Mathematical Foundations of Data Sciences

Gabriel Peyré
CNRS & DMA, École Normale Supérieure

CHAPTER 19

Nonsmooth Convex Optimization

September 9, 2026

Many recovery and learning problems combine smooth losses with constraints or penalties that are not differentiable. Solving them efficiently requires understanding both the cost of an iteration and the conditions for convergence. We begin with subgradient, projection, and barrier methods, then develop proximal maps and splitting algorithms, including ADMM and primal–dual schemes.

The main references for this chapter are [5, 6, 3]. Further background is provided in [7, 2, 1].

We consider a general convex optimization problem

$$\min_{x \in \mathcal{H}} f(x) \tag{19.1}$$

where $\mathcal{H} = \mathbb{R}^p$ is a finite-dimensional Euclidean space, and seek algorithms with a low cost per iteration. The first-order methods below use gradients, subgradients, or proximal maps. Unless stated otherwise, functions are proper, lower semicontinuous, and convex, and a minimizer is assumed to exist.

19.1 Descent Methods

We extend the gradient method of Section 14.4 to nonsmooth objectives and constraints.

19.1.1 Gradient Descent

The optimization problem (9.26) is an unconstrained problem of the form (19.1) with a smooth objective: $f : \mathcal{H} \to \mathbb{R}$ is continuously differentiable with Lipschitz gradient. Here $\nabla f(x) \in \mathcal{H}$ is the gradient of the objective with respect to its variable x. It is distinct from the spatial gradient ∇x of a signal or image, which is a vector field. The functional gradient satisfies the first-order expansion

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

The Lipschitz-gradient assumption gives the quadratic remainder $O(\|r\|_{\mathcal{H}}^2)$; differentiability alone gives $o(\|r\|_{\mathcal{H}})$. Examples of gradient computations appear in Section 9.5.3.

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

$$x^{(\ell+1)} \overset{\text{def.}}{=} x^{(\ell)} - \tau_\ell \nabla f(x^{(\ell)}). \tag{19.2}$$

The step size $\tau_\ell > 0$ balances stability against progress per iteration.

19.1.2 Subgradient Method

For a nonsmooth objective f, replace the gradient in (19.2) by a subgradient:

$$x^{(\ell+1)} \overset{\text{def.}}{=} x^{(\ell)} - \tau_\ell g^{(\ell)} \quad \text{where} \quad g^{(\ell)} \in \partial f(x^{(\ell)}). \tag{19.3}$$

A subgradient step need not decrease the objective. Convergence generally requires diminishing step sizes, as the example $f(x) = |x|$ illustrates. This can make the method slow. When the objective admits a useful decomposition, the proximal methods developed below can exploit it more effectively.

Theorem 19.1. *Assume f is convex, a minimizer $x^\star$ exists, and the chosen subgradients satisfy $\|g^\ell\| \leqslant G$. If $\sum_\ell \tau_\ell = +\infty$ and $\sum_\ell \tau_\ell^2 < +\infty$, then the iterates converge to a minimizer. More generally,*

$$\min_{0\leqslant \ell < N} \{f(x^\ell) - f(x^\star)\} \leqslant \frac{\|x^0 - x^\star\|^2 + G^2 \sum_{\ell<N} \tau_\ell^2}{2\sum_{\ell<N} \tau_\ell}.$$

For a prescribed horizon N and positive bounds D, G, the constant step $\tau_\ell = D/(G\sqrt{N})$ gives a bound $DG/\sqrt{N}$ when $\|x^0 - x^\star\| \leqslant D$. A step $1/(\ell+1)$ satisfies the convergence conditions but does not, by itself, give an $O(N^{-1/2})$ bound.

19.1.3 Projected Gradient Descent

We consider a constrained optimization problem

$$\min_{x\in C} f(x) \tag{19.4}$$

where $C \subset \mathbb{R}^S$ is a nonempty closed convex set and $f : \mathbb{R}^S \to \mathbb{R}$ is convex and continuously differentiable. The convergence theorem below additionally assumes a Lipschitz gradient.

To impose the constraint, follow the gradient step (19.2) by projection:

$$x^{(\ell+1)} \overset{\text{def.}}{=} \mathrm{Proj}_C\left(x^{(\ell)} - \tau_\ell \nabla f(x^{(\ell)})\right). \tag{19.5}$$

Here Proj_C is the orthogonal projection onto C, defined by

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

The projection is unique because C is nonempty, closed, and convex. Projected gradient retains analogous convergence guarantees under the following assumptions.

Theorem 19.2. *Assume f is convex with L-Lipschitz gradient, C is nonempty, closed, and convex, and a constrained minimizer exists. For a fixed $0 < \tau < 2/L$, projected gradient descent converges to a constrained minimizer. If $0 < \tau \leqslant 1/L$, then for $\ell \geqslant 1$,*

$$f(x^\ell) - f(x^\star) \leqslant \frac{\|x^0 - x^\star\|^2}{2\tau\ell}.$$

If f is μ-strongly convex, the iterates converge linearly.

Proof. This is the special case $g = \iota_C$ of Theorem 19.14. In the strongly convex case, the gradient step is a contraction for the admissible fixed step sizes, and composition with the nonexpansive projector preserves that property. □

The main practical difficulty in (19.5) is evaluating the projection. For several sets arising in ℓ^1 minimization, it can be computed efficiently.

19.2 Interior Point Methods

We briefly describe interior-point methods, which often require relatively few but expensive iterations. They extend Newton-type methods to problems whose constraints have a suitable barrier representation, such as nonnegativity of vectors or positive definiteness of matrices.

To illustrate the main idea, we consider the following problem

$$\min_{x\in\mathbb{R}^d, Ax\leqslant y} f(x) \tag{19.6}$$

for $A \in \mathbb{R}^{m\times d}$. Related formulations replace the vector Ax by a matrix and the componentwise inequality $\leqslant$ by a positive-semidefinite constraint.

Example 19.3 (Lasso). The Lasso problem (applied to the regression problem $Bw \approx b$ for $B \in \mathbb{R}^{n\times p}$)

$$\min_{w\in\mathbb{R}^p} \frac{1}{2}\|Bw - b\|^2 + \lambda\|w\|_1 \tag{19.7}$$

with $\lambda > 0$ can be recast as (19.6) by introducing nonnegative variables $x = (x_-, x_+) \in \mathbb{R}^{2p}$ (so that $d = 2p$) and $w = x_+ - x_-$ with $(x_+, x_-) \geqslant 0$, so that $f(x) = \frac{1}{2}\|B(x_+ - x_-) - b\|^2 + \lambda\langle x, \mathbb{1}\rangle$, $A = -\mathrm{Id}_{2p}$, and $y = 0$ (so that $m = 2p$).

Example 19.4 (Dual of the Lasso). Alternatively, solve the dual of (19.7):

$$\min_{\|B^\top q\|_\infty \leqslant 1} f(q) = \frac{\lambda}{2}\|q\|^2 - \langle q,\, b\rangle \tag{19.8}$$

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

Interior-point methods approximate (19.6) by introducing a logarithmic barrier:

$$\min_{x\in\mathbb{R}^d,\; Ax<y} f_t(x) \overset{\text{def.}}{=} f(x) - \frac{1}{t}\mathrm{Log}(y - Ax) \tag{19.9}$$

where

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

The function $-\mathrm{Log}$ is strictly convex and diverges at the boundary of the positive orthant. Assuming strict feasibility and existence of central-path minimizers $x(t)$, the objective values $f(x(t))$ approach the original optimal value as $t \to +\infty$.

For a fixed t, assume f_t is twice continuously differentiable with positive definite Hessian along the iterates. Apply Newton's method to (19.9), using a line search such as Armijo backtracking (14.14) to choose $0 < \tau_\ell \leqslant 1$ while maintaining $Ax^{(\ell)} < y$:

$$x^{(\ell+1)} \overset{\text{def.}}{=} x^{(\ell)} - \tau_\ell [\partial^2 f_t(x^{(\ell)})]^{-1} \nabla f_t(x^{(\ell)}) \tag{19.10}$$

The gradient and Hessian are

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

A natural stopping criterion for the inner Newton iterations is

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

The barrier method approximately minimizes f_t through (19.10), tracing the central path $t \mapsto x(t)$ for increasing parameters $t = t_k = \mu^k t_0$ with $\mu > 1$. A warm start makes this continuation strategy efficient: initialize the Newton iterations (19.10) for $x(t_k)$ at the preceding solution $x(t_{k-1})$. Along the central path, the logarithmic barrier gives $f(x(t_k)) - f(x^\star) \leqslant m/t_k$, where m is the number of scalar constraints. To reach error ε, take $k = 0, \dots, K$ with

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

Thus $O(|\log(\varepsilon)|)$ barrier-parameter updates suffice for accuracy ε. The total cost also depends on the number of inner Newton iterations (19.10), whose control requires further assumptions on f.

The relevant complexity theory uses self-concordant barriers. A convex function φ on a line is self-concordant when

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

The logarithmic barrier has this property. For a linear objective, the appropriately scaled objective $tf - \mathrm{Log}(y - Ax)$ is self-concordant; arbitrary positive rescaling does not preserve the same self-concordance constant. With a suitable path-following neighborhood and barrier-parameter schedule, the number of Newton steps can be bounded polynomially in the problem parameters and logarithmically in $1/\varepsilon$. The bound depends on the barrier parameter (equal to m for this logarithmic barrier), not only on the requested accuracy. A fixed number of Newton steps per outer iteration cannot be asserted for an arbitrary update factor $\mu > 1$.

19.3 Proximal Algorithm

For a nonsmooth objective f, an explicit gradient step may be undefined. An implicit step remains available even when f is nonsmooth.

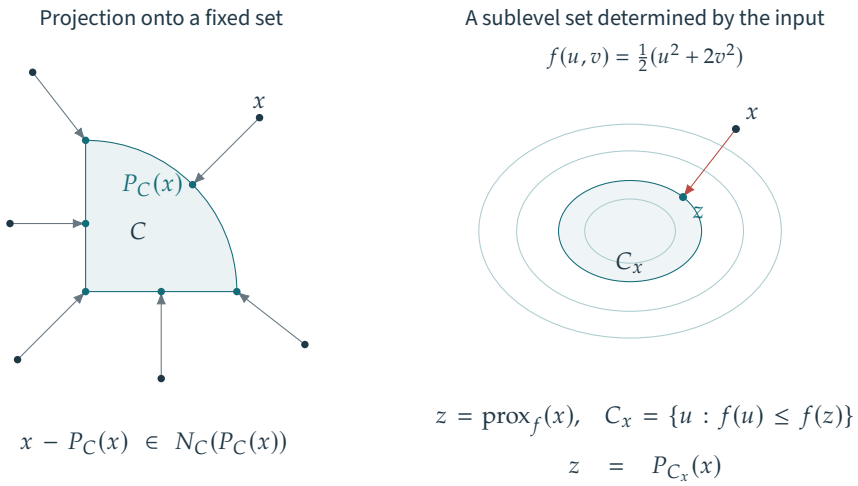

Figure 19.1. Proximal map and projection map.

19.3.1 Proximal Map

For a step size $\tau > 0$, define the proximal map by

$$\forall\, x, \quad \mathrm{Prox}_{\tau f}(x) \overset{\text{def.}}{=} \operatorname*{argmin}_{x'} \frac{1}{2}\|x - x'\|^2 + \tau f(x'). \tag{19.11}$$

It balances a decrease in f against a quadratic penalty for moving away from x. Since $f \in \Gamma_0(\mathcal{H})$, the objective $\frac{1}{2}\|x - \cdot\|^2 + \tau f$ is strongly convex and coercive, so $\mathrm{Prox}_{\tau f}$ is well defined and single-valued.

For an indicator $f = \iota_C$, the proximal map is the projection $\mathrm{Prox}_{\iota_C} = \mathrm{Proj}_C$. It therefore generalizes projection to convex functions. For each fixed input, the proximal point is also the projection onto a suitable sublevel set of f. Like an orthogonal projection, the proximal map is nonexpansive.

Proposition 19.5. *One has* $\|\mathrm{prox}_f(x) - \mathrm{prox}_f(y)\| \leqslant \|x - y\|$.

Examples Several common proximal maps have closed forms.

Proposition 19.6. *One has*

$$\mathrm{Prox}_{\frac{\tau}{2}\|\cdot\|^2}(x) = \frac{x}{1+\tau}, \quad \textit{and} \quad \mathrm{Prox}_{\tau\|\cdot\|_1}(x) = \mathcal{S}^1_\tau(x), \tag{19.12}$$

where the soft-thresholding is defined as

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

(see also (10.5)). For $A \in \mathbb{R}^{P\times N}$, one has

$$\mathrm{Prox}_{\frac{\tau}{2}\|A\cdot - y\|^2}(x) = (\mathrm{Id}_N + \tau A^*A)^{-1}(x + \tau A^* y). \tag{19.13}$$

Proof. The proximal map of $\|\cdot\|_1$ was derived informally in Proposition 10.1. For a rigorous proof, use separability to reduce to a scalar input r. The optimality condition for its proximal point z is $0 \in z - r + \tau\partial|\cdot|(z)$. The candidate $z = 0$ is optimal precisely when $0 \in -r + [-\tau, \tau]$, or equivalently $r \in [-\tau, \tau]$. For $z \neq 0$, optimality requires $z = r - \tau\,\mathrm{sign}(z)$. When $r \notin [-\tau, \tau]$, the candidate $z = r - \tau\,\mathrm{sign}(r)$ has the same sign as r, so it is the unique minimizer. For the quadratic case

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

□

For some nonconvex functions, the proximal minimization also has solutions, but they may be nonunique. For example, $\mathrm{Prox}_{\tau\|\cdot\|_0}$ is hard thresholding at $\sqrt{2\tau}$; see Proposition 10.1. At $|x_i| = \sqrt{2\tau}$, both 0 and x_i are minimizers.

19.3.2 Basic Properties

The following identities simplify proximal computations.

Proposition 19.7. *One has*

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

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

$$\mathrm{Prox}_{\tau f}(x) = (\mathrm{Prox}_{\tau f_k}(x_k))_{k=1}^K. \tag{19.14}$$

Proof. The optimality condition for $z = \mathrm{Prox}_{f+\langle y,\cdot\rangle}(x)$ is $0 \in z - x + \partial f(z) + y$, or $x - y \in z + \partial f(z)$. Thus $z = \mathrm{Prox}_f(x - y)$. For the translated function, substitute $w = z - y$ in $\min_z \{\frac{1}{2}\|z - x\|^2 + f(z - y)\}$ to obtain $z = y + \mathrm{Prox}_f(x - y)$. Separability follows by minimizing each coordinate block independently. □

Composition with a linear map having orthonormal rows admits an explicit formula.

Proposition 19.8. *If* $A \in \mathbb{R}^{P\times N}$ *satisfies* $AA^* = \mathrm{Id}_P$, *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* $\mathrm{Prox}_{f\circ A} = A^* \circ \mathrm{Prox}_f \circ A$.

19.3.3 Related Concepts

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

$$h \in U^{-1}(g) \quad \Longleftrightarrow \quad g \in U(h) \tag{19.15}$$

The proximal map is the resolvent of the subdifferential.

Proposition 19.9. *One has* $\mathrm{Prox}_{\tau f} = (\mathrm{Id} + \tau\partial f)^{-1}$.

Proof. The optimality condition gives the equivalences

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

The final relation is an equality because the inverse is single-valued here. □

This resolvent interpretation connects proximal algorithms with the theory of maximally monotone operators.

Link with duality. Moreau decomposition relates the proximal maps of a function and its conjugate.

Theorem 19.10 (Moreau decomposition)**.** *One has*

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

Thus a tractable proximal map for f gives one for f^*, and conversely. For example, Proposition 18.7 gives the following expression for soft thresholding

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

The minimum and maximum in this clipping formula act componentwise.

For an indicator $f = \iota_C$ of a closed convex cone C,

$$(\iota_C)^* = \iota_{C^\circ} \quad \text{where} \quad C^\circ \overset{\text{def.}}{=} \{y \,;\, \forall\, x \in C, \langle x, y\rangle \leqslant 0\} \tag{19.16}$$

where C° is the polar cone. This conjugacy relation (19.16) is specific to cones: for a general convex set C, the conjugate of its indicator need not be an indicator. Equation (19.16) yields Moreau polar decomposition:

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

The notation $+^\perp$ indicates that the two summands are orthogonal. For a linear subspace $C = V$, this reduces to the orthogonal decomposition $\mathbb{R}^p = V \oplus^\perp V^\perp$.

Link with Moreau–Yosida regularization. A proximal step is also a gradient step on the Moreau–Yosida envelope f_μ of f from (18.4).

Proposition 19.11. *One has*

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

19.4 Proximal Gradient Algorithms

Splitting algorithms exploit a decomposition of the objective into terms whose proximal maps can be computed separately. Different decompositions of the same objective produce different splitting algorithms. A useful decomposition balances the cost of the proximal maps against the convergence of the resulting iteration. Step sizes and other parameters also need to be chosen; theoretical convergence bounds and line-search strategies can guide their selection.

19.4.1 Proximal Point Algorithm

One has the following equivalence

$$x^\star \in \operatorname{argmin} f \quad \Leftrightarrow \quad 0 \in \partial f(x^\star) \quad \Leftrightarrow \quad x^\star \in (\mathrm{Id} + \tau \partial f)(x^\star) \tag{19.17}$$

$$\Leftrightarrow \quad x^\star = (\mathrm{Id} + \tau \partial f)^{-1}(x^\star) = \mathrm{Prox}_{\tau f}(x^\star). \tag{19.18}$$

This shows that being a minimizer of f is equivalent to being a fixed point of $\mathrm{Prox}_{\tau f}$. This fixed-point characterization suggests the proximal point iterations

$$x^{(\ell+1)} \overset{\text{def.}}{=} \mathrm{Prox}_{\tau_\ell f}(x^{(\ell)}). \tag{19.19}$$

The proximal point method does not require an upper stability bound on the step size. Steps must nevertheless be large enough in aggregate; the following sufficient condition is convenient.

Theorem 19.12. *If $0 < \tau_{\min} \leqslant \tau_\ell \leqslant \tau_{\max} < +\infty$, then $x^{(\ell)} \to x^\star$ for a minimizer of f.*

This implicit step (19.19) should be compared with a gradient descent step (19.2)

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

The implicit resolvent $(\mathrm{Id} + \tau_\ell \partial f)^{-1}$ replaces the explicit map $\mathrm{Id} - \tau_\ell \nabla f$. For small τ_ℓ and smooth f, the two agree to first order. The implicit step also remains well defined for nonsmooth objectives and converges under the assumptions above, without the upper step-size restriction of explicit gradient descent. This resembles the greater stability of implicit Euler integration. The computational tradeoff is that a general proximal map may be as difficult to evaluate as the original minimization problem.

19.4.2 Forward–Backward Splitting

For a general objective, evaluating $\mathrm{Prox}_{\gamma f}$ may be as difficult as solving the original problem. A tractable alternative exploits a decomposition into a smooth term and a term with an accessible proximal map:

$$\min_x \mathcal{E}(x) \overset{\text{def.}}{=} f(x) + g(x) \tag{19.20}$$

where $g \in \Gamma_0(\mathcal{H})$ can be nonsmooth, whereas f is convex and has a Lipschitz gradient.

For this structure, the fixed-point argument (19.17) becomes

$$x^\star \in \operatorname{argmin} f + g \quad \Leftrightarrow \quad 0 \in \nabla f(x^\star) + \partial g(x^\star) \quad \Leftrightarrow \quad x^\star - \tau \nabla f(x^\star) \in (\mathrm{Id} + \tau \partial g)(x^\star)$$

$$\Leftrightarrow \quad x^\star = (\mathrm{Id} + \tau \partial g)^{-1} \circ (\mathrm{Id} - \tau \nabla f)(x^\star).$$

This fixed point suggests the following algorithm, known as forward–backward splitting

$$x^{(\ell+1)} \overset{\text{def.}}{=} \mathrm{Prox}_{\tau_\ell g}\left(x^{(\ell)} - \tau_\ell \nabla f(x^{(\ell)})\right). \tag{19.21}$$

Derivation using surrogate functionals. A quadratic surrogate gives another derivation of the algorithm and helps explain its convergence.

Replace the objective $\mathcal{E}(x)$ at the current iterate by a surrogate $\mathcal{E}(x, x^{(\ell)})$ that is easier to minimize, and define

$$x^{(\ell+1)} \overset{\text{def.}}{=} \operatorname*{argmin}_x \mathcal{E}(x, x^{(\ell)}). \tag{19.22}$$

Require the surrogate to majorize the objective and agree with it at the current point:

$$\mathcal{E}(x) \leqslant \mathcal{E}(x, x') \quad \text{and} \quad \mathcal{E}(x, x) = \mathcal{E}(x) \tag{19.23}$$

and $\mathcal{E}(x) - \mathcal{E}(x, x')$ should be a smooth function. Property (19.23) ensures that $\mathcal{E}$ does not increase along the iterations:

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

For proximal-gradient surrogates with a fixed $0 < \tau < 1/L$, sufficient decrease and the optimality conditions imply that every accumulation point is a minimizer. The majorization and touching conditions alone do not establish this conclusion for arbitrary surrogate schemes.

To construct a surrogate for (19.20), use the quadratic upper bound (14.19) supplied by the L-Lipschitz gradient of f:

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

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

$$\mathcal{E}(x, x') \overset{\text{def.}}{=} f(x') + \langle \nabla f(x'),\, x - x' \rangle + \frac{1}{2\tau}\|x - x'\|^2 + g(x) \tag{19.24}$$

satisfies the surrogate conditions (19.23). Minimizing this surrogate is exactly a proximal-gradient step.

Proposition 19.13. *The update* (19.22) *for the surrogate* (19.24) *is exactly* (19.21).

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. The majorization argument uses $\tau \leqslant 1/L$, whereas convergence of the iterates holds on the larger interval $0 < \tau < 2/L$.

Theorem 19.14. *Let f be convex with L-Lipschitz gradient and $g \in \Gamma_0(\mathcal{H})$, and assume* $\operatorname{argmin}(f + g) \neq \emptyset$. *For a fixed step $0 < \tau < 2/L$, the forward–backward iterates converge to a minimizer. If $0 < \tau \leqslant 1/L$, then*

$$(f + g)(x^\ell) - \min(f + g) \leqslant \frac{\|x^0 - x^\star\|^2}{2\tau\ell}, \qquad \ell \geqslant 1.$$

If f is μ-strongly convex, the iterates converge linearly. In particular, for $0 < \tau \leqslant 1/L$, $\|x^{\ell+1} - x^\star\| \leqslant (1 - \tau\mu)\|x^\ell - x^\star\|$.

Taking $g = \iota_C$ in (19.21) recovers projected gradient descent (19.5), because the proximal map of an indicator is the projection onto the constraint set.

Applying (19.21) efficiently requires a tractable proximal map $\operatorname{Prox}_{\tau g}$. A closed form is available for several useful penalties, including the ℓ^1 penalty used by ISTA in Section 10.3.3.

19.5 Primal-Dual Algorithms

Convex duality, developed in Section 18.3, suggests new algorithms and allows existing methods to be applied to dual formulations.

19.5.1 Forward–Backward Splitting on the Dual

As a first example, apply a familiar method to the Fenchel–Rockafellar problem (18.12):

$$p^\star = \inf_x f(x) + g(Ax), \tag{19.25}$$

Assume additionally that f is μ-strongly convex, and, for simplicity, that f and g are finite and continuous everywhere. Even if f is smooth, applying forward–backward splitting directly to the primal problem requires $\operatorname{Prox}_{\tau(g\circ A)}$. This map can be difficult to evaluate despite having a simple $\operatorname{Prox}_{\tau g}$. Proposition 19.8 gives an explicit formula in the special case $AA^* = \mathrm{Id}$.

Example 19.15 (TV denoising). Chambolle [4] developed this approach for total variation denoising:

$$\min_x \frac{1}{2}\|y - x\|^2 + \lambda\|\nabla x\|_{1,2} \tag{19.26}$$

where $\nabla x \in \mathbb{R}^{N\times d}$ is the discrete gradient of a signal ($d = 1$) or image ($d = 2$) with N spatial samples. The mixed ℓ^1–ℓ^2 norm sums the Euclidean norms of the vectors $v_i \in \mathbb{R}^d$:

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

Here

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

Thus f is strongly convex with constant $\mu = 1$, and $A = \nabla$ is the discrete gradient operator.

Example 19.16 (Support Vector Machine). We consider a supervised classification problem with data $(a_i \in \mathbb{R}^p, y_i \in \{-1, +1\})_{i=1}^n$. With regularization parameter $\lambda > 0$, the support vector machine objective is

$$\min_{x\in\mathbb{R}^p} \sum_i \ell(-y_i\langle x,\, a_i\rangle) + \frac{\lambda}{2}\|x\|^2 = \frac{\lambda}{2}\|x\|^2 + g(Ax) \tag{19.27}$$

where $\ell(s) = \max(0, s + 1)$ is the hinge loss and the notation is $A_{i,j} = -y_i a_i[j]$ and $g(u) = \sum_i \ell(u_i)$. It corresponds to the split (19.25) using $f(x) = \frac{\lambda}{2}\|x\|^2$.

Continuity supplies the qualification for Fenchel–Rockafellar duality, Theorem 18.12, giving

$$p^\star = \sup_u \; -g^*(u) - f^*(-A^*u). \tag{19.28}$$

Strong convexity of f with constant μ makes the conjugate f^* differentiable with $1/\mu$-Lipschitz gradient. Apply forward–backward splitting (19.21) to the negative dual objective:

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

Choose a fixed step $0 < \tau < 2/L$, where $L > 0$ bounds the Lipschitz constant of the smooth dual gradient; $L = \|A\|^2/\mu$ is valid when $A \neq 0$. The fixed-step assumption ensures that Theorem 19.14 applies.

Once a dual maximizer $u^\star$ is computed, equivalently a minimizer of the negative dual objective, the primal-dual relations (18.18) recover the unique primal minimizer:

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

using differentiability of f^*.

Example 19.17 (TV denoising). In the particular case of the TV denoising problem (19.26), one has

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

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

The dual, expressed as a minimization, is therefore

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

Projected gradient descent, a special case of forward–backward splitting, gives

$$\mathrm{Prox}_{\tau g^*}(u) = \left(\frac{u_i}{\max\{1, \|u_i\|/\lambda\}} \right)_i \quad \text{and} \quad \nabla f^\star(h) = h + y.$$

Furthermore, one has $\mu = 1$ and $A^*A = -\Delta$ for the usual negative-semidefinite discrete Laplacian. With unit grid spacing, standard forward differences satisfy $\|A\|^2 \leqslant 4d$; equality depends on the boundary conditions and grid size.

Example 19.18 (Support Vector Machine). For the SVM problem (19.27), one has

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

where

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

because the conjugate of $\max(0, s)$ is the indicator of its subdifferential at zero, namely $[0, 1]$. 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 by forward–backward splitting, here equivalent to projected gradient descent. One thus has

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

19.5.2 Douglas–Rachford Splitting

We consider here the structured minimization problem

$$\min_{x \in \mathbb{R}^p} f(x) + g(x), \tag{19.29}$$

Unlike the setting of Section 19.4.2, f need not be smooth. We assume that the proximal maps of both f and g are tractable.

Example 19.19 (Constrained Lasso). Douglas–Rachford splitting applies, for example, to the noiseless constrained Lasso problem (10.11):

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

Take $f = \iota_{C_y}$ with $C_y \overset{\text{def.}}{=} \{x \;;\; Ax = y\}$, and $g = \|\cdot\|_1$. As noted in Section 10.3.1, this problem is equivalent to a linear program. The proximal map of g is soft thresholding (19.12). The proximal map of f is projection onto the affine space C_y, obtained by solving the linear system in (18.11). This projection is particularly efficient for inpainting and for deconvolution operators diagonalized by the Fourier transform.

The Douglas–Rachford iterations read

$$\tilde{x}^{(\ell+1)} \overset{\text{def.}}{=} \left(1 - \frac{\mu}{2}\right)\tilde{x}^{(\ell)} + \frac{\mu}{2}\,\mathrm{rProx}_{\tau g}(\mathrm{rProx}_{\tau f}(\tilde{x}^{(\ell)})) \quad \text{and} \quad x^{(\ell+1)} \overset{\text{def.}}{=} \mathrm{Prox}_{\tau f}(\tilde{x}^{(\ell+1)}), \tag{19.30}$$

where the reflected proximal map is

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

Assume $0 \in \partial f(x^\star) + \partial g(x^\star)$ for some $x^\star$ (for example, a minimizer exists and the relative interiors of the domains intersect). Then, for any $\tau > 0$, any $0 < \mu < 2$, and any $\tilde{x}_0$, the shadow iterates $x^{(\ell)}$ converge to a minimizer of $f + g$, which may depend on the initialization.

Interchanging f and g gives another valid iteration.

More than two functions. A product-space formulation treats K functions $(f_k)_k$ symmetrically:

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

where $\Delta = \{X \,;\, x_1 = \dots = x_K\}$ is the diagonal. The proximal operator of g is

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

The proximal map of f separates into the maps of $(f_k)_k$ by (19.14). Douglas–Rachford then applies through (19.30).

Handling a linear operator. For an objective of the form (19.25), introduce an auxiliary variable:

$$\inf_x f_1(x) + f_2(Ax) = \inf_{z=(x,y)} f(z) + g(z) \quad \text{where} \quad \begin{cases} f(z) = f_1(x) + f_2(y) \\ g(z) = \iota_C(x,y), \end{cases}$$

where $C = \{(x,y) \,;\, Ax = y\}$. Douglas–Rachford splitting applies because the proximal map of f separates into those of f_1 and f_2 by (19.14). The proximal map of g is projection onto C. The following proposition gives two equivalent formulas, allowing a choice between systems in the input and output spaces of A.

Proposition 19.20. *One has*

$$\mathrm{Proj}_C(x,y) = (x - A^*\tilde{y}, y + \tilde{y}) = (\tilde{x}, A\tilde{x}) \quad \textit{where} \quad \begin{cases} \tilde{y} \overset{\text{def.}}{=} (\mathrm{Id}_P + AA^*)^{-1}(Ax - y) \\ \tilde{x} \overset{\text{def.}}{=} (\mathrm{Id}_N + A^*A)^{-1}(A^*y + x). \end{cases} \tag{19.31}$$

Proof. Minimizing $\frac{1}{2}\|z - x\|^2 + \frac{1}{2}\|Az - y\|^2$ gives $(\mathrm{Id}_N + A^*A)z = x + A^*y$, hence $z = \tilde{x}$. Alternatively, the Lagrange multiplier for $Az = w$ satisfies $(\mathrm{Id}_P + AA^*)\tilde{y} = Ax - y$, with $z = x - A^*\tilde{y}$ and $w = y + \tilde{y}$. □

Remark 19.21 (Inverting structured linear operators). Projection computations often require the inverse of a matrix of the form AA^*, A^*A, $\mathrm{Id}_P + AA^*$ or $\mathrm{Id}_N + A^*A$ (see for instance (19.31)). The structure of the operator can make this inexpensive. For inpainting, AA^* is diagonal. For deconvolution or partial Fourier measurements such as MRI, A^*A is diagonalized by the Fourier transform. When inversion is expensive, primal-dual methods can replace it by applications of A and A^*, trading cheaper iterations for a potentially larger iteration count.

19.5.3 Alternating Direction Method of Multipliers

Douglas–Rachford splitting, introduced in Section 19.5.2, can minimize $f + g \circ A$ without strong convexity of f. Its direct application requires the proximal maps $\mathrm{Prox}_{\tau f}$ and $\mathrm{Prox}_{\tau g \circ A}$. The alternating direction method of multipliers (ADMM) is related to Douglas–Rachford splitting on the dual problem (19.28). This formulation uses $\mathrm{Prox}_{\tau g^*}$, computable from $\mathrm{Prox}_{\tau g}$ by Moreau decomposition, and $\mathrm{Prox}_{\tau(f^* \circ A^*)}$. The primal updates derived below express these operations as alternating minimizations.

For $A \in \mathbb{R}^{n \times p}$, introduce $y = Ax$ as in (18.13) and scale the multiplier by $\gamma > 0$. Define

$$\mathcal{L}((x,y),z) \overset{\text{def.}}{=} f(x) + g(y) + \gamma\langle z,\, Ax - y\rangle.$$

The primal problem becomes

$$\inf_{x \in \mathbb{R}^p} f(x) + g(Ax) = \inf_{y = Ax} f(x) + g(y) = \inf_{x \in \mathbb{R}^p, y \in \mathbb{R}^n} \sup_{z \in \mathbb{R}^n} \mathcal{L}((x,y),z). \tag{19.32}$$

Add a quadratic penalty for violating $Ax = y$ to obtain the augmented Lagrangian

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

Replacing $\mathcal{L}$ by $\mathcal{L}_\gamma$ in (19.32) preserves the constrained problem: the supremum over z enforces $Ax = y$, where the added penalty vanishes.

The ADMM method then updates

$$y^{(\ell+1)} \overset{\text{def.}}{=} \operatorname*{argmin}_{y} \mathcal{L}_\gamma((x^{(\ell)}, y), z^{(\ell)}), \tag{19.33}$$

$$x^{(\ell+1)} \overset{\text{def.}}{=} \operatorname*{argmin}_{x} \mathcal{L}_\gamma((x, y^{(\ell+1)}), z^{(\ell)}), \tag{19.34}$$

$$z^{(\ell+1)} \overset{\text{def.}}{=} z^{(\ell)} + Ax^{(\ell+1)} - y^{(\ell+1)}. \tag{19.35}$$

For comparison, the method of multipliers jointly minimizes the augmented Lagrangian in x and y, then updates the multiplier:

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

ADMM instead updates x and y alternately, which often makes the individual subproblems tractable.

Step (19.35) is gradient ascent in the multiplier z with step $1/\gamma$. Assume for this calculation that f is differentiable. Combining the x and z updates enforces $\nabla f(x^{(\ell+1)}) + \gamma A^\top z^{(\ell+1)} = 0$, the stationarity relation in x. Indeed, the optimality condition for (19.34) reads

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

Step (19.33) 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 (19.34) uses a quadratic penalty composed with A:

$$x^{(\ell+1)} \in \operatorname*{argmin}_{x} \left\{ \frac{1}{2}\|Ax - y^{(\ell+1)} + z^{(\ell)}\|^2 + \frac{1}{\gamma} f(x) \right\} \overset{\text{def.}}{=} \operatorname{Prox}^{A}_{f/\gamma}(y^{(\ell+1)} - z^{(\ell)}).$$

Here $\operatorname{Prox}^{A}_{f/\gamma}(u)$ denotes the minimizer set of $\frac{1}{2}\|Ax - u\|^2 + f(x)/\gamma$; it can be set-valued when A is not injective. Under the usual qualification for this subproblem, if

$$p = \operatorname{Prox}_{\gamma(f^* \circ A^*)}(\gamma u),$$

then every minimizing x satisfies $Ax = u - p/\gamma$ and $A^* p \in \partial f(x)$. If A has full column rank, this gives

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

For noninjective A, applying A^+ alone need not recover a minimizer; the component in $\ker(A)$ must also minimize f.

Example 19.22 (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 block soft thresholding for isotropic TV (componentwise soft thresholding for anisotropic TV), while the "twisted" proximal operator is

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

Introducing the multiplier w for $Bx = y$ gives first-order conditions for $(x^\star, w^\star)$ in the form of a linear system:

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

Here $\Delta = -\nabla^\top \nabla$ is the discrete Laplacian.

19.5.4 Primal-Dual Splitting

If neither $\operatorname{Prox}_{\tau g \circ A}$ nor $\operatorname{Prox}_{\tau f^\star \circ A^*}$ is tractable, a primal-dual method can work directly with the saddle problem. Its efficiency depends on the available operator evaluations, proximal maps, and conditioning. For the structured problem (19.25), biconjugacy $g = (g^*)^*$ gives the saddle formulation

$$\inf_x f(x) + g(Ax) = \inf_x f(x) + \sup_u \langle Ax,\, u\rangle - g^*(u) = \sup_u \inf_x f(x) + \langle Ax,\, u\rangle - g^*(u). \tag{19.36}$$

We assume a primal-dual saddle point exists, so strong duality holds; strong convexity of f is not required.

Example 19.23 (Total variation regularization of inverse problems). For example, TV regularization for the inverse problem $\mathcal{K}x = y$ leads to

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

where $A = \nabla$, $f(x) = \frac{1}{2}\|y - \mathcal{K}x\|^2$, and $g = \lambda\|\cdot\|_{1,2}$. For this split, the proximal map of f requires solving a system involving $\mathrm{Id} + \tau\mathcal{K}^*\mathcal{K}$, while the proximal map of g^* is a pointwise projection. If that linear solve is costly, use instead $Ax = (\mathcal{K}x, \nabla x)$, $f = 0$, and $g(z, v) = \frac{1}{2}\|y - z\|^2 + \lambda\|v\|_{1,2}$. The two blocks of g^* then have simple proximal maps, so each iteration uses only $\mathcal{K}, \mathcal{K}^*, \nabla$, and ∇^*.

A standard primal-dual algorithm, described in [6], initializes $\tilde{x}^0 = x^0$ and updates

$$\begin{aligned} z^{(\ell+1)} &= \operatorname{Prox}_{\sigma g^*}\left(z^{(\ell)} + \sigma A\tilde{x}^{(\ell)}\right), \\ x^{(\ell+1)} &= \operatorname{Prox}_{\tau f}\left(x^{(\ell)} - \tau A^* z^{(\ell+1)}\right), \\ \tilde{x}^{(\ell+1)} &= x^{(\ell+1)} + \theta(x^{(\ell+1)} - x^{(\ell)}). \end{aligned}$$

For $\theta = 1$, $\sigma, \tau > 0$, and $\sigma\tau\|A\|^2 < 1$, the primal and dual iterates converge to a saddle point, provided one exists. The value $\theta = 0$ is not covered by this general convergence guarantee.

References

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

[5] 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.

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

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