4 Proximal Gradient

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

Alert 4.1: Convention
Gray boxes are not required hence can be omitted for unenthusiastic readers.
This note is likely to be updated again soon.

Definition 4.2: Problem
In this lecture we study the following problem:
\[ \inf_{w \in \mathbb{R}^d} f(w), \text{ where } f(w) = \ell(w) + r(w) \]
is the sum of a smooth function \( \ell \) 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 4.3: Lasso (Tibshirani 1996)
We continue our discussion of linear regression in Example 2.3. Suppose we know the optimal solution \( w \) is sparse, i.e., only a few entries in \( w \) are nonzero, how do we exploit this information and make our estimation more efficient? Obviously, if an oracle knew which entries of \( w \) 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 \( \ell_1 \)-norm regularizer on \( w \):
\[
\min_w \frac{1}{n} \| X^T w - y \|_2^2 + \lambda \|w\|_1,
\]
where \( \lambda > 0 \) is some tuning hyperparameter. The \( \ell_1 \) norm \( \| \cdot \|_1 \) is incorporated to induce sparsity in the minimizer \( w \), a claim that will become clear once we see the algorithm.

We note that even when the optimal solution \( w \) 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 present \( w \), which is typical in embedded/mobile system and known as model compression/quantization.


Definition 4.4: Proximal map and Moreau envelope (Moreau 1965)
The proximal map and Moreau envelope of a (closed) function \( f \) is defined respectively as:
\[
P_f^\eta(z) := \arg\min_w \frac{1}{2\eta} \|w - z\|_2^2 + f(w), \quad M_f^\eta(z) := \inf_w \frac{1}{2\eta} \|w - z\|_2^2 + f(w),
\]
where \( \eta > 0 \) is a smoothing parameter. We define the proximal threshold
\[
\tau_f := \frac{1}{\left( -2 \cdot \lim_{\|w\| \to \infty} \frac{f(w)}{\|w\|_2^2} \right)_+}, \text{ where } 1/0 := \infty.
\]
Then, for all \( \eta \in [0, \tau_f] \), it follows from (a slight generalization of) Theorem 1.33 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.


Exercise 4.5: Moreau envelope preserves minimizer

Prove that

- $\forall \eta > 0$, $M^\eta_f \leq f$
- $\forall \eta > 0$, $\inf f = \inf M^\eta_f$ and $\arg\min f = \arg\min M^\eta_f$.
- $\forall w$, $M^\eta_f(w) \to f(w)$ as $\eta \downarrow 0$.

(The last one may be a bit difficult.)

Moreover, it can be shown that the Moreau envelope $M^\eta_f$ 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.


Alert 4.6: Notation

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

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

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

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

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

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

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 $w$, then $w$ is trivially the (global) minimizer and there is nothing to minimize.

Example 4.7: Euclidean projection as proximal map

With the above notation, it is clear that

$$P_C(w) = P_{\iota_C}(w)$$
Example 4.8: 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 \( \lambda \)-Lipschitz continuous (assuming \( \lambda \geq 0 \)):

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

Let us compute the proximal map:

\[
P_\eta^f(s) = \underset{t}{\operatorname{argmin}} \frac{1}{2\eta} (s - t)^2 + \lambda t_+ = \begin{cases} 
\frac{1}{2\eta} (s - t)^2 + \lambda t, & t \geq 0 \\
\frac{1}{2\eta} (s - t)^2, & t \leq 0
\end{cases}
\]

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:

\[
P_\eta^f(s) = \begin{cases} 
s - \lambda \eta, & \text{if } s \geq \lambda \eta \\
 s \wedge 0, & \text{if } s \leq \lambda \eta
\end{cases}
\]

with minimum value

\[
M_\eta^f(s) = \begin{cases} 
\lambda s - \frac{\eta \lambda^2}{2}, & \text{if } s \geq \lambda \eta \\
\frac{1}{2\eta}(s_+)^2, & \text{if } s \leq \lambda \eta
\end{cases}
\]

Exercise 4.9: \( L \)-smoothness and uniform approximation of Moreau’s envelope

Prove the following:

- The Moreau envelope in Example 4.8 is differentiable and \( \frac{1}{\eta} \)-smooth.
- The following uniform approximation holds:

\[
\sup_{s \in \mathbb{R}} |M_\eta^f(s) - \lambda(s)_+| \leq \frac{\eta \lambda^2}{2},
\]

where recall that the positive part \( \lambda s_+ \) is \( \lambda \)-Lipschitz continuous.

Example 4.10: Soft-shrinkage

In your assignment you are going to derive the proximal map for the \( \ell_1 \) 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 get truncated to 0, i.e. gaining sparsity! Interestingly, this sparsity-promoting property of the \( \ell_1 \) 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 4.11: If it works for the $\ell_2$ 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 3.9 first.

- Any norm is not differentiable at the origin.

- Unlike the $\ell_2$ norm case, squaring the $\ell_1$ norm does not make it differentiable. This is a laughable mistake that is not uncommon in reality.

**Example 4.12: Sparsemax**

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

$$\text{softmax}(w) := \arg\min_{z \in \Delta} \left< w, z \right> + \lambda \sum_j z_j \log z_j$$

$$\propto \exp(-w/\lambda),$$

where $\Delta := \{ z \in \mathbb{R}^d_+ : 1^T z = 1 \}$ is the simplex (so that $z \in \Delta$ 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

$$\text{sparsemax}(w) := \arg\min_{z \in \Delta} \left< w, z \right> + \frac{\lambda}{2} \| z \|_2^2$$

$$= P_{\Delta}(-w/\lambda).$$

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; Martins et al. 2020) and references therein for applications in attention and transformers, and (e.g. Muzellec et al. 2017; Blondel et al. 2018; Dessein et al. 2018) for applications in optimal transportation.

**Remark 4.13: More on proximal map**

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

Algorithm 4.14: Proximal point algorithm (PPA) (Martinet 1970; Rockafellar 1976)

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

<table>
<thead>
<tr>
<th>Algorithm: Proximal Point Algorithm (PPA)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> $w_0$</td>
</tr>
<tr>
<td>1 for $t = 0, 1, \ldots$ do</td>
</tr>
<tr>
<td>2 choose step size $\eta_t$</td>
</tr>
<tr>
<td>3 $w_{t+1} = P_{\eta_t}^{f}(w_t) = \arg\min_w \frac{1}{2\eta_t} |w - w|^2 + f(w)$ // proximal step</td>
</tr>
</tbody>
</table>

We will not present any convergence property of PPA, except pointing 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:

$$w_{t+1} = w_t - \eta_t \nabla f(w_{t+1}).$$


Algorithm 4.15: Proximal gradient algorithm (PG) (Fukushima and Mine 1981)

<table>
<thead>
<tr>
<th>Algorithm: Proximal Gradient (PG) for composite minimization</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong> $w_0 \in \text{dom } f$</td>
</tr>
<tr>
<td>1 for $t = 0, 1, \ldots$ do</td>
</tr>
<tr>
<td>2 choose step size $\eta_t$</td>
</tr>
<tr>
<td>3 $z_t = w_t - \eta_t \nabla \ell(w_t)$ // gradient step w.r.t. $\ell$</td>
</tr>
<tr>
<td>4 $w_{t+1} = P_{\eta_t}^{\ell}(z_t) = \arg\min_w \frac{1}{2\eta_t} |z_t - w|^2 + r(w)$ // proximal step w.r.t. $r$</td>
</tr>
</tbody>
</table>

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. $\ell$. 

• In the second step, we ignore the smooth function $\ell$ and perform the proximal map w.r.t. $r$ (i.e. backward gradient update).

Obviously, the efficiency of PG largely depends on how fast we can compute the proximal map $P_{\eta}^r(w)$, 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 $\ell$ by its linearization:

$$w_{t+1} = \arg\min_w \ell(w_t) + \langle w - w_t, \nabla \ell(w_t) \rangle + \frac{1}{2\eta_t} \| w - w_t \|^2 + r(w).$$  \hfill (4.2)


**Definition 4.16: Bregman divergence (Bregman 1967)**

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

$$D_f(z; w) = f(z) - f(w) - \langle z - w, \nabla f(w) \rangle \geq 0,$$

where the inequality follows from convexity. In general, the Bregman divergence is not symmetric, i.e. $D_f(z; w) \neq D_f(w; z)$.


**Example 4.17: Bregman divergence of Bregman divergence**

Let $f(w) = \frac{1}{2} \| w - w_0 \|^2$ for any $w_0$. Then, we easily verify that $D_f(z; w) = \frac{1}{2} \| z - w \|^2$, 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 $f(w) = D_h(w; w_0)$, then

$$D_f(z; w) = D_h(z; w_0) - D_h(w; w_0) - \langle z - w, \nabla h(w) - \nabla h(w_0) \rangle = D_h(z; w).$$


**Remark 4.18: Proximal map with Bregman divergence**

We can also replace the quadratic function in the definition of proximal map, see (4.1), with a general Bregman divergence, see (Kan and Song 2012; Bauschke et al. 2018; Soueycatt et al. 2020), or with a more general norm (Lescarret 1967; Combettes and Reyes 2013; Bacák and Kohlenbach 2018).


Remark 4.19: $L$-smoothness through Bregman divergence

We note that the $L$-smoothness Definition 2.11 can now be written compactly as

$$D_f(z; w) \leq \frac{L}{2} \|w - z\|^2,$$

which motivates us to replace the quadratic on the right-hand side with a general Bregman divergence as well. See (Bauschke et al. 2017; Bolte et al. 2018; Teboulle 2018; Bauschke et al. 2019).


Proposition 4.20: 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,

$$w_\star \in \arg\min_{w} \ell(w) + r(w) \iff \forall w, \; \ell(w) + r(w) \geq \ell(w_\star) + r(w_\star) + D_\ell(w; w_\star).$$

Proof. $\Leftarrow$: By convexity of $\ell$ we know the Bregman divergence $D_\ell(w; w_\star) \geq 0$, hence $w_\star$ is clearly a (global) minimizer.

$\Rightarrow$: Since $w_\star$ is a minimizer, we have $0 \in \partial(\ell + r)(w_\star) = \nabla \ell(w_\star) + \partial r(w_\star)$ where the second equality follows from convexity and continuous differentiability. Using convexity of $r$ we have $r(w) \geq r(w_\star) + \langle w - w_\star, \partial r(w_\star) \rangle$. Therefore, $\ell(w) + r(w) \geq \ell(w_\star) + r(w_\star) + \ell(w) - \ell(w_\star) + \langle w - w_\star, -\nabla \ell(w_\star) \rangle \geq \ell(w_\star) + r(w_\star) + D_\ell(w; w_\star). \Box$


Theorem 4.21: Convergence of PG in function value (Beck and Teboulle 2009; Tseng 2008)

Let $\ell : \mathbb{R}^d \to \mathbb{R}$ be convex and $L$-smooth, $r : \mathbb{R} \to \mathbb{R} \cup \{\infty\}$ be closed and convex, and $\eta_t$ is chosen so that (4.3) below holds. Then, for all $w$ and $t \geq 1$, the sequence $\{w_t\}$ generated by Algorithm 4.15 satisfy:

$$f(w_t) \leq f(w) + \frac{\|w - w_0\|^2}{2t\eta_t}, \quad \text{where} \quad \bar{\eta}_t := \frac{1}{t} \sum_{s=0}^{t-1} \eta_s.$$

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

From the definition of $w_{t+1}$ in (4.2) and Proposition 4.20 we know for any $w$:

$$f(w_{t+1}) = \ell(w_{t+1}) + r(w_{t+1}) \leq \ell(w_t) + r(w_{t+1}) - \eta_t \langle w_{t+1} - w_t, \nabla \ell(w_t) \rangle + \frac{1}{2\eta_t} \|w_{t+1} - w_t\|^2 + r(w_{t+1}) \tag{4.3}$$
\[
\ell(w_t) + \langle w - w_t, \nabla \ell(w_t) \rangle + \frac{1}{2\eta_t} \|w - w_t\|^2 + r(w) - \frac{1}{2\eta_t} \|w - w_{t+1}\|^2
\leq \ell(w) + r(w) + \frac{1}{2\eta_t} \|w - w_t\|^2 - \frac{1}{2\eta_t} \|w - w_{t+1}\|^2,
\]
where the last inequality is due to the convexity of \( \ell \). Take \( w = w_t \) we see \( f(w_{t+1}) \leq f(w_t) \), i.e., the algorithm is descending. Summing from \( t = 0 \) to \( t = T - 1 \):
\[
T\eta_T \cdot |f(w_T) - f(w)| \leq \sum_{t=0}^{T-1} \eta_t [f(w_{t+1}) - f(w)] \leq \frac{1}{2} \|w - w_0\|_2^2,
\]
where we have chosen \( \eta_t \equiv 1/L \) 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 \( w_0 \) 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 (4.3) is not vacuous. If we do not know \( L \) in advance, we can apply backtracking (see Remark 2.20) until (4.3) is satisfied.


**Remark 4.22: Convergence of iterates**

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


**Example 4.23: Elastic net (Zou and Hastie 2005)**

Suppose we have two duplicate features \( x_1 = x_2 \), corresponding to weights \( w_1 \) and \( w_2 \). It is not difficult to see that if \( w_\star \) is an optimal solution for Lasso (see Example 4.3), 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

$$\min \frac{1}{n} \|Xw - y\|_2^2 + \lambda \|w\|_1 + \frac{\gamma}{2} \|w\|_2^2$$

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

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

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

$$P_{\eta \lambda \|\cdot\|_1 + \frac{\gamma}{2} \|\cdot\|_2^2}(w) = P_{\eta \lambda \|\cdot\|_1}(P_{\frac{\gamma}{2} \|\cdot\|_2^2}(w)).$$

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

$$P_{f+g} = P_f \circ P_g.$$


---

**Alert 4.24: Devil is in the details**

The proximal gradient algorithm allows us to deal with non-smooth functions (and potentially use bigger step size by pushing functions to $r$, as in Example 4.23). 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 e.g. Beck and Hallak 2018; Latorre et al. 2012).


Latorre, Fabian, Paul Rolland, Nadav Hallak, and Volkan Cevher (2012). “Efficient Proximal Mapping of the 1-pathnorm of Shallow Networks”. In: ICML.