be smaller than $2/L$ where L is the Lipschitz constant of $A\circ\nabla f^*\circ A^*$, which is smaller than $\|A\|^2/\mu$.

Last but not least, one some (not necessarily unique) dual minimizer $u^\star$ is computed, the primal-dual relationships (16.18) ensures that one retrieves the unique primal minimizer $x^\star$ as

$$-A^*u^\star \in \partial f(x^\star) \quad \Leftrightarrow \quad x^\star \in (\partial f)^{-1}(-A^*u^\star) = \partial f^*(-A^*u^\star) \quad \Leftrightarrow \quad x^\star = \nabla f^*(-A^*u^\star)$$

where we used here the crucial fact that f^* is smooth.

Example 8 (TV denoising). In the particular case of the TV denoising problem (17.29), one has

$$g^* = \iota_{\|\cdot\|_{\infty,2}\leqslant\lambda} \quad \text{where} \quad \|v\|_{\infty,2} \overset{\text{def.}}{=} \max_i \|v_i\| \quad \Longrightarrow$$

$$f^\star(h) = \frac{1}{2}\|h\|^2 + \langle h,\, y\rangle,$$

so that the dual problem (re-written as a min) reads

$$\min_{\|v\|_{\infty,2}\leqslant\lambda} \frac{1}{2}\|-\nabla^* v + y\|^2.$$

This can be minimized by projected gradient descent (special case of FB) using

$$\mathrm{Prox}_{\tau g^*}(u) = \left(\min(\|v_i\|,\lambda)\frac{v_i}{\|v_i\|}\right)_i \quad \text{and} \quad \nabla f^\star(h) = h + y.$$

Furthermore, one has $\mu=1$ and $A^*A=\Delta$ is the usual finite difference approximation of the Laplacian, so that $\|A\|^2 = \|\Delta\| = 4d$ where d is the dimension.

Example 9 (Support Vector Machine). For the SVM problem (17.30), one has

$$\forall\, u \in \mathbb{R}^n, \quad g^\star(u) = \sum_i \ell^\star(u_i)$$

where

$$\ell^\star(t) = \sup_s ts - \max(0, s+1) = -t + \sup_u ts - \max(0, s) = -t + \iota_{[0,1]}(t)$$

where we used the fact that the Legendre transform of $\max(0,s)$ is the indicator of the sub-differential at 0, which is $[0,1]$ **[ToDo: drawing]**. Also, one has $f(x) = \frac{\lambda}{2}\|x\|^2$ so that

$$f^\star(z) = \lambda(\|\cdot\|^2/2)(z/\lambda) = \lambda \frac{\|z/\lambda\|^2}{2} = \frac{\|z\|^2}{2\lambda}.$$

The dual problem reads

$$\min_{0 \leqslant v \leqslant 1} -\langle \mathbb{1}, v\rangle + \frac{\|A^\top v\|^2}{2\lambda}$$

which can be solved using forward backward, in this case being equivalent to gradient descent. One thus has

$$\mathrm{Prox}_{\tau g^*}(u) = \min_{v \in [0,1]^n} \frac{1}{2}\|u - v\|^2 - \langle \mathbb{1}, v\rangle = \mathrm{Proj}_{[0,1]^n}(u - \mathbb{1}) = \min(\max(u - \mathbb{1}, 0), \mathbb{1}).$$

17.5.2 Alternating Direction Method of Multipliers

In order to minimize the general splitting (17.28), of the form $f + g \circ A$, in the absence of strong convexity property of f, one can use Douglas-Rachford splitting presented in Section 17.4.3. This requires to be able to compute $\mathrm{Prox}_{\tau f}$ and $\mathrm{Prox}_{\tau g \circ A}$. The Alternating Direction Method of Multipliers (ADMM) algorithm takes a different route, and rather assume one is able to compute, in addition to $\mathrm{Prox}_{\tau g^*}$ (thus being equivalent to the computation of $\mathrm{Prox}_{\tau g}$) and $\mathrm{Prox}_{\tau f^* \circ A^*}$. In fact, it can be shown that ADMM is equivalent to applying DR on the dual problem (17.31), which is the reason why it requires the use of the proximal map of the dual functions.

For $A \in \mathbb{R}^{n \times p}$, we consider the problem derived in (16.13) using Lagrange duality and an extra scaling by $\gamma > 0$

$$\inf_{x \in \mathbb{R}^p} f(x) + g(Ax) = \inf_{y = Ax} f(x) + g(y) = \inf_{x \in \mathbb{R}^p, y \in \mathbb{R}^n} \sup_{z \in \mathbb{R}^n} \mathcal{L}((x,y),z) \overset{\text{def.}}{=} f(x) + g(y) + \gamma\langle z, Ax - y\rangle. \quad (17.32)$$

The augmented Lagrangian corresponds to solving the problem

$$\inf_{x \in \mathbb{R}^p, y \in \mathbb{R}^n} \sup_{z \in \mathbb{R}^n} \mathcal{L}_\gamma((x,y),z) \overset{\text{def.}}{=} \mathcal{L}((x,y),z) + \frac{\gamma}{2}\|Ax - y\|^2 = f(x) + g(y) + \gamma\langle z, Ax - y\rangle + \frac{\gamma}{2}\|Ax - y\|^2.$$

which is equivalent to (17.32) in the sense that they have the same solutions because the additional term $\|Ax - y\|^2$ is zero at the solutions.

The ADMM method then updates

$$y^{(\ell+1)} \overset{\text{def.}}{=} \operatorname*{argmin}_y \mathcal{L}_\gamma((x^{(\ell)}, y), z^{(\ell)}), \quad (17.33)$$

$$x^{(\ell+1)} \overset{\text{def.}}{=} \operatorname*{argmin}_x \mathcal{L}_\gamma((x, y^{(\ell+1)}), z^{(\ell)}), \quad (17.34)$$

$$z^{(\ell+1)} \overset{\text{def.}}{=} z^{(\ell)} + Ax^{(\ell+1)} - y^{(\ell+1)}. \quad (17.35)$$

This is very similar to the dual ascent method, which perform a gradient descent on the dual (which a well chosen step size), which can be shown to be the iterations

$$(x^{(\ell+1)}, y^{(\ell+1)}) \overset{\text{def.}}{=} \operatorname*{argmin}_{x,y} \mathcal{L}_\gamma((x,y), z^{(\ell)}),$$
$$z^{(\ell+1)} \overset{\text{def.}}{=} z^{(\ell)} + Ax^{(\ell+1)} - y^{(\ell+1)},$$

excepted that for ADMM the x and y variable are updated alternatively (hence the naming), which in many situation gives tractable steps.

Step (17.35) can be understood as a gradient ascent on $\mathcal{L}_\gamma$ on the z variable, using a specific step size of $1/\gamma$, so that the primal-dual relation $\nabla f(x^\star) + \gamma A^\top z^\star = 0$ (which are the optimality condition with respect to x, assuming $Ax^\star = y^\star$, here we assume f is smooth for simplicity) is imposed at the end of the z update. Indeed, the optimality condition for (17.34) reads

$$0 = \nabla_x \mathcal{L}_\gamma((x^{(\ell+1)}, y^{(\ell+1)}), z^{(\ell)}) = \nabla f(x^{(\ell+1)}) + \gamma A^\top z^{(\ell)} + A^\top(Ax^{(\ell+1)} - y^{(\ell+1)}) = \nabla f(x^{(\ell+1)}) + \gamma A^\top z^{(\ell+1)}.$$

Step (17.33) is a proximal step, since

$$\begin{aligned} y^{(\ell+1)} &= \operatorname*{argmin}_y g(y) + \gamma\langle z^{(\ell)}, Ax^{(\ell)} - y\rangle + \frac{\gamma}{2}\|Ax^{(\ell)} - y\|^2 \\ &= \operatorname*{argmin}_y \frac{1}{2}\|y - Ax^{(\ell)} - z^{(\ell)}\|^2 + \frac{1}{\gamma} g(y) = \operatorname{Prox}_{g/\gamma}(Ax^{(\ell)} + z^{(\ell)}). \end{aligned}$$

Step (17.34) is more involved, since it is a proximal step for a metric twisted by A

$$\begin{aligned} x^{(\ell+1)} &= \operatorname*{argmin}_x f(x) + \gamma\langle z^{(\ell)}, Ax - y^{(\ell+1)}\rangle + \frac{\gamma}{2}\|Ax - y^{(\ell+1)}\|^2 \\ &= \operatorname*{argmin}_x \frac{1}{2}\|Ax - y^{(\ell+1)} + z^{(\ell)}\|^2 \frac{1}{\gamma} f(x) \overset{\text{def.}}{=} \operatorname{Prox}^A_{f/\gamma}(z^{(\ell)} - y^{(\ell+1)}). \end{aligned}$$

The following formula, which can be shown after some tedious computation, shows that actually computing this twisted proximal map is equivalent to the computation of the proximal operator of $f^* \circ A^*$

$$\operatorname{Prox}^A_{f/\gamma}(u) = A^+\left(u - \frac{1}{\gamma}\operatorname{Prox}_{\gamma f^* \circ A^*}(\gamma u)\right).$$

Example 10 (Constrained TV recovery). We consider the problem of minimizing a TV norm under affine constraints

$$\min_{Bx=y} \|\nabla x\|_1$$

which is of the form $f + g \circ A$ when defining $A = \nabla$, $g = \|\cdot\|_1$ and $f = \iota_{B\cdot=y}$. The proximal map of g is the soft thresholding, while the "twisted" proximal operator is

$$\operatorname{Prox}^A_{f/\gamma}(u) = \operatorname*{argmin}_{Bx=y} \frac{1}{2}\|Ax - u\|^2 = \operatorname*{argmin}_{Bx=y} \min_w \frac{1}{2}\|\nabla x - u\|^2 + \langle w, Bx - y\rangle$$

where we added Lagrange multipliers w for the constraint $Bx = y$. From the first order conditions, an optimal pair $(x^\star, w^\star)$ is obtained by solving a linear system

$$\begin{pmatrix} \nabla^\top\nabla & B^\top \\ B & 0 \end{pmatrix}\begin{pmatrix} x^\star \\ w^\star \end{pmatrix} = \begin{pmatrix} \nabla^\top u \\ y \end{pmatrix}$$

and note that $\Delta = -\nabla^\top\nabla$ is usually called the (discrete) Laplacian.

17.5.3 Primal-Dual Splitting

In the case where one cannot compute neither $\operatorname{Prox}_{\tau g\circ A}$ or $\operatorname{Prox}_{\tau f^*\circ A^*}$, then one needs to use more general proximal methods, which directly operate over the primal-dual saddle point. They are more general, but in general less efficient than purely primal or dual methods. We thus consider the general structure problem of the form (17.28), which we intend to solve in the primal-dual, using the fact that $g = (g^*)^*$ (otherwise one can re-use the approach using Lagrange duality performed in (16.13)) form as

$$\inf_x f(x) + g(Ax) = \inf_x f(x) + \sup_u \langle Ax, u\rangle - g^*() = \sup_u \inf_x f(x) + \langle Ax, u\rangle - g^*(u). \tag{17.36}$$

We make no assumption such as strong convexity of f.

Example 11 (Total Variation regularization of inverse problems). A typical instance of such a problem is for the TV regularization of the inverse problem $\mathcal{K}x = y$, which corresponds to solving

$$\min_x \frac{1}{2}\|y - \mathcal{K}x\|^2 + \lambda\|\nabla x\|_{1,2}.$$

where $A = \nabla$, $f(x) = \frac{1}{2}\|y - \mathcal{K}\cdot\|^2$ and $g = \lambda\|\cdot\|_{1,2}$. If one would be using directly $f \circ A$ (for instance in combination with DR splitting), computing its proximal operator of f, which, following (17.13), requires inverting either $\mathrm{Id}_P + AA^*$ or $\mathrm{Id}_N + A^*A$, see Remark 10. The goal is to avoid this here.

A standard primal-dual algorithm, which is detailed in [], reads

$$\begin{aligned} z^{(\ell+1)} &\overset{\text{def.}}{=} \mathrm{Prox}_{\sigma g^*}(z^{(\ell)} + \sigma A(\tilde{x}^{(\ell)}) \\ x^{(\ell+1)} &\overset{\text{def.}}{=} \mathrm{Prox}_{\tau f}(x^{(\ell)} - \tau A^*(z^{(\ell+1)})) \\ \tilde{x}^{(\ell)} &\overset{\text{def.}}{=} x^{(\ell+1)} + \theta(x^{(\ell+1)} - x^{(\ell)}) \end{aligned}$$

if $0 \leqslant \theta \leqslant 1$ and $\sigma\tau\|K\|^2 < 1$, then $x^{(\ell)}$ converges to a minimizer of (17.36) .

Bibliography

[1] Amir Beck. *Introduction to Nonlinear Optimization: Theory, Algorithms, and Applications with MATLAB*. SIAM, 2014.

[2] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[3] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[4] E. Candès and D. Donoho. New tight frames of curvelets and optimal representations of objects with piecewise C^2 singularities. *Commun. on Pure and Appl. Math.*, 57(2):219–266, 2004.

[5] E. J. Candès, L. Demanet, D. L. Donoho, and L. Ying. Fast discrete curvelet transforms. *SIAM Multiscale Modeling and Simulation*, 5:861–899, 2005.

[6] A. Chambolle. An algorithm for total variation minimization and applications. *J. Math. Imaging Vis.*, 20:89–97, 2004.

[7] Antonin Chambolle, Vicent Caselles, Daniel Cremers, Matteo Novaga, and Thomas Pock. An introduction to total variation for image analysis. *Theoretical foundations and numerical methods for sparse recovery*, 9(263-340):227, 2010.

[8] Antonin Chambolle and Thomas Pock. An introduction to continuous optimization for imaging. *Acta Numerica*, 25:161–319, 2016.

[9] S.S. Chen, D.L. Donoho, and M.A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1999.

[10] Philippe G Ciarlet. Introduction à l'analyse numérique matricielle et à l'optimisation. 1982.

[11] P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *SIAM Multiscale Modeling and Simulation*, 4(4), 2005.

[12] I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Commun. on Pure and Appl. Math.*, 57:1413–1541, 2004.

[13] D. Donoho and I. Johnstone. Ideal spatial adaptation via wavelet shrinkage. *Biometrika*, 81:425–455, Dec 1994.

[14] Heinz Werner Engl, Martin Hanke, and Andreas Neubauer. *Regularization of inverse problems*, volume 375. Springer Science & Business Media, 1996.

[15] M. Figueiredo and R. Nowak. An EM Algorithm for Wavelet-Based Image Restoration. *IEEE Trans. Image Proc.*, 12(8):906–916, 2003.

[16] Simon Foucart and Holger Rauhut. *A mathematical introduction to compressive sensing*, volume 1. Birkhäuser Basel, 2013.

[17] Stephane Mallat. *A wavelet tour of signal processing: the sparse way*. Academic press, 2008.

[18] D. Mumford and J. Shah. Optimal approximation by piecewise smooth functions and associated variational problems. *Commun. on Pure and Appl. Math.*, 42:577–685, 1989.

[19] Neal Parikh, Stephen Boyd, et al. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.

[20] Gabriel Peyré. *L'algèbre discrète de la transformée de Fourier*. Ellipses, 2004.

[21] J. Portilla, V. Strela, M.J. Wainwright, and Simoncelli E.P. Image denoising using scale mixtures of Gaussians in the wavelet domain. *IEEE Trans. Image Proc.*, 12(11):1338–1351, November 2003.

[22] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D*, 60(1-4):259–268, 1992.

[23] Otmar Scherzer, Markus Grasmair, Harald Grossauer, Markus Haltmeier, Frank Lenzen, and L Sirovich. *Variational methods in imaging*. Springer, 2009.

[24] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.

[25] Jean-Luc Starck, Fionn Murtagh, and Jalal Fadili. *Sparse image and signal processing: Wavelets and related geometric multiscale analysis*. Cambridge university press, 2015.