2 Proximal Gradient

Goal

Proximal map, Moreau envelope, composite minimization, proximal gradient algorithm.

Alert 2.1: Convention

Gray boxes are not required hence can be omitted for unenthusiastic readers.

This note is likely to be updated again soon.

Definition 2.2: Problem

In this lecture we study the following problem:

$$\inf_{\mathbf{w}\in\mathbb{R}^d} \quad f(\mathbf{w}), \quad \text{where} \quad f(\mathbf{w}) = \ell(\mathbf{w}) + r(\mathbf{w}) \tag{2.1}$$

is the sum of a smooth function ℓ and a possibly non-differentiable function r. This problem is sometimes called composite minimization. Due to the possible non-differentiability of r (and hence f), we cannot directly apply gradient descent.

Example 2.3: Lasso (Tibshirani, 1996)

We continue our discussion of linear regression in Example 1.3. Suppose we know the optimal solution $\mathbf{w}$ is sparse, i.e., only a few entries in $\mathbf{w}$ are nonzero, how do we exploit this information and make our estimation more efficient? Obviously, if an oracle knew which entries of $\mathbf{w}$ are zero, it would simply fix those entries to 0 and reduce the dimensionality of our problem. Without hindsight, Lasso (Tibshirani, 1996) is an algorithm that performs (almost) as well as the mighty oracle, by adding simply an ℓ_1-norm regularizer on $\mathbf{w}$:

$$\min_{\mathbf{w}} \ \underbrace{\frac{1}{n}\|\mathsf{X}^\top\mathbf{w} - \mathbf{y}\|_2^2}_{\ell} + \underbrace{\lambda\|\mathbf{w}\|_1}_{r},$$

where $\lambda > 0$ is some tuning hyperparameter. The ℓ_1 norm $\|\cdot\|_1$ is incorporated to induce sparsity in the minimizer $\mathbf{w}$, a claim that will become clear once we see the algorithm.

We note that even when the optimal solution $\mathbf{w}$ is dense, it might still make sense to "sparsify" it. For instance, we may have memory or communication restrictions on the bits that can be used to represent $\mathbf{w}$, which is typical in embedded/mobile system and known as model compression/quantization.

Tibshirani, R. (1996). "Regression Shrinkage and Selection via the Lasso". *Journal of the Royal Statistical Society: Series B*, vol. 58, no. 1, pp. 267–288.

Definition 2.4: Proximal map and Moreau envelope (Moreau, 1965)

The proximal map and Moreau envelope of a (closed) function f is defined respectively as:

$$\mathrm{P}_f^\eta(\mathbf{z}) := \operatorname*{argmin}_{\mathbf{w}} \tfrac{1}{2\eta}\|\mathbf{w}-\mathbf{z}\|_2^2 + f(\mathbf{w}), \qquad \mathrm{M}_f^\eta(\mathbf{z}) := \inf_{\mathbf{w}} \tfrac{1}{2\eta}\|\mathbf{w}-\mathbf{z}\|_2^2 + f(\mathbf{w}),$$

where $\eta > 0$ is a *smoothing* parameter. We define the proximal threshold

$$\mathsf{t}_f := \frac{1}{\left(-2\cdot\liminf\limits_{\|\mathbf{w}\|\to\infty}\frac{f(\mathbf{w})}{\|\mathbf{w}\|_2^2}\right)_+}, \quad \text{where} \quad 1/0 := \infty.$$

Then, for all $\eta \in]0, \mathrm{t}_f[$, it follows from (a slight generalization of) Theorem 0.33 that the proximal map is well-defined (i.e. non-empty and in fact compact-valued). In particular, if f is lower bounded by a constant (which holds frequently in practice), or if f is convex (hence lower bounded by a linear function), then the proximal map is well-defined for any $\eta > 0$ and reduces to a singleton for the latter case.

Moreau, J. J. (1965). "Proximité et Dualtité dans un Espace Hilbertien". *Bulletin de la Société Mathématique de France*, vol. 93, pp. 273–299.

Exercise 2.5: Moreau envelope preserves minimizer

Prove that

- $\forall \eta > 0, \quad \mathrm{M}_f^\eta \leq f$.
- $\forall \eta > 0, \quad \inf f = \inf \mathrm{M}_f^\eta \qquad$ and $\qquad \operatorname{argmin} f = \operatorname{argmin} \mathrm{M}_f^\eta$.
- $\forall \mathbf{w}, \quad \mathrm{M}_f^\eta(\mathbf{w}) \to f(\mathbf{w})$ as $\eta \downarrow 0$.

(The last one may be a bit difficult.)

Moreover, it can be shown that the Moreau envelope M_f^η is "smoother" than f (Moreau, 1965; Lasry and Lions, 1986; Jourani et al., 2014; Kecis and Thibault, 2015). When the pointwise convergence in the last item is uniform, we can use the Moreau envelope to develop a faster algorithm for certain non-smooth functions, as will be discussed in a later lecture. Below, the proximal map is our main object of interest.

Moreau, J. J. (1965). "Proximité et Dualtité dans un Espace Hilbertien". *Bulletin de la Société Mathématique de France*, vol. 93, pp. 273–299.

Lasry, J. M. and P. L. Lions (1986). "A remark on regularization in Hilbert spaces". *Israel Journal of Mathematics*, vol. 55, pp. 257–266.

Jourani, A., L. Thibault, and D. Zagrodny (2014). "Differential properties of the Moreau envelope". *Journal of Functional Analysis*, vol. 266, no. 3, pp. 1185–1237.

Kecis, I. and L. Thibault (2015). "Moreau envelopes of s-lower regular functions". *Nonlinear Analysis: Theory, Methods & Applications*, vol. 127, pp. 157–181.

Alert 2.6: Notation

From now on we allow a function f to take value ∞ (but not $-\infty$). In particular, for $f : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$, we define its domain

$$\operatorname{dom} f = \{\mathbf{w} : f(\mathbf{w}) < \infty\}.$$

This simple trick allows us to embed the domain of a function into itself. It also allows us to identify a set $C \subseteq \mathbb{R}^d$ as an indicator function

$$\iota_C(\mathbf{w}) = \begin{cases} 0, & \text{if } \mathbf{w} \in C \\ \infty, & \text{otherwise} \end{cases}.$$

In particular, we can rewrite a constrained minimization problem as a *seemingly* unconstrained composite minimization problem:

$$\inf_{\mathbf{w} \in C} \; f(\mathbf{w}) \qquad \equiv \qquad \inf_{\mathbf{w} \in \mathbb{R}^d} \; f(\mathbf{w}) + \iota_C(\mathbf{w}) \, .$$

The reason why we do not allow f to take $-\infty$ is because we are interested in minimizing: if a function takes $-\infty$ at some point $\mathbf{w}$, then $\mathbf{w}$ is trivially the (global) minimizer and there is nothing to minimize.

Example 2.7: Euclidean projection as proximal map

With the above notation, it is clear that

$$\mathrm{P}_C(\mathbf{w}) = \mathrm{P}^\eta_{\iota_C}(\mathbf{w})$$

for any $\eta > 0$.

Example 2.8: Positive part

Let $f(t) = \lambda t_+ := \lambda \max\{t, 0\} := \lambda t \vee 0$ be the positive part (or relu, rectified linear unit, in fancier NN terminology). It is clear that f is not differentiable but λ-Lipschitz continuous (assuming $\lambda \geq 0$):

$$|t_+ - s_+| \leq \lambda \cdot |t - s|.$$

Let us compute the proximal map:

$$\mathrm{P}^\eta_f(s) = \operatorname*{argmin}_t \tfrac{1}{2\eta}(s-t)^2 + \lambda t_+ = \operatorname*{argmin}_t \begin{cases} \frac{1}{2\eta}(s-t)^2 + \lambda t, & t \geq 0 \\ \frac{1}{2\eta}(s-t)^2, & t \leq 0 \end{cases}.$$

We have two possible minimizers $t = (s - \lambda\eta)_+$ and $t = s \wedge 0 := \min\{s, 0\}$ with minimum value $\frac{1}{2\eta}((\lambda\eta) \wedge s)^2 + \lambda(s - \lambda\eta)_+$ and $\frac{1}{2\eta}(s_+)^2$, respectively. Comparing the two minimum values in the two cases and taking the smaller one:

$$\mathrm{P}^\eta_f(s) = \begin{cases} s - \lambda\eta, & \text{if } s \geq \lambda\eta \\ s \wedge 0, & \text{if } s \leq \lambda\eta \end{cases}, \quad \text{with minimum value} \quad \mathrm{M}^\eta_f(s) = \begin{cases} \lambda s - \frac{\eta\lambda^2}{2}, & \text{if } s \geq \lambda\eta \\ \frac{1}{2\eta}(s_+)^2, & \text{if } s \leq \lambda\eta \end{cases}.$$

Exercise 2.9: L-smoothness and uniform approximation of Moreau's envelope

Prove the following:

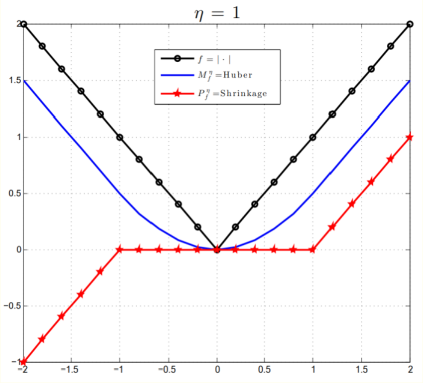
- The Moreau envelope in Example 2.8 is differentiable and $\frac{1}{\eta}$-smooth.
- The following uniform approximation holds:

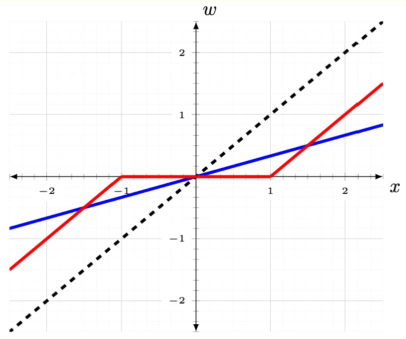
$$\sup_{s \in \mathbb{R}} |\mathrm{M}^\eta_f(s) - \lambda(s)_+| \leq \tfrac{\eta \cdot \lambda^2}{2},$$

  where recall that the positive part λs_+ is λ-Lipschitz continuous.

Example 2.10: Soft-shrinkage

In your assignment you are going to derive the proximal map for the ℓ_1 norm, a.k.a. the soft-shrinkage operator. In the univariate case, the picture looks like the following plot on the left:





A striking property we observe is that small inputs to the proximal map (red curve) get truncated to 0, i.e. gaining sparsity! Interestingly, this sparsity-promoting property of the ℓ_1 norm can be attributed to its nondifferentiability at the origin. In contrast, the proximal map of the squared Euclidean norm (shown in blue on the right plot) loses this sparsity-inducing property since it is smooth.

Exercise 2.11: If it works for the ℓ_2 norm, it should work for any norm?

Do the following:

- Derive the proximal map of the Euclidean norm (without squaring). Does it induce sparsity? You may want to go over Example 19.9 first.
- Any norm is not differentiable at the origin.
- Unlike the ℓ_2 norm case, squaring the ℓ_1 norm does not make it differentiable. This is a laughable mistake that is not uncommon in reality.

Example 2.12: Sparsemax

The celebrated `softmax` can be derived as the minimizer of the following constrained problem:

$$\begin{aligned}\texttt{softmax}(\mathbf{w}) &:= \operatorname*{argmin}_{\mathbf{z}\in\Delta}\ \langle \mathbf{w}, \mathbf{z}\rangle + \lambda \sum_j z_j \log z_j \\ &\propto \exp(-\mathbf{w}/\lambda),\end{aligned}$$

where $\Delta := \{\mathbf{z} \in \mathbb{R}^d_+ : \mathbf{1}^\top \mathbf{z} = 1\}$ is the simplex (so that $\mathbf{z} \in \Delta$ is a discrete distribution), and the last term is the so-called (negative) entropy. The problem of `softmax` is that its output is always dense due to exponentiation. If we replace negative entropy with a quadratic function, we obtain

$$\begin{aligned}\texttt{sparsemax}(\mathbf{w}) &:= \operatorname*{argmin}_{\mathbf{z}\in\Delta}\ \langle \mathbf{w}, \mathbf{z}\rangle + \tfrac{\lambda}{2}\|\mathbf{z}\|_2^2 \\ &= \mathrm{P}_\Delta(-\mathbf{w}/\lambda).\end{aligned}$$

Adam and Mácha (2020) gave a nice historic recount of an $O(d \log d)$ time algorithm for computing `sparsemax`, from which it is clear that `sparsemax` indeed induces sparsity (as the name suggests). Of course, one can change the negative entropy to a variety of functions, leading to different variants of the `softmax`, see Correia et al. (2019) and Martins et al. (2020) and references therein for applications in attention and transformers, and for applications in optimal transportation (e.g. Muzellec et al., 2017; Blondel et al., 2018; Dessein et al., 2018).

Adam, L. and V. Mácha (2020). "Projections onto the canonical simplex with additional linear inequalities". *Optimization Methods and Software*, pp. 1–29.

Correia, G. M., V. Niculae, and A. F. T. Martins (2019). "Adaptively Sparse Transformers". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2174–2184.

Martins, A. F. T., M. Treviso, A. Farinhas, V. Niculae, M. A. T. Figueiredo, and P. M. Q. Aguiar (2020). "Sparse and Continuous Attention Mechanisms".

Muzellec, B., R. Nock, G. Patrini, and F. Nielsen (2017). "Tsallis Regularized Optimal Transport and Ecological Inference". In: *AAAI Conference on Artificial Intelligence*.

Blondel, M., V. Seguy, and A. Rolet (2018). "Smooth and Sparse Optimal Transport". In: *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, pp. 880–889.

Dessein, A., N. Papadakis, and J.-L. Rouas (2018). "Regularized Optimal Transport and the Rot Mover's Distance". *Journal of Machine Learning Research*, vol. 19, no. 15, pp. 1–53.

Remark 2.13: More on proximal map

For those who are familiar with convex analysis, the "bible" of Rockafellar and Wets (1998) contains lots of useful results on the proximal map, some of which were also documented in Yu (2013) and Yu et al. (2015). We mention some notable further contributions: Penot (1998), Hare and Poliquin (2007), Hiriart-Urruty and Le (2013), and Planiden and Wang (2016, 2019).

Rockafellar, R. T. and R. J.-B. Wets (1998). "Variational Analysis". Springer.

Yu, Y.-L. (2013). "On Decomposing the Proximal Map". In: *Advances in Neural Information Processing Systems 27 (NIPS)*.

Yu, Y., X. Zheng, M. Marchetti-Bowick, and E. P. Xing (2015). "Minimizing Nonconvex Non-Separable Functions". In: *The 17^{th} International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Penot, J.-P. (1998). "Proximal Mappings". *Journal of Approximation Theory*, vol. 94, no. 2, pp. 203–221.

Hare, W. L. and R. A. Poliquin (2007). "Prox-Regularity and Stability of the Proximal Mapping". *Journal of Convex Analysis*, vol. 14, no. 3, pp. 589–6068.

Hiriart-Urruty, J.-B. and H. Y. Le (2013). "From Eckart and Young approximation to Moreau envelopes and vice versa". *RAIRO: Operations Research*, vol. 47, no. 3, pp. 299–310.

Planiden, C. and X. Wang (2016). "Strongly Convex Functions, Moreau Envelopes, and the Generic Nature of Convex Functions with Strong Minimizers". *SIAM Journal on Optimization*, vol. 26, no. 2, pp. 1341–1364.

— (2019). "Proximal Mappings and Moreau Envelopes of Single-Variable Convex Piecewise Cubic Functions and Multivariable Gauge Functions". In: *Nonsmooth Optimization and Its Applications*, pp. 89–130.

Algorithm 2.14: Proximal point algorithm (PPA, Martinet, 1970; Rockafellar, 1976)

We now present our first algorithm for non-smooth minimization (constrained or not):

```
Algorithm: Proximal Point Algorithm (PPA)
Input: $\mathbf{w}_0$
for $t = 0, 1, \ldots$ do
    choose step size $\eta_t$
    $\mathbf{w}_{t+1} \leftarrow \mathrm{P}_f^{\eta_t}(\mathbf{w}_t) = \operatorname{argmin}_{\mathbf{w}} \frac{1}{2\eta_t}\|\mathbf{w}_t - \mathbf{w}\|_2^2 + f(\mathbf{w})$        // proximal step
```

We will derive the convergence property of PPA later in **??**. For now let us point out that:

- PPA is not practical in its current form: we aim to minimize a function f and PPA blows it up by minimizing a sequence of functions that are simply f regularized by a quadratic.
- PPA can handle nonsmooth, nonconvex functions and allow very relaxed choices of the step size.
- PPA is an extremely important theoretical tool. Many of our later algorithms can be interpreted as various approximations of PPA.
- PPA is a backward form of gradient descent:

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \nabla f(\mathbf{w}_{t+1}).$$

Martinet, B. (1970). “Régularisation d’Inéquations Variationnelles par Approximations Successives”. *Revue Française d’Informatique et de Recherche Opérationnelle, Série Rouge*, vol. 4, no. 3, pp. 154–158.

Rockafellar, R. T. (1976). “Monotone Operators and the Proximal Point Algorithm”. *SIAM Journal on Control and Optimization*, vol. 14, no. 5, pp. 877–898.

Remark 2.15: Time flows backwards

In the gradient descent Algorithm 1.4, we climb down from $\mathbf{w}_0$ to a stationary point $\mathbf{w}_*$:

$$\mathbf{w}_0 \to \mathbf{w}_1 \to \cdots \to \mathbf{w}_t \to \mathbf{w}_{t+1} \to \cdots \to \mathbf{w}_*, \quad \text{where of course } \mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \nabla f(\mathbf{w}_t).$$

In contrast, PPA aims to find $\tilde{\mathbf{w}}_{t+1}$ so that if time flows backwards, we would climb up to $\tilde{\mathbf{w}}_t$ hence all the way back to $\mathbf{w}_0$ through gradient ascent:

$$\mathbf{w}_0 \leftarrow \tilde{\mathbf{w}}_1 \leftarrow \cdots \leftarrow \tilde{\mathbf{w}}_t \leftarrow \tilde{\mathbf{w}}_{t+1} \leftarrow \cdots \leftarrow \mathbf{w}_*, \quad \text{where} \quad \tilde{\mathbf{w}}_t = \tilde{\mathbf{w}}_{t+1} + \eta_t \nabla f(\tilde{\mathbf{w}}_{t+1}).$$

This interpretation reveals the difficulty in PPA: without hindsight, we need to carefully construct $\{\tilde{\mathbf{w}}_t\}$ so that if time reverses, we obtain a sequence generated by gradient ascent.

Exercise 2.16: Tenet

Let $\mathbf{z} = \mathbf{w} - \eta \nabla f(\mathbf{w})$ be the gradient descent update, and $\tilde{\mathbf{z}} = \mathrm{P}_f^{\tilde{\eta}}(\mathbf{w})$ be the PPA update, i.e. $\mathbf{w} = \tilde{\mathbf{z}} + \tilde{\eta} \nabla f(\tilde{\mathbf{z}})$. Prove that, in general, no matter what $\tilde{\eta}$ we choose (to correspond to η), we may not match $\mathbf{z} = \tilde{\mathbf{z}}$. [Hint: a quadratic f may suffice.]

In other words, the two paths below

$$\begin{tikzcd}
& \mathbf{w}_1 \arrow[r, bend left] & \cdots \arrow[r, bend left] & \mathbf{w}_t \arrow[r, bend left] & \mathbf{w}_{t+1} \arrow[r, bend left] & \cdots \arrow[dr, bend left] & \\
\mathbf{w}_0 \arrow[ur, bend left] & & & & & & \mathbf{w}_* \arrow[dl, bend left] \\
& \tilde{\mathbf{w}}_1 \arrow[ul, bend left] & \cdots \arrow[l, bend left] & \tilde{\mathbf{w}}_t \arrow[l, bend left] & \tilde{\mathbf{w}}_{t+1} \arrow[l, bend left] & \cdots \arrow[l, bend left] &
\end{tikzcd}$$

do not overlap.

Algorithm 2.17: Proximal gradient (PG, Bruck, 1977; Fukushima and Mine, 1981)

```
Algorithm: Proximal Gradient (PG) for composite minimization
  Input: $\mathbf{w}_0 \in \mathrm{dom}\, f$
  for $t = 0, 1, \ldots$ do
      choose step size $\eta_t$
      $\mathbf{z}_t \leftarrow \mathbf{w}_t - \eta_t \nabla \ell(\mathbf{w}_t)$                                   // gradient step w.r.t. $\ell$
      $\mathbf{w}_{t+1} \leftarrow \mathrm{P}_r^{\eta_t}(\mathbf{z}_t) = \mathrm{argmin}_{\mathbf{w}} \frac{1}{2\eta_t}\|\mathbf{z}_t - \mathbf{w}\|_2^2 + r(\mathbf{w})$     // proximal step w.r.t. $r$
```

In a nutshell, the algorithm consists of two steps:

- In the first step, we ignore the (possibly) nonsmooth function r and perform (forward) gradient descent w.r.t. ℓ.
- In the second step, we ignore the smooth function ℓ and perform the proximal map w.r.t. r (i.e. backward gradient update).

For the above reason, PG is also known as the forward-backward splitting algorithm.

Obviously, the efficiency of PG largely depends on how fast we can compute the proximal map $\mathrm{P}_r^{\eta}(\mathbf{w})$, which is itself a minimization problem. The following special cases are obvious:

- $r \equiv 0$: we then recover the gradient descent algorithm.

- $r = \iota_C$: we then recover the projected gradient algorithm.
- $\ell \equiv 0$: we then recover the proximal point algorithm.
- We may also interpret PG as an approximation of PPA, where we replace the smooth function ℓ by its linearization:

$$\mathbf{w}_{t+1} = \operatorname*{argmin}_{\mathbf{w}} \ \ell(\mathbf{w}_t) + \langle \mathbf{w} - \mathbf{w}_t, \nabla \ell(\mathbf{w}_t)\rangle + \tfrac{1}{2\eta_t}\|\mathbf{w} - \mathbf{w}_t\|_2^2 + r(\mathbf{w}). \tag{2.2}$$

Bruck, R. E. (1977). "On the weak convergence of an ergodic iteration for the solution of variational inequalities for monotone operators in Hilbert space". *Journal of Mathematical Analysis and Applications*, vol. 61, no. 1, pp. 159–164.

Fukushima, M. and H. Mine (1981). "A Generalized Proximal Point Algorithm for Certain Non-Convex Minimization Problems". *International Journal of Systems Science*, vol. 12, no. 8, pp. 989–1000.

Definition 2.18: Bregman divergence (Bregman, 1967)

A differentiable convex function $f : \mathbb{R}^d \to \mathbb{R}$ induces a Bregman divergence:

$$\mathrm{D}_f(\mathbf{z}; \mathbf{w}) = f(\mathbf{z}) - f(\mathbf{w}) - \langle \mathbf{z} - \mathbf{w}, \nabla f(\mathbf{w})\rangle \geq 0,$$

where the inequality follows from convexity (see Theorem 0.29). In general, Bregman divergences are not symmetric, i.e. $\mathrm{D}_f(\mathbf{z}; \mathbf{w}) \neq \mathrm{D}_f(\mathbf{w}; \mathbf{z})$.

Bregman, L. M. (1967). "The Relaxation Method of Finding the Common Point of Convex Sets and Its Application to the Solution of Problems in Convex Programming". *USSR Computational Mathematics and Mathematical Physics*, vol. 7, no. 3, pp. 200–217. [English translation in *Zh. Vȳchisl. Mat. mat. Fiz.* vol. 7, no. 3, pp. 620–631, 1967].

Example 2.19: Bregman divergence of Bregman divergence

Let $f(\mathbf{w}) = \frac{1}{2}\|\mathbf{w} - \mathbf{w}_0\|_2^2$ for any $\mathbf{w}_0$. Then, we easily verify that $\mathrm{D}_f(\mathbf{z}; \mathbf{w}) = \frac{1}{2}\|\mathbf{z} - \mathbf{w}\|_2^2$, i.e. the squared Euclidean distance, which is the *only* symmetric Bregman divergence (an observation attributed to A. N. Iusem by Bauschke and Borwein (2001)).

More generally, if h is continuously differentiable, and $f(\mathbf{w}) = \mathrm{D}_h(\mathbf{w}; \mathbf{w}_0)$, then

$$\mathrm{D}_f(\mathbf{z}; \mathbf{w}) = \mathrm{D}_h(\mathbf{z}; \mathbf{w}_0) - \mathrm{D}_h(\mathbf{w}; \mathbf{w}_0) - \langle \mathbf{z} - \mathbf{w}, \nabla h(\mathbf{w}) - \nabla h(\mathbf{w}_0)\rangle = \mathrm{D}_h(\mathbf{z}; \mathbf{w}).$$

Bauschke, H. H. and J. M. Borwein (2001). "Joint and Separate Convexity of the Bregman Distance". In: *Inherently Parallel Algorithms in Feasibility and Optimization and their Applications.* Vol. 8, pp. 23–36.

Proposition 2.20: Composite optimality (Tseng, 2008)

Let $\ell : \mathbb{R}^d \to \mathbb{R}$ be differentiable convex and $r : \mathbb{R}^d \to \mathbb{R} \cup \{\infty\}$ be convex. Then,

$$\mathbf{w}_\star \in \operatorname*{argmin}_{\mathbf{w}} \ \ell(\mathbf{w}) + r(\mathbf{w}) \iff \forall \mathbf{w}, \quad \ell(\mathbf{w}) + r(\mathbf{w}) \geq \ell(\mathbf{w}_\star) + r(\mathbf{w}_\star) + \mathrm{D}_\ell(\mathbf{w}; \mathbf{w}_\star).$$

Proof: $\Longleftarrow$: By convexity of ℓ we know the Bregman divergence $\mathrm{D}_\ell(\mathbf{w}; \mathbf{w}_\star) \geq 0$, hence $\mathbf{w}_\star$ is clearly a (global) minimizer.

$\Longrightarrow$: Since $\mathbf{w}_\star$ is a minimizer, we have $\mathbf{0} \in \partial(\ell + r)(\mathbf{w}_\star) = \nabla \ell(\mathbf{w}_\star) + \partial r(\mathbf{w}_\star)$ where the second equality follows from convexity and continuous differentiability. Using convexity of r we have

$$r(\mathbf{w}) \geq r(\mathbf{w}_\star) + \langle \mathbf{w} - \mathbf{w}_\star, \partial r(\mathbf{w}_\star) \rangle .$$

Therefore,

$$\begin{aligned} \ell(\mathbf{w}) + r(\mathbf{w}) &\geq \ell(\mathbf{w}_\star) + r(\mathbf{w}_\star) + \ell(\mathbf{w}) - \ell(\mathbf{w}_\star) + \langle \mathbf{w} - \mathbf{w}_\star, -\nabla \ell(\mathbf{w}_\star) \rangle \\ &\geq \ell(\mathbf{w}_\star) + r(\mathbf{w}_\star) + \mathrm{D}_\ell(\mathbf{w}; \mathbf{w}_\star). \end{aligned}$$

■

Tseng, P. (2008). "On Accelerated Proximal Gradient Methods for Convex-Concave Optimization".

Theorem 2.21: Convergence of PG in function value (Beck and Teboulle, 2009; Tseng, 2008)

Let $\ell : \mathbb{R}^d \to \mathbb{R}$ be convex and $\mathsf{L} = \mathsf{L}_2^{[1]}$-smooth, $r : \mathbb{R} \to \mathbb{R} \cup \{\infty\}$ be closed and convex, and η_t is chosen so that (2.3) *below holds. Then, for all $\mathbf{w}$ and $t \geq 1$, the sequence $\{\mathbf{w}_t\}$ generated by Algorithm 2.17 satisfies:*

$$f(\mathbf{w}_t) \leq f(\mathbf{w}) + \frac{\|\mathbf{w} - \mathbf{w}_0\|_2^2}{2t\bar{\eta}_t}, \quad \textit{where} \quad \bar{\eta}_t := \frac{1}{t}\sum_{s=0}^{t-1} \eta_s.$$

Proof: We learned the following slick proof from Tseng (2008).

From the definition of $\mathbf{w}_{t+1}$ in (2.2) and Proposition 2.20 we know for any $\mathbf{w}$:

$$\begin{aligned} f(\mathbf{w}_{t+1}) = \ell(\mathbf{w}_{t+1}) + r(\mathbf{w}_{t+1}) &\leq \ell(\mathbf{w}_t) + \langle \mathbf{w}_{t+1} - \mathbf{w}_t, \nabla \ell(\mathbf{w}_t) \rangle + \tfrac{1}{2\eta_t}\|\mathbf{w}_{t+1} - \mathbf{w}_t\|_2^2 + r(\mathbf{w}_{t+1}) \qquad (2.3) \\ &\leq \ell(\mathbf{w}_t) + \langle \mathbf{w} - \mathbf{w}_t, \nabla \ell(\mathbf{w}_t) \rangle + \tfrac{1}{2\eta_t}\|\mathbf{w} - \mathbf{w}_t\|_2^2 + r(\mathbf{w}) - \tfrac{1}{2\eta_t}\|\mathbf{w} - \mathbf{w}_{t+1}\|_2^2 \\ &\leq \ell(\mathbf{w}) + r(\mathbf{w}) + \tfrac{1}{2\eta_t}\|\mathbf{w} - \mathbf{w}_t\|_2^2 - \tfrac{1}{2\eta_t}\|\mathbf{w} - \mathbf{w}_{t+1}\|_2^2, \end{aligned}$$

where the last inequality is due to the convexity of ℓ. Take $\mathbf{w} = \mathbf{w}_t$ we see $f(\mathbf{w}_{t+1}) \leq f(\mathbf{w}_t)$, i.e., the algorithm is descending. Summing from $t = 0$ to $t = T - 1$:

$$T\bar{\eta}_T \cdot [f(\mathbf{w}_T) - f(\mathbf{w})] \leq \sum_{t=0}^{T-1} \eta_t [f(\mathbf{w}_{t+1}) - f(\mathbf{w})] \leq \frac{1}{2}\|\mathbf{w} - \mathbf{w}_0\|_2^2, \quad \text{where} \quad \bar{\eta}_T := \frac{1}{T}\sum_{t=0}^{T-1} \eta_t,$$

Dividing both sides by $T\bar{\eta}_T$ completes the proof. ■

If there exists a minimizer $\mathbf{w}_\star$, then we have

$$f(\mathbf{w}_t) - f_\star \leq \frac{\mathsf{L}\|\mathbf{w}_\star - \mathbf{w}_0\|_2^2}{2t}$$

where we have chosen $\eta_t \equiv 1/\mathsf{L}$ to minimize the bound. So the function value converges to the global minimum (thanks to convexity) at the rate of $O(1/t)$. As before, the dependence on L and $\mathbf{w}_0$ makes intuitive sense. Again, the rate of convergence does not depend on d, the dimension!

The L-smoothness condition is used only to make sure inequality (2.3) is not vacuous. If we do not know L in advance, we can apply Amijo's backtracking as before (see Remark 1.20) until (2.3) is satisfied. Note that the term $r(\mathbf{w}_{t+1})$ is cancelled from both sides in Equation (2.3).

The open-loop step size in Remark 19.18 also applies in verbatim here.

Beck, A. and M. Teboulle (2009). "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems". *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202.

Tseng, P. (2008). "On Accelerated Proximal Gradient Methods for Convex-Concave Optimization".

Remark 2.22: Convergence of iterates

It can be proved using fixed point theorems that the iterates $\mathbf{w}_t$ also converge (provided that a minimizer exists), see e.g. Tseng (2000), Daubechies et al. (2004), and Combettes and Wajs (2005) and the excellent book of Bauschke and Combettes (2017) for much more. Moreover, $\mathbf{w}_t$ converges (to a stationary point) even when f is nonconvex, provided that it is "definable," see e.g. Attouch and Bolte (2009), Attouch et al. (2013), and Bolte et al. (2014).

Tseng, P. (2000). "A Modified Forward-Backward Splitting Method for Maximal Monotone Mappings". *SIAM Journal on Control and Optimization*, vol. 38, no. 2, pp. 431–446.

Daubechies, I., M. Defrise, and C. De Mol (2004). "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint". *Communications on Pure and Applied Mathematics*, vol. 57, no. 11, pp. 1413–1457.

Combettes, P. L. and V. R. Wajs (2005). "Signal Recovery by Proximal Forward-Backward Splitting". *Multiscale Modeling and Simulation*, vol. 4, no. 4, pp. 1168–1200.

Bauschke, H. H. and P. L. Combettes (2017). "Convex Analysis and Monotone Operator Theory in Hilbert Spaces". 2nd. Springer.

Attouch, H. and J. Bolte (2009). "On the convergence of the proximal algorithm for nonsmooth functions involving analytic features". *Mathematical Programming*, vol. 116, no. 1, pp. 5–16.

Attouch, H., J. Bolte, and B. F. Svaiter (2013). "Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods". *Mathematical Programming*, vol. 137, no. 1, pp. 91–129.

Bolte, J., S. Sabach, and M. Teboulle (2014). "Proximal alternating linearized minimization for nonconvex and nonsmooth problems". *Mathematical Programming, Series A*, vol. 146, pp. 459–494.

Example 2.23: Elastic net (Zou and Hastie, 2005)

Suppose we have two duplicate features $\mathbf{x}_1 = \mathbf{x}_2$, corresponding to weights w_1 and w_2. It is not difficult to see that if $\mathbf{w}_\star$ is an optimal solution for Lasso (see Example 2.3), then any w_1 and w_2 with $w_1 + w_2 = w_{\star,1} + w_{\star,2}$ and $|w_1| + |w_2| = |w_1 + w_2|$ will remain optimal. On the other hand, the elastic net

$$\min_{\mathbf{w}} \tfrac{1}{n}\|\mathsf{X}^\top \mathbf{w} - \mathbf{y}\|_2^2 + \lambda\|\mathbf{w}\|_1 + \tfrac{\gamma}{2}\|\mathbf{w}\|_2^2$$

will select both features and give them even weights. Here we have two choices:

- Set $\ell = \frac{1}{n}\|\mathsf{X}^\top \mathbf{w} - \mathbf{y}\|_2^2 + \frac{\gamma}{2}\|\mathbf{w}\|_2^2$ and $r(\mathbf{w}) = \lambda\|\mathbf{w}\|_1$.
- Set $\ell = \frac{1}{n}\|\mathsf{X}^\top \mathbf{w} - \mathbf{y}\|_2^2$ and $r(\mathbf{w}) = \lambda\|\mathbf{w}\|_1 + \frac{\gamma}{2}\|\mathbf{w}\|_2^2$.

The former has a bigger L-smoothness parameter (a difference of γ) while the latter requires computing a more complicated proximal map. Fortunately, it can be shown from brute-force computation that

$$\mathrm{P}^\eta_{\lambda\|\cdot\|_1 + \frac{\gamma}{2}\|\cdot\|_2^2}(\mathbf{w}) = \mathrm{P}^\eta_{\frac{\gamma}{2}\|\cdot\|_2^2}\left(\mathrm{P}^\eta_{\lambda\|\cdot\|_1}(\mathbf{w})\right).$$

A general sufficient condition, along with some illustrating examples, was derived by Yu (2013) for the decomposition to hold:

$$\mathrm{P}^\eta_{f+g} = \mathrm{P}^\eta_f \circ \mathrm{P}^\eta_g.$$

Zou, H. and T. Hastie (2005). "Regularization and variable selection via the elastic net". *Journal of the Royal Statistical Society, Series B*, vol. 67, pp. 301–320.

Yu, Y.-L. (2013). "On Decomposing the Proximal Map". In: *Advances in Neural Information Processing Systems 27 (NIPS)*.

Alert 2.24: Devil is in the details

The proximal gradient Algorithm 2.17 allows us to deal with non-smooth functions (and potentially use bigger step size by pushing functions to r, as in Example 2.23). However, we remind that this is built on the premise that the proximal map of r can be computed cheaply, which is itself a non-smooth minimization problem! There has been a lot of work on how to compute the proximal map of various functions in ML, even to this day, see for example Beck and Hallak (2018) and Latorre et al. (2020).

Beck, A. and N. Hallak (2018). "Proximal Mapping for Symmetric Penalty and Sparsity". *SIAM Journal on Optimization*, vol. 28, no. 1, pp. 496–527.

Latorre, F., P. Rolland, N. Hallak, and V. Cevher (2020). "Efficient Proximal Mapping of the 1-path-norm of Shallow Networks". In: *ICML*.