Portland State
UNIVERSITY

Proximal Mappings & Optimization

Honors Paper

George Blikas

June 2, 2016

Contents

Abstract

The purpose of this lithograph is to illustrate some of the advantages of proximal mappings, and their role in convex optimization algorithms. The main body of this paper focuses on proximal algorithms. In addition, we state and prove some elementary properties of proximal mappings, show some of the applications of the proximal mapping, and elaborate on the relationship between the proximal operator and convex optimization.

A list of some of the commonly used terminology and notation can be found in the appendices on page 21.

1 Motivation

It is clear that the modern world requires efficient algorithms to solve optimization problems. As data bases grow larger, it is becoming more of a challenge to efficiently search, and predict data behaviour. The proximal mapping is a helpful tool in overcoming these obstacles.

The so-called *proximal algorithms* are exceedingly useful for solving large-scale, non-smooth, or constrained problems. Whereas many other algorithms require several 'low-level' operations, the evaluation of *proximal operator* functions can lead to closed-form solutions to convex optimization problems [12].

Given a convex function f, the proximal operator of f, evaluated at a point, describes a trade-off between minimizing f and being near said point. Even though we do not directly deal with f directly, the proximal map has some very desirable characteristics. For instance, it is separable across variables and minimizers of the proximal mapping of f are minimizers of f. Figure 1 is a graphical depiction of this. In the next section, we present some standard facts about the proximal

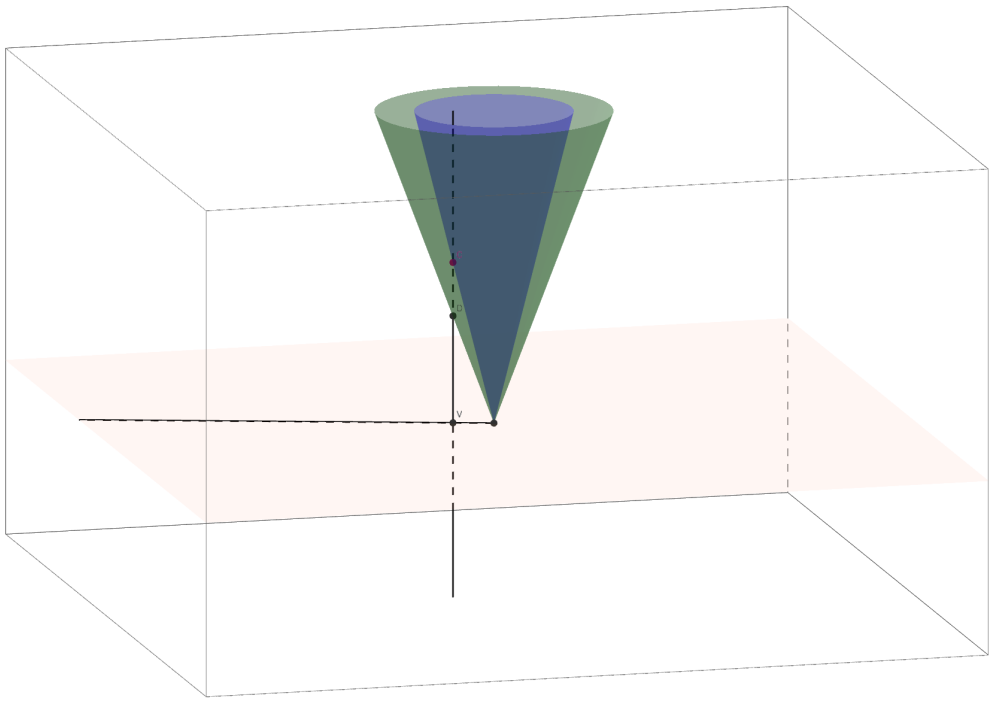


Figure 1: An illustration depicting the difference between minimizing f (green cone) and being close to v. The blue cone represents the proximal mapping of f. While the output value for this iteration would be D, a trade-off must be made; we move to C.

operator of f.

2 Important Concepts

All of our efforts will be aimed at solving–at least–the unconstrained problem

$$\underset{x\in\operatorname{dom} f}{\text{minimize}}\, f(x),$$

where it is assumed that $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ is a closed, proper, convex function, unless otherwise stated. Further, although proximal operator theory can be established in any Hilbert space, we only deal with $\mathbb{R}^n$.

In understanding the proximal operator of a function f, it is necessary to describe its Moreau Envelope and Fenchel conjugate. Both these concepts are so fundamental in convex analysis that it is only natural we present them first.

This section mentions some of the core concepts needed for proximal operator theory.

§2.1 Fenchel Conjugates and the Moreau Envelope

As these concepts are at the foundation of convex analysis and convex optimizations, they deserve special attention. Obviously, we cannot prove or mention all the details necessary for our analysis, here. So, we first present the definition of the *Fenchel Conjugate* of a function f, then we explore the *Moreau Envelope*, or *Moreau-Yosida Regularization*, of f.

§2.1.1 Fenchel Conjugate

At the simplest level, the Fenchel conjugate of a function f describes a connection between the dual of a space $\mathcal{H}$, and itself. We provide the reader with some of the intuition for the Fenchel conjugate of a function f.

Although one can find the definition in the appendix, we restate it here for completeness.

Definition

Fenchel Conjugate:
Given a function $f : \mathbb{R}^n \to (-\infty, \infty]$, the *Fenchel Conjugate*, $f^* : \mathbb{R}^n \to [-\infty, \infty]$, of f is defined as the following:

$$\begin{aligned} f^*(v) &= \sup\{\langle v, x\rangle - f(x) : x \in \mathbb{R}^n\} \\ &= \sup\{\langle v, x\rangle - f(x) : x \in \operatorname{dom} f\} \end{aligned}$$

It is well-known that all elements of $\mathcal{H}^* = (\mathbb{R}^n)^*$ are of the form $\langle x, y\rangle$, for some $y \in \mathcal{H}$, and all $x \in \mathcal{H}$ [10]. So, considering the definition above, we can see that f^* is actually the supremum of a set of affine functions: $\langle \cdot, x\rangle - f(x)$. It follows from the fact that a function is convex iff its epigraph is convex that f^* is convex and lower semi-continuous.[1]

[1] For proof, see the appendix on page 24

For a more geometric representation of this concept, the following image is a numerical approximation for the Fenchel conjugate of $f(x) = x^2 + 1$; fig 2. Each dot represent $f^*(v)$. That is, each dot is precisely the supremum over all $\langle \cdot, x\rangle - f(x)$, $x \in \mathbb{R}$; the edge connecting the vertexes is an approximation of f^*.

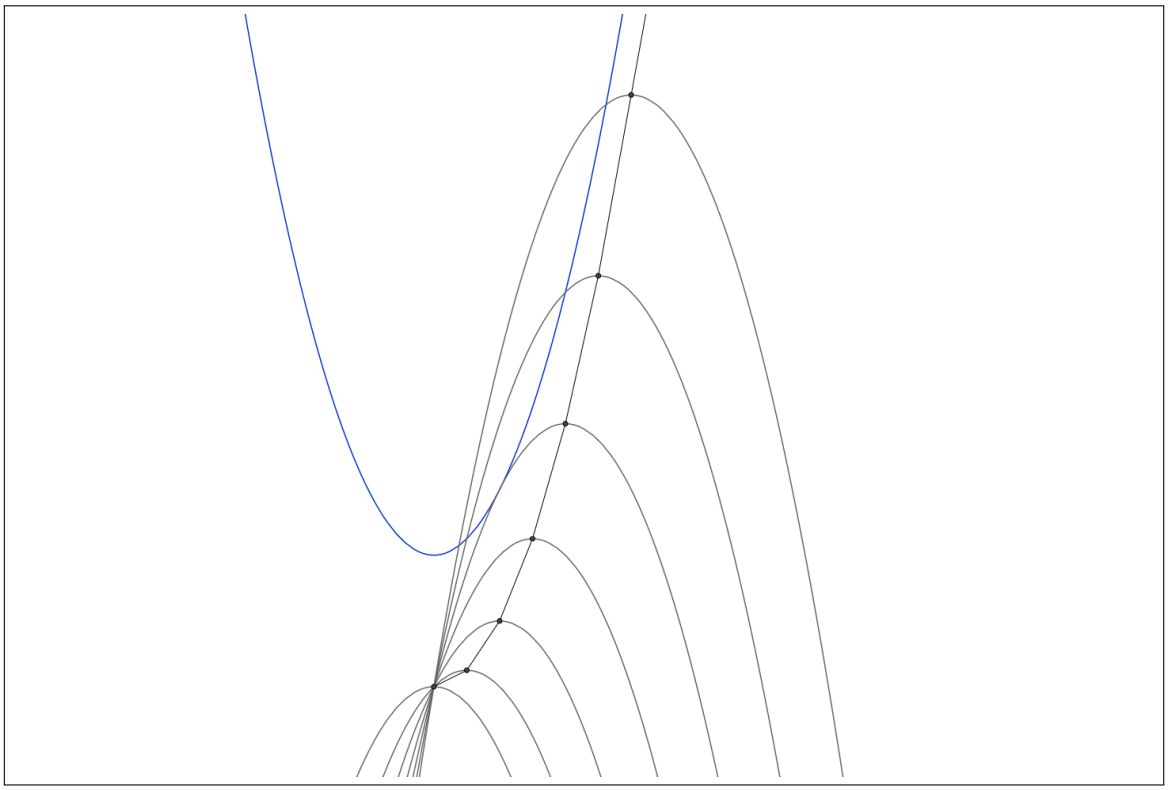


Figure 2: A numerical approximation for the Fenchel conjugate of $f(x) = x^2 + 1$. Points were only sample to the right hand side of f.

Figure 2 also illustrates an immediate consequence elicited from the convexity of f^*, as described above: $f(x) + f^*(u) \geq \langle x, u\rangle$, better known as the *Fenchel-Young Inequality* [2]. A useful note to keep in mind is that after re-arranging terms, we obtain

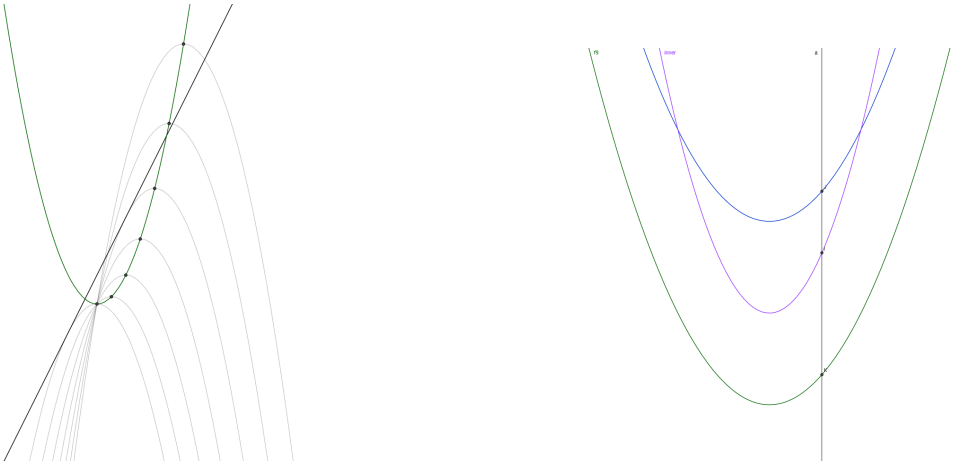
$$f(x) \geq \langle x, u\rangle - f^*(x)$$

for each $x, v \in \mathbb{R}^n$. From this we can see that $f(x)$ dominates a DC function.

The last popular result that we leave the reader with assumes that f be differentiable at some $\bar{x} \in \mathbb{R}^n$. In this case, we have

$$f^*(\nabla f(\bar{x})) = \langle \bar{x}, \nabla f(\bar{x})\rangle - f(\bar{x})$$

[2] Cf. Above.



(a) Green: f^* as computed above.

(b) Green: f^*, Violet: $\langle x, \nabla f(x) \rangle$, Blue: f.

Figure 3: Two visualizations of the equality, above.

Figure 3 is graphical representation of the previous equality.

§2.1.2 Moreau Envelope

The Moreau envelope is precursor to the proximal mapping. There is clearly a connection between the Moreau Envelope, of a convex function f, and the scaled proximal operator on f. Precisely,

Definition

Let $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ be a convex function. Then, the *Moreau Envelope* of f, $f_\mu : \mathbb{R}^n \to \bar{\mathbb{R}}$, is given by the following:

$$f_\mu(x) = \inf_y \left\{ f(y) + \frac{1}{2\mu} ||x - y||_2^2 \right\}$$

for $\mu > 0$. f_μ is often denoted $M_{\mu f}$.

and, we will see that the proximal operator uses some condition for attainment. That is, under certain assumptions, f_μ actually attains its infimum. Later, we will show that this is the case.

As illustrations of convex optimizations are indispensable, we present a visualization of $f_{\mu=1}$, $f(x) = x^2 + 1$; fig 4. As one observes, $f(x) \approx f_1(x)$. However, by varying μ, one can obtain successively better approximations of $f(x)$. See figure 5.

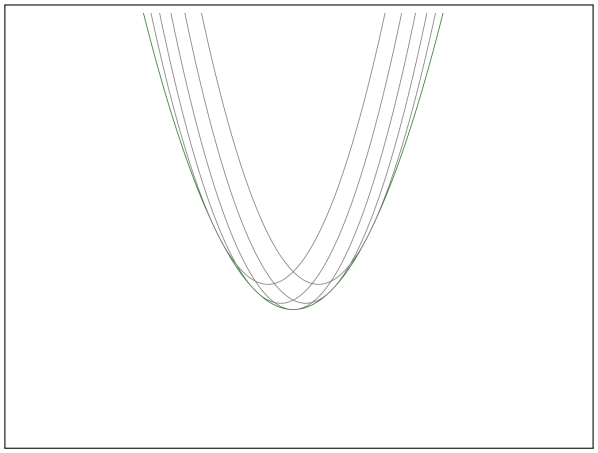


Figure 4: A numerical approximation for the Moreau envelope of $f(x) = x^2 + 1$, $\mu = 1$.

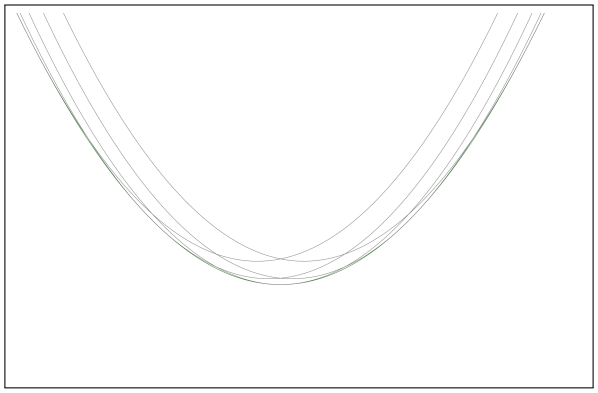


Figure 5: A numerical approximation for the Moreau envelope of $f(x) = x^2 + 1$, $\mu = 10$.

3 The Proximal Operator

At the heart of this paper is the proximal operator and the concept of convexity. In this section, we give the definition of the proximal operator of f, and illustrate a connection between it and the Moreau envelope of f.

§3.1 Definition

To aid us throughout this paper, we present the definition of the proximal operator[3] of f:

$$\underset{f}{\mathbf{prox}}(v) := \underset{x\in\mathbb{R}^n}{\arg\min}\left\{f(x) + \frac{1}{2}||x-v||_2^2\right\}$$

In later sections, we will elaborate on the development of the proximal operator of f, show that $\mathbf{prox}_f$ is strongly convex, and that it has a unique minimizer, $\bar{v} \in \mathbb{R}^n$.

As is often the case, one typically considers a scaled variant on $\mathbf{prox}_f$. This is given by the following:

Definition

Let $\lambda \in \mathbb{R}$, such that $\lambda > 0$. Then the scaled *proximal operator*, $\mathbf{prox}_f : \mathbb{R}^n \to \mathbb{R}$, of f is given by

$$\underset{\lambda f}{\mathbf{prox}}(v) = \underset{x\in\mathbb{R}^n}{\arg\min}\left\{f(x) + \frac{1}{2\lambda}||x-v||_2^2\right\}$$

As a natural extension of the unscaled operator, we can think of λ as a weight penalty (reward) for being close to x. Thus, in iterative evaluations of proximal operators, λ can be carefully chosen to facilitate convergence to an optimal solution.

§3.2 Interpretations and Connections

The proximal operator of f can be interpreted from many different viewpoints. However, maybe the most important one is the relationship that the proximal operator shares with the Moreau envelope; it can be shown that

$$\nabla M_{\mu f}(x) = \frac{1}{\mu}(x - \underset{\mu f}{\mathbf{prox}}(x))$$

Rearranging, we obtain

$$\underset{\mu f}{\mathbf{prox}}(x) = x - \mu\nabla M_{\mu f}(x)$$

Further, the Moreau decomposition gives

$$\underset{f}{\mathbf{prox}}(x) = \nabla M_{f^*}(x)$$

[3]This definition can also be found in the appendix 21

Thus, if f is convex, then $\mathbf{prox}_f$ is the value for which M_f attains its infimum [6]. In addition, the above shows that we can consider the scaled proximal operator of f as a gradient step for minimizing the Moreau envelope of f [12].

As an extension to what we have just mentioned, we move on to the main purpose of this article: proximal algorithms.

4 Proximal Algorithms

As the name suggests, proximal algorithms are used to solve convex optimization problems. For an algorithm to be a *proximal algorithm*, it must evaluate a proximal operator of some function at each iteration. Some common examples of this are the proximal minimization algorithms, proximal gradient methods, and under certain assumptions, as we will show, the proximal operator can approximate a general gradient method.

Some things should be kept in mind, when considering proximal algorithms. First of all, as with any algorithm, a proximal algorithm is only useful if each proximal operator can be computed efficiently. This can be one of the most limiting factors in the application of proximal algorithms. And secondly, although this is generally the case, we will see that proximal algorithms are extremely useful in large-scale operations [12].

Through the rest of the article, we use the notation $x^{(k)}$ for the k-th iterate of $x \in \mathbb{R}^n$, and in general, any notation of the form $< \text{expression} >^{(k)}$. The main portion of this paper is the proof of the convergence rate of the proximal gradient method. However, we first mention the most introductory example of a proximal algorithms.

§4.1 Examples

§4.1.1 Proximal Minimization Algorithm

The most introductory example of a proximal algorithm is the *proximal minimization algorithm*, or *proximal point algorithm*. This is a very simple method for determining optimal values. The recursion is given in the following pseudo-code:

Algorithm 1 Proximal Point Algorithm

1: Pick $x^{(0)} \in \mathbb{R}^n$, $\mu \geq 0$, and $N \in \mathbb{N}$
2:
3: **for** k = 1, 2, ..., N **do**
4: $\quad x^{(k+1)} := \mathbf{prox}_{\mu f}(x^{(k)})$
5:
6: Output: $x^{(N+1)}$

One thing to keep in mind with algorithm 1 is that it introduces unnecessary computation in some cases. That is, we are asked to compute f with the addition of some quadratic function. However, algorithm 1 is the most useful proximal algorithm to keep in mind, because at some point all proximal algorithms must do something similar, as in line 4.

§4.1.2 Proximal Gradient Method

The *proximal gradient method* is an optimization technique that relies on our objective function having a decomposition as

$$\underset{x \in \mathbb{R}^n}{\text{minimize}}\, f(x) = h(x) + g(x),$$

where $h : \mathbb{R}^n \to \mathbb{R}$, is a lower semi-continuous convex function and $g : \mathbb{R}^n \to \bar{\mathbb{R}}$, convex differentiable, such that ∇g is $L - Lipschitz$.

The proximal gradient method, and those similar to it, such as the accelerated version [15] are the most popular. Their wide range application merits their investigation. As stated previously, we prove that the iteration defined by the proximal gradient method,

$$x^{(k+1)} := \underset{\mu^{(k)}h}{\mathbf{prox}} \left(x^{(k)} - \mu^{(k)} \nabla g(x^{(k)}) \right)$$

converges to an optimal solution:

Throughout the proof, we will use the notation $f \in \mathcal{O}(z(x))$ instead of the more conventional $f(x) = \mathcal{O}(z(x))$. By $f \in \mathcal{O}(z(x))$ we mean that f is a member of a set which consists of all functions, g, for which

$$\lim_{x\to\infty} |g(x)| \leq C \lim_{x\to\infty} |z(x)|$$

for some $C \in \mathbb{R}$, $C \geq 0$. If we wish to be specific about such a constant C, we will write $\mathcal{O}_C(z(x))$.

Suppose that the optimal value of f is $f(x^*)$, and that $t^{(k)}$ is chosen through line search at each iteration.

Proposition

If $t^{(k)} \in (0, 1/L]$ for each $k \in \mathbb{N}$, then

$$|f(x^{(k)}) - f(x^*)| \in \mathcal{O}(1/n)$$

To help facilitate the proof, we first define a new function in terms of the scaled proximal operator of h, and prove two lemmas:

Define a new function by

$$G_t(x) := \frac{1}{t}\big(x - \underset{th}{\mathbf{prox}}(x - t\nabla g(x))\big),$$

such that $t > 0$. It should be noted that

$$x^{(k)} = x^{(k-1)} - t^{(k)} G_{t^{(k)}}(x^{(k-1)})$$

Indeed, upon substitution, we have the following:

$$\begin{aligned} x^{(k)} &= x^{(k-1)} - t^{(k)}(G_{t^{(k)}}(x^{(k-1)}) \\ &= x^{(k-1)} - t^{(k)}\Big(\frac{1}{t^{(k)}}(x^{(k-1)} - \underset{t^{(k)}h}{\mathbf{prox}}\left(x^{(k-1)} - t^{(k)}\nabla g(x^{(k-1)})\right)\Big) \\ &= x^{(k-1)} - x^{(k-1)} + \underset{t^{(k)}h}{\mathbf{prox}}\left(x^{(k-1)} - t^{(k)}\nabla g(x^{(k-1)})\right) \\ &= \underset{t^{(k)}h}{\mathbf{prox}}\left(x^{(k-1)} - t^{(k)}\nabla g(x^{(k-1)})\right) \end{aligned}$$

The advantage to $G_t(x)$ is that it does away with the proximal operator.
From the characterization of subgradient,

$$G_t(x) - \nabla g(x) \in \partial h(x - tG_t(x))$$

Lemma

Quadratic Upper Bound for L-Lipshcitz Functions

$$g(y) \leq g(x) + \nabla g(x)^T(y-x) + \frac{L||y-x||^2}{2}$$

where g is L-Lipshcitz and for $x \in \mathbb{R}^n$, all $y \in \mathbb{R}^n$.

To prove this, let $v = y - x$ and consider the mapping $\phi : \mathbb{R}^n \to \bar{\mathbb{R}}$, given by

$$\phi(t) = g(x + tv)$$

Then, $\phi'(t) = \nabla g(x+tv)^T v$ and we have the following:

$$\begin{aligned} g(y) &= g(x) + \int_0^1 \phi'(t)dt \\ &= g(x) + \int_0^1 \nabla g(x+tv)^T v dt \\ &= g(x) + \int_0^1 \nabla g(x+tv)^T v dt + \nabla g(x)^T v - \int_0^1 \nabla g(x)^T v dt \\ &= g(x) + \nabla g(x)^T v + \int_0^1 [\nabla g(x+tv)^T v - \nabla g(x)^T] v dt \\ &\leq g(x) + \nabla g(x)^T v + \int_0^1 ||\nabla g(x+tv)^T v - \nabla g(x)^T||_2 ||v||_2 dt \\ &\leq g(x) + \nabla g(x)^T v + \int_0^1 L \cdot t \cdot ||v||_2^2 dt \\ &= g(x) + \nabla g(x)^T v + \frac{L \cdot ||v||_2^2}{2} \end{aligned}$$

This completes the proof of the lemma.

Lemma

$$f(x^{(k+1)}) \leq f(z) + \langle G_{t^{(k)}}(x^{(k)}), x^{(k)} - z\rangle - \frac{t^{(k)}}{2}||G_{t^{(k)}}(x^{(k)})||_2^2$$

for $z \in \mathbb{R}^n$.

To show this, consider the previous lemma and let $y = x - tG_t(x)$. Then we have

$$g(x - tG_t(x)) \leq g(x) - t\nabla g(x)^T G_t(x) + \frac{t^2 \cdot L}{2}||G_t(x)||_2^2$$

Now, $0 \leq t \leq 1/L$, implies $t^2 \cdot L \leq t$ and so,

$$g(x - tG_t(x)) \leq g(x) - t\nabla g(x)^T G_t(x) + \frac{t}{2}||G_t(x)||_2^2 \tag{1}$$

To complete the proof, let $v := G_t(x) - \nabla g(x)$. Then,

$$\begin{aligned} f(x - tG_t(x)) &\leq g(x) - t\nabla g(x)^T G_t(x) + \frac{t}{2}||G_t(x)||_2^2 + h(x - tG_t(x)) && (2)\\ &\leq g(z) + \nabla g(x)^T(x - z) - t\nabla g(x)^T G_t(x) + && (3)\\ &\qquad \frac{t}{2}||G_t(x)||_2^2 + h(z) + v^T(x - z - tG_t(x)) && (4)\\ &= g(z) + h(z) + G_t(x)^T(x - z) - \frac{t}{2}||G_t(x)||_2^2 && (5) \end{aligned}$$

Note, line (1) follows from the previous inequality that was shown. (2) follows from the convexity of g and h and the fact that $v \in \partial h(x - tG_t(x))$. The above lemma follows directly as a consequence of this definition, and the lemma is complete.

We are now ready to give the proof.

Proof. Let $z = x$. Then,

$$f(x^{(k+1)}) \leq f(x) - \frac{t^{(k)}}{2}||G_t(x)||_2^2$$

For all $x \in \operatorname{dom} f$. And, letting $z = x^*$ (the optimal solution of f) we have

$$\begin{aligned} f(x^{(k+1)}) - f(x^*) &\leq G_{t^{(k)}}(x^{(k)})(x^{(k)} - x^*) - \frac{t^{(k)}}{2}||G_{t^{(k)}}(x^{(k)})||_2^2\\ &\leq \frac{1}{2 \cdot t^{(k)}}\Big(||x^{(k)} - x^*||_2^2 - ||x^{(k+1)} - x^*||_2^2\Big) \end{aligned}$$

To finish off the proof, we add up over $n \in \mathbb{N}$ iterations i.e. n terms:

$$\begin{aligned}\sum_{i=0}^{n}[f(x^{(i+1)}) - f(x^*)] &= \sum_{i=0}^{n} \frac{1}{2 \cdot t^{(k)}}\Big(||x^{(k)} - x^*||_2^2 - ||x^{(k+1)} - x^*||_2^2\Big) \\ &\leq \frac{1}{2 \cdot t^{(1)}}||x^{(0)} - x^*||_2^2\end{aligned}$$

Since $f(x^{(k+1)}) \leq f(x^{(k)})$, we have that on the k-th iterate,

$$\begin{aligned} k \cdot [f(x^{(k)}) - f(x^*)] &\leq \frac{1}{2 \cdot t^{(1)}}||x^{(0)} - x^*||_2^2 \\ \iff f(x^{(k)}) - f(x^*) &\leq \frac{1}{2 \cdot t^{(1)} \cdot k}||x^{(0)} - x^*||_2^2 \\ \implies |f(x^{(k)}) - f(x^*)| &\leq \Big|\frac{1}{2 \cdot t^{(1)} \cdot k}||x^{(0)} - x^*||_2^2\Big| \\ \iff |f(x^{(k)}) - f(x^*)| &\leq \frac{1}{2 \cdot t^{(1)} \cdot k}||x^{(0)} - x^*||_2^2 \\ &= \frac{1}{k}\Big(\frac{1}{2 \cdot t^{(1)}}||x^{(0)} - x^*||_2^2\Big)\end{aligned}$$

Thus, $|f(x^{(n)}) - f(x^*)| \in \mathcal{O}_C(1/n)$, where $C = \frac{1}{2 \cdot t^{(1)}}||x^{(0)} - x^*||_2^2$. □

§4.1.3 Accelerated Proximal Gradient Method

It was later shown that a minor modification of the proximal point algorithm would allow for a convergence rate of $\mathcal{O}_C(1/n^2)$ [5]. This is the well-known *accelerated proximal gradient method* (monotone). The most traditional formulation of the accelerated proximal gradient method is presented in algorithm 2.

Algorithm 2 Accelerated Proximal Gradient Algorithm

Pick $x^{(0)} \in \mathbb{R}^n$, $\mu \geq 0$, $\lambda^{(k)} = \lambda \in (0, 1/L]$, and $N \in \mathbb{N}$

for k = 0, 1, 2, ..., N **do**

  $\omega^{(k)} := \frac{k-2}{k+1}$

  $y^{(k+1)} := x^{(k)} + \omega^{(k)}\big(x^{(k)} - x^{(k-1)}\big)$

  $x^{(k+1)} := \mathbf{prox}_{\lambda h}\Big(y^{(k+1)} - \lambda \nabla g(y^{(k+1)})\Big)$

Output: $x^{(N+1)}$

As this algorithm is of more practical use (its convergence rate is faster), we prove that it is

$\mathcal{O}_C(1/n^2)$. But, in order to make our analysis easier, we recast algorithm 2 in a more convenient manner:

Algorithm 3 Accelerated Proximal Gradient Algorithm

Pick $x^{(0)} \in \mathbb{R}^n$, $u^{(0)} := x^{(k)}$, $\mu \geq 0$, $\lambda^{(k)} = \lambda \in (0, 1/L]$, and $N \in \mathbb{N}$

for k = 1, 2, ..., N **do**

$\theta^{(k)} = \frac{2}{k+1}$

$y^{(k)} := (1 - \theta^{(k)})x^{(k-1)} + \theta^{(k)}u^{(k-1)}$

$x^{(k)} := \mathbf{prox}_{\lambda h}\left(y^{(k)} - \lambda \nabla g(y^{(k)})\right)$

$u^{(k)} := x^{(k-1)} + \frac{1}{\theta^{(k)}}(x^{(k)} - x^{(k-1)})$

Output: $u^{(N+1)}$

Proposition

Considering f as above, we will show that for the accelerated proximal gradient method presented in algorithm 2, that

$$|f(x^{(k)}) - f(x^*)| \in \mathcal{O}_C(1/k^2), \quad C = \frac{2||x^{(0)} - x^*||_2^2}{\lambda}$$

This proof does not have nearly as many details as should be necessary. For a 'complete' proof please see [5] and [1].

Proof. We will let x^* denote the optimal solution to our minimization problem.

First, we note that if $\lambda \in (0, 1/L]$, then by the Quadratic Upper Bound for L-Lipschitz Functions, we have

$$g(x) \leq g(y) + \nabla g(y)^T(x - y) + \frac{||x - y||_2^2}{2\lambda}$$

for any y, implies

$$g(x^{(k+1)}) \leq g(y) + \nabla g(y)^T(x^{(k+1)} - y) + \frac{||x^{(k+1)} - y||_2^2}{2\lambda} \tag{1}$$

for any z. Further, by the sub-gradient characterization, we have that

$$h(v) \leq h(z) + \frac{1}{\lambda}(v - w)^T(z - v), \quad \text{for all } v, w, z \in \mathbb{R}^n. \tag{2}$$

By equation 2, and the definition of proximal operator, it follows that

$$h(x^{(k+1)}) \leq h(z) + \frac{1}{\lambda}(x^{(k+1)} - y^{(k)})^T(z - x^{(k+1)}) + \nabla g(y^{(k)})^T(z - x^{(k+1)}) \tag{3}$$

for any $z \in \mathbb{R}^n$.

By adding 1 and 3 at $y = z$, we obtain

$$f(x^{(k+1)}) \leq f(z) + \frac{1}{\lambda}(x^{(k+1)} - y^{(k)})^T(z - x^{(k+1)}) + \frac{||x^{(k+1)} - z||_2^2}{2\lambda} + \nabla g(y^{(k)})^T(z - x^{(k+1)}) + \nabla g(z)^T(x^{(k+1)} - z) \quad (4)$$

And, by definition of the sub-gradient of g, we have that for all $z \in \mathbb{R}^n$,

$$f(x^{(k+1)}) \leq f(z) + \frac{1}{\lambda}(x^{(k+1)} - y^{(k)})^T(z - x^{(k+1)}) + \frac{||x^{(k+1)} - z||_2^2}{2\lambda} \quad (5)$$

And, by substituting $z = x^*$, and $z = x^{(k)}$ in 5, we obtain

$$f(x^{(k+1)}) - f(x^*) - (1 - \theta^{(k)})(f(x^{(k)}) - f(x^*)) \leq \frac{1}{2\lambda}||x^{(k+1)} - y^{(k)}||_2^2 + \frac{1}{\lambda}(x^{(k)} - y^{(k)})^T(\theta^{(k)} + (1 - \theta^{(k)})x^{(k)} - x^{(k+1)}) \quad (6)$$

But then, we have that

$$f(x^{(k+1)}) - f(x^*) - (1 - \theta^{(k)})(f(x^{(k)}) - f(x^*)) \leq \frac{(\theta^{(k)})^2}{2\lambda}\Big(||u^{(k)} - x^*||_2^2 - ||u^{(k+1)} - x^*||_2^2\Big) \quad (7)$$

A simple argument show that

$$\frac{1 - \theta^{(k)}}{(\theta^{(k)})^2} \leq \frac{1}{(\theta^{(k-1)})^2}$$

Using this fact, 7, and summing of k terms, we obtain

$$\frac{\lambda}{(\theta^{(k)})^2}(f(x^{(k)}) - f(x^*)) + \frac{1}{2}||u^{(k)} - x^*||_2^2 \leq \frac{||x^{(0)} - x^*||_2^2}{2} \quad (8)$$

$$\implies |f(x^{(k)}) - f(x^*)| \leq \frac{(\theta^{(k)})^2}{2\lambda}||x^{(0)} - x^*||_2^2 \quad (9)$$

$$= \frac{2}{\lambda(k+1)^2}||x^{(0)} - x^*||_2^2 \quad (10)$$

$$\leq \frac{2}{\lambda k^2}||x^{(0)} - x^*||_2^2 \quad (11)$$

Thus,

$$|f(x^{(k)}) - f(x^*)| \in \mathcal{O}_C(1/k^2), \quad C = \frac{2}{\lambda}||x^{(0)} - x^*||_2^2$$

□

5 Implementation of the Proximal Methods

§5.1 Proximal Gradient Method

In this section, we present the pseudo-code necessary to develop an implementation of the proximal gradient method without line search, and a Python sample. The Python sample will be the general method for an Iterative Shrinkage-Thresholding Algorithm [4].

We assume the decomposition of f, g, h as in §4.1.2, and $t^{(k)} = t \in (0, 1/L]$.

§5.1.1 Pseudo-code

Algorithm 4 Proximal Gradient Algorithm

Pick $x^{(0)} \in \mathbb{R}^n$, $t \in (0, 1/L]$, and $\epsilon \in \mathbb{R}$.

Calculate $x^{(1)} = \mathbf{prox}_{th}\left(x^{(0)} - t\nabla g(x^{(0)})\right)$

Initialize $k = 0$.

while $|f(x^{(k+1)}) - f(x^{(k)})| > \epsilon$ **do**

    $x^{(k+1)} := \mathbf{prox}_{th}\left(x^{(k)} - t\nabla g(x^{(k)})\right)$

    $k := k + 1$

Output: $x^{(k)}$

[4] The shrinkage operator is given in the selected proof section.

§5.1.1.1 Python Implementation First, we present the pseudo-code, above. Then, we give the reader a least squares gradient program for actual implementation [2]. Further, we point the reader to these links [7], [14], for further implementations.

Listing 1 Main

```
def ista(x_init, grad, prox, n_iter=100, step=1., callback=None):
    """ISTA algorithm.

    Arguments
    ---------
    x_init : array-like
        Initial parameter values.
    grad : function
        Gradient function.
    prox : function
        Proximal operator.
    """
    x = x_init.copy()

    for _ in range(n_iter):
        x = prox(x - step * grad(x), step)

        # Update metrics after each iteration.
        if callback is not None:
            callback(x)
    return x
```

Listing 2 Sample

```
def least_squares_grad(x, features, labels):
    """Evaluates the gradient of the least square function."""
    n_samples = features.shape[0]
    x = x.reshape(1, n_features)  # Added for scipy.optimize compatibility
    grad_array = (features.dot(x.T) - labels) * features
    return np.sum(grad_array, axis=0) / n_samples

def prox_enet(x, l_l1, l_l2, t=1.):
    """Proximal operator for the elastic net at x"""
    x_abs = np.abs(x)
    prox_l1 = np.sign(x) * (x_abs - t * l_l1) * (x_abs > t * l_l1)
    return prox_l1 / (1. + t * l_l2)

step = norm(features.T.dot(features) / n_samples, 2)
```

§5.2 Accelerated Proximal Gradient Method

In this section, we present the pseudo-code necessary to develop an implementation of the accelerate proximal gradient method.

Algorithm 5 Accelerated Proximal Gradient Algorithm

Pick $x^{(0)} \in \mathbb{R}^n$, $\mu \geq 0$, $\lambda^{(k)} = \lambda \in (0, 1/L]$, $\epsilon > 0$ and $N \in \mathbb{N}$

for k = 0, 1, 2, ..., N OR $|f(x^{(k+1)}) - f(x^{(k)})| > \epsilon$ **do**

$\omega^{(k)} := \frac{k-2}{k+1}$

$y^{(k+1)} := x^{(k)} + \omega^{(k)}\left(x^{(k)} - x^{(k-1)}\right)$

$x^{(k+1)} := \mathbf{prox}_{\lambda h}\left(y^{(k+1)} - \lambda \nabla g(y^{(k+1)})\right)$

Output: $x^{(N+1)}$

§5.2.0.1 Matlab Implementation We present an implementation of a LASSO variant, which uses a fast proximal gradient method. For a more complete list of code, please see [4].

We assume the decomposition of f, g, h as in §4.1.2, and $t^{(k)} = t \in (0, 1/L]$.

Listing 3 Fast Proximal Gradient

```
            lambda = 1;
                tic;

                x = zeros(n,1);
                xprev = x;
                for k = 1:MAX_ITER
                    y = x + (k/(k+3))*(x - xprev);
                    while 1
                        grad_y = AtA*y - Atb;
                        z = prox_l1(y - lambda*grad_y, lambda*gamma);
                        if f(z) <= f(y) + grad_y'*(z - y) + (1/(2*lambda))*sum_square(z - y)
                            break;
                        end
                        lambda = beta*lambda;
                    end
                    xprev = x;
                    x = z;

                    h.fast_optval(k) = objective(A, b, gamma, x, x);
                    if k > 1 && abs(h.fast_optval(k) - h.fast_optval(k-1)) < ABSTOL
                        break;
                    end
                end

                h.x_fast = x;
                h.p_fast = h.fast_optval(end);
                h.fast_toc = toc;
```

Appendices

A Definitions and Examples

We present the some of the core terms, and the associated notation used throughout this paper. In addition, we provide the reader with some insightful examples:

Definition Affine Set:
A subset $\Omega \subset \mathbb{R}^n$ is *affine* iff for any $a, b \in \Omega$,

$$\mathcal{L}[a, b] = \{\lambda a + (1 - \lambda)b : \lambda \in \mathbb{R}\} \subset \Omega$$

Definition Level Set:
Let $h : \mathbb{R}^n \to (-\infty, \infty]$. For any $\alpha \in \mathbb{R}$ we define the level set $\mathscr{L}_\alpha$ as the following:

$$\mathscr{L}_\alpha = \{x \in \mathbb{R}^n : h(x) \leq \alpha\}$$

Definition Subgradient:
Let $f : \mathbb{R}^n \to (-\infty, \infty]$ be a convex function and let $\overline{x} \in dom f$. An element $v \in \mathbb{R}^n$ is called a *subgradient* of f at $\overline{x}$ if

$$\langle v, x - \overline{x}\rangle \leq f(x) - f(\overline{x})$$

for all $x \in \mathbb{R}^n$; the collection of all subradients at $\overline{x}$ is called the *sub-differential* of the function at $\overline{x}$, denoted $\partial f(\overline{x})$.

Definition Ω-*distance*:
Given $\Omega \subset \mathbb{R}^n$ and $x \in \mathbb{R}^n$, we define the distance associated with Ω to be the following:

$$d(x, \Omega) = \inf\{||x - \omega|| : \omega \in \Omega\}$$

Definition Support Function:
Given a non-empty subset $\Omega \subset \mathbb{R}^n$ the support function of Ω is defined by the following:

$$\sigma_\Omega(x) := \sup\{\langle x, \omega\rangle : \omega \in \Omega\}$$

for all $x \in \mathbb{R}^n$.

Ex 1. Here we investigate the support function of a convex set $\Omega \subset \mathbb{R}^n$. Consider the unit square $\Omega = [0, 1] \times [0, 1]$. And consider $(1, 0) \in [0, 1] \times [0, 1]$. The supporting hyperplane to this point is the vertical line $x = 1$. It is, also, clear that $\sigma_\Omega((1, 0)) = \sup\{\langle(1, 0), \omega\rangle : \omega \in \Omega\} = 1$. Thus, the supporting hyper-plane at $(1, 0)$ can be described as $H = \{\langle(1, 0), \omega\rangle = 1 : \omega \in \Omega\}$. Considering the other points along the boundary, we see that the convex set Ω can be uniquely described by its support function. Although this example was simple, we can see that any

convex set may have several supporting hyperplanes.

Definition Fenchel Conjugates:
Given a function $f : \mathbb{R}^n \to (-\infty, \infty]$, the *Fenchel Conjugate*, $f^* : \mathbb{R}^n \to [-\infty, \infty]$ is defined as the following:

$$\begin{aligned} f^*(v) &= \sup\{\langle v, x\rangle - f(x) : x \in \mathbb{R}^n\} \\ &= \sup\{\langle v, x\rangle - f(x) : x \in \text{dom} f\} \end{aligned}$$

We now present and discuss some of the functional significance of the Fenchel Conjugate: The most immediate example of the Fenchel Conjugate is the indicator associated with $\Omega \subset \mathbb{R}^n$: $I_\Omega(v) = f(v)$.

Ex 2. We compute $f^*(v)$:

Since,

$$I_\Omega(v) = \begin{cases} 0 & v \in \Omega \\ \infty & v \notin \Omega \end{cases}$$

It follows that,

$$\begin{aligned} f^*(v) &= \sup\{\langle v, x\rangle - f(x) : x \in \mathbb{R}^n\} \\ &= \sup\{\langle v, x\rangle : x \in \mathbb{R}^n\} \\ &= \sup\{\langle v, x\rangle : x \in \Omega\} \\ &= \sigma_\Omega(v) \end{aligned}$$

Ex 3. As another example, we compute the Fenchel Conjugate of $f(x) = x^2$:
From the definition, consider

$$\begin{aligned} f^*(v) &= \sup\{\langle x, v\rangle - f(x) : x \in \mathbb{R}\} \\ &= \sup\{xv - x^2 : x \in \mathbb{R}\} \\ &= \sup\{x(v - x) : x \in \mathbb{R}\} \\ &= \left(\frac{v}{2}\right)^2 \end{aligned}$$

Ex 4. As another example, we compute the Fenchel Conjugate of $f(x) = |x|$:
From the definition, consider

$$\begin{aligned} f^*(v) &= \sup\{\langle x, v\rangle - f(x) : x \in \mathbb{R}^n\} \\ &= \sup\{xv - |x| : x \in \mathbb{R}^n\} \end{aligned}$$

Thus, there are three cases, either $x > 0$, $x = 0$ or $x < 0$; Consider the following:

$$f^*(v) = \sup_{x\in\mathbb{R}} \begin{cases} xv - x & x > 0 \\ 0 & x = 0 \\ xv + x & x < 0 \end{cases}$$

So, we see that if $|v| > 1$, then we have $f^*(v) = \infty$, and if $|v| \leq 1$, we have $f^*(v) = 0$. Therefore, we conclude that

$$f^*(v) = \begin{cases} \infty & |v| > 1 \\ 0 & |v| \leq 1 \end{cases}$$

Ex 5. As a final example we compute the Fenchel Conjugate of $f(x) = e^x$:
By the definition, we have the following:

$$\begin{aligned} f^*(v) &= \sup\{\langle x, v\rangle - f(x) : x \in \mathbb{R}\} \\ &= \sup\{xv - e^x : x \in \mathbb{R}\} \end{aligned}$$

Taking the derivative of $g(x) = xv - e^x$ for some $v \in \mathbb{R}$, we see that

$$\begin{aligned} \frac{d}{dx}(xv - e^x) &= v - e^x \quad \text{and} \\ \frac{d^2}{(dx)^2}(xv - e^x) &= -e^x \end{aligned}$$

Thus, $g(x)$ is a concave-down function. Therefore,

$$0 = v - e^x \iff x = \ln(v)$$

And so, $g(x)$ achieves its maximum at $\ln(v) = x$. We conclude that

$$\begin{aligned} f^*(v) &= \sup\{xv - e^x : x \in \mathbb{R}\} \\ &= \begin{cases} 0 & v = 0 \\ \infty & v < 0 \\ \ln(v)v - v & v > 0 \end{cases} \end{aligned}$$

B Selected Proofs

Proposition

Let $f : \mathbb{R}^n \to (-\infty, \infty]$ be a functions with dom $f \neq \emptyset$. Then, we have the following:

1. $\langle v, x\rangle \leq f(x) + f^*(v) \quad$ for all $x, v \in \mathbb{R}$
2. $f^{**}(x) \leq f(x) \quad$ for all $x \in \mathbb{R}^n$

Proof.

- First, if $f(x) = \infty$ for some $x \in \mathbb{R}^n$, then we are done. Otherwise, if $x \in \text{dom}\, f$, then by the properties of Fenchel Conjugates we have $f^*(v) \geq \langle v, x\rangle - f(x)$.
- Likewise,
$$\sup\{\langle v, x\rangle - f^*(v) | v \in \mathbb{R}^n\} \leq f(x) \quad \text{for all } x, v \in \mathbb{R}^n$$

□

Proposition

Let $\bar{x} \in \text{dom}\, f$, where $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ is a convex function. Further, suppose that $\partial f(\bar{x}) \neq \emptyset$. Then, we have the following:
$$f^{**}(\bar{x}) = f(\bar{x})$$

Proof. By a previous proposition, we have $f^{**}(x) \leq f(x)$ and so it suffice to only show that $f^{**}(x) \geq f(x)$. Indeed, by the previous theorem, if $v \in \partial f(\bar{x})$, then $\langle v, \bar{x}\rangle = f(\bar{x}) + f^*(v)$. Thus,
$$f(\bar{x}) = \langle v, \bar{x}\rangle - f^*(v) \leq \sup\{\langle v, \bar{x}\rangle - f^*(v) : v \in \mathbb{R}^n\} = f^{**}(\bar{x})$$
Which completes the proof. □

Proposition

Let $h : \mathbb{R}^n \to \bar{R}$ be a convex function, such that $\partial h(x) \neq \emptyset$. Then, the following holds:
$$x \in \partial h(y) \quad \text{iff} \quad y \in \partial h^*(x)$$

Proof. ($\Longleftrightarrow$)

$$\begin{aligned} x \in \partial h(y) &\iff \langle x, y\rangle = h(y) + h^*(x) \\ \text{(by previous proposition)} &\iff \langle x, y\rangle = h^*(y) + h^{**}(x) \\ &\iff \langle x, y\rangle = h^*(y) + h(x) \\ &\iff y \in \partial h^*(x) \end{aligned}$$

□

Proposition

Subgradient Characterization:

$$u = \underset{h}{\mathbf{prox}}(x) \iff x - u \in \partial h(u)$$

Proof. Let $h : \mathbb{R}^n \to (-\infty, \infty]$ be a closed, proper, convex function, and let $x \in \mathbb{R}^n$. We note that for an arbitrary convex set $\Omega \subset \mathbb{R}^n$, $\Omega \neq \emptyset$ and so $\operatorname{ri}\Omega \neq \emptyset$, where ri is the relative interior.

Define $\phi(u) = h(u) + \frac{||u-x||^2}{2}$. Since h is a proper, closed, and convex, its domain $\operatorname{dom} h$ is a convex set and we have $\operatorname{dom} h \neq \emptyset$; further, $\operatorname{ri}(\operatorname{dom} h) \neq \emptyset$, by the comments above. Likewise, we have that $\operatorname{dom} \frac{||u-x||^2}{2} = \mathbb{R}^n$, which is convex and so we have $\operatorname{ri}(\operatorname{dom} \frac{||u-x||^2}{2}) \neq \emptyset$. This implies that

$$\operatorname{ri}(\operatorname{dom} h) \cap \operatorname{ri}(\operatorname{dom} \frac{||u-x||^2}{2}) = \operatorname{ri}(\operatorname{dom} h) \neq \emptyset$$

For simplicity let $h_2 = \frac{||u-x||^2}{2}$. Thus, by a previous proposition, we have

$$\begin{aligned} u = \underset{h}{\mathbf{prox}}(x) \iff 0 &\in \partial(h + h_2)(u) \\ &= \partial h(u) + \partial h_2(u) \\ &= \partial h(u) + \nabla h_2(u) \\ &= \partial h(u) + (u - x) \end{aligned}$$

Thus, $x - u \in \partial h(u)$ □

Proposition

Let $h : \mathbb{R}^n \to (-\infty, \infty]$ be a proper, closed, convex function. Let $p : \mathbb{R}^n \to \mathbb{R}^n$ such that $p(x) = \mathbf{prox}_h(x)$. Then, p is 1-Lipschitz:

$$||p(x) - p(\overline{x})|| \leq ||x - \overline{x}||$$

for all $x, \overline{x} \in \mathbb{R}$.

Before proving this, we first prove a useful lemma:

Lemma

Suppose h is a convex, extended, real-valued function. If $x_1 \in \partial h(u_1)$ and $x_2 \in \partial h(u_2)$, then

$$\langle x_1 - x_2, u_1 - u_2 \rangle \geq 0$$

Proof. By the definition of the subgradient,

$$\begin{cases} \langle x_1, x - u_1 \rangle \leq h(x) - h(u_1) & \forall x \in \mathbb{R} \\ \langle x_2, x - u_2 \rangle \leq h(x) - h(u_2) & \forall x \in \mathbb{R} \end{cases}$$

In particular it is true for the following $u_1, u_2 \in \mathbb{R}$:

$$\begin{cases} \langle x_1, u_2 - u_1 \rangle \leq h(u_2) - h(u_1) \\ \langle x_2, u_1 - u_2 \rangle \leq h(u_1) - h(u_2) \end{cases}$$

Thus, after adding the two equations, we have

$$\begin{aligned} & \langle x_1 - x_2, u_2 - u_1 \rangle \leq 0 \\ \iff & -\langle x_1 - x_2, -u_2 + u_1 \rangle \leq 0 \\ \iff & \langle x_1 - x_2, u_1 - u_2 \rangle \geq 0 \end{aligned}$$

□

With this lemma in mind, we continue with the proof:

Proof. Let h, p be given as stated above. Suppose that $u = p(x) = \mathbf{prox}_h(x)$ and $\overline{u} = p(\overline{x}) = \mathbf{prox}_h(\overline{x})$. Then, by the subdifferential characterization of $\mathbf{prox}_h(x)$, we have

$$\begin{cases} x - u \in \partial h(u) \\ \overline{x} - \overline{u} \in \partial h(\overline{u}) \end{cases}$$

We want to show that $\langle u - \overline{u}, x - \overline{x} \rangle \geq ||u - \overline{u}||^2$.

From the lemma, it follows that

$$\begin{aligned} & \langle x - u - (\overline{x} - \overline{u}), u - \overline{u} \rangle \geq 0 \\ \iff & \langle x - \overline{x} - (u - \overline{u}), u - \overline{u} \rangle \geq 0 \\ \iff & \langle x - \overline{x}, u - \overline{u} \rangle - \langle u - \overline{u}, u - \overline{u} \rangle \geq 0 \\ \iff & \langle x - u, u - \overline{u} \rangle \geq ||u - \overline{u}||^2 \end{aligned}$$

And, since $\langle x - \overline{x}, u - \overline{u} \rangle = ||x - \overline{x}|| \cdot ||u - \overline{u}||$, we have

$$||u - \overline{u}||^2 \leq ||x - \overline{x}|| \cdot ||u - \overline{u}|| \iff ||u - \overline{u}|| \leq ||x - \overline{x}||$$

Thus, $||p(x) - p(\overline{x})|| \leq ||x - \overline{x}||$. □

***Corollary* B.1.** *Let Ω be a non-empty closed convex set in $\mathbb{R}$. Then, the Euclidean Projection, $\mathcal{P}(x : \Omega) = p(x)$, is non-expansive.*

Proof. From a previous example, we have that

$$p(x) = \underset{I_\Omega}{\mathbf{prox}}(x) = \arg\min_{u \in \mathbb{R}} \{||u - x||\}$$

which is convex and non-empty and that $p : \mathbb{R} \to \mathbb{R}$. Thus, by the previous proposition, $p(x) = \mathcal{P}(x : \Omega)$ is non-expansive. □

6 References

[1] Amir Beck, M. T. (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *Imaging Sciences*, (1):183–202.

[2] arnaud (March 2, 2015). *Proximal gradient methods*.

[3] Bertsekas, D. (April 1999). *Nonlinear Programming*. Athena Scientific, Belmont, Massachusetts.

[4] Boyd, S. (xxxx). Lasso implementation.

[5] Geoff Gordon, R. T. (xxxx). Accelerated first-order methods.

[6] Jourani, A., Thibault, L., and Zagrodny, D. (2014). Differential properties of the moreau envelope. *Journal of Functional Analysis*, (266):1185 – 1237.

[7] Jupyter (March 5, 2016). *Logistic and linear regression with first order methods*. Github.com.

[8] Lange, K. (April, 2007). The mm algorithm.

[9] Le, H. Y. and Hiriart-Urruty, J. (2012). *From Eckart & Young Approximation To Moreau Envelopes. Mathematics Subject Classification*, page xxx.

[10] McDonald, J. N. and Weiss, N. A. (1999). *A Course in Real Analysis*. Academic Press Inc., San Diego, CA.

[11] Nam, N. M. (2013). *An Easy Path to Convex Analysis and Applications*. Morgan & Claypool Publishers.

[12] Parikh, N. and S, B. (2013). Proximal algorithms. *Foundations and Trends in Optimization*, Vol. 1(No. 3):123 – 231.

[13] Pendavingh, R. (xxxx). Semidefinite matrices & convex functions.

[14] rrobin314 (September 25, 2013). Ista.m. Github.com.

[15] Vandenberghe, L. (Spring 2013 – 2014). 10: Proximal point method.

[16] Vandenberghe, L. (Spring 2016). 6: Proximal gradient method.

[17] yu Sun, W., Sampaio, R., and Candido, M. (2003). *Proximal Point Algorithm for Minimization of DC Function. Journal of Comutational Mathematics*, (4):451–462.