Proximity Operators and Nonsmooth Optimization

Zev Woodstock
woodstock@zib.de

ZIB-AISST Tutorial Lecture Series
March 16, 2022

Outline

1. Motivation
2. Define our setting
3. Theory and tools
4. Algorithms

We can't use Calculus 1 for everything

A common paradigm:

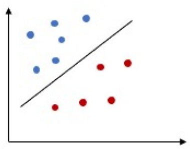
1. Define an objective function
2. Optimize with a first-order method (e.g., SGD with automatic gradients)

We can't use Calculus 1 for everything

A common paradigm:

1. Define an objective function
2. Optimize with a first-order method (e.g., SGD with automatic gradients)

[Pontil et al, 2019] Sparse linear binary classifier



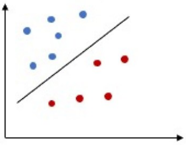
credit: adeveloperdiary.com

We can't use Calculus 1 for everything

A common paradigm:

1. Define an objective function
2. Optimize with a first-order method (e.g., SGD with automatic gradients)

[Pontil et al, 2019] Sparse linear binary classifier



credit: adeveloperdiary.com

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \sum_{i\in I_1} \max\{0, 1-\langle x \mid a_i\rangle\} + \sum_{i\in I_2} \max\{0, 1+\langle x \mid a_i\rangle\} + \lambda\|x\|_1$$

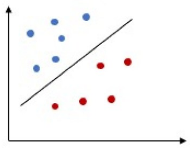
We can't use Calculus 1 for everything

A common paradigm:

1. Define an objective function
2. Optimize with a first-order method (e.g., SGD with automatic gradients)

Issue: For many objective functions, a gradient does not exist.

[Pontil et al, 2019] Sparse linear binary classifier



credit: adeveloperdiary.com

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \sum_{i\in I_1} \max\{0, 1-\langle x \mid a_i\rangle\} + \sum_{i\in I_2} \max\{0, 1+\langle x \mid a_i\rangle\} + \lambda\|x\|_1$$

We can't use Calculus 1 for everything

A common paradigm:

1. Define an objective function
2. Optimize with a first-order method (e.g., SGD with automatic gradients)

Issue: For many objective functions, a gradient does not exist.

Engineers:

credit: ripleys.com

Our setting

Let $\mathcal{H}$ be a real Hilbert space with inner product $\langle \cdot \mid \cdot \rangle$ (e.g. $\mathbb{R}^n$ with the dot product $\langle x \mid y \rangle = x^T y$).

Our setting

Let $\mathcal{H}$ be a real Hilbert space with inner product $\langle \cdot \mid \cdot \rangle$ (e.g. $\mathbb{R}^n$ with the dot product $\langle x \mid y \rangle = x^T y$).

Let $\Gamma_0(\mathcal{H}) = \{f : \mathcal{H} \to \mathbb{R} \cup \{+\infty\} | \ f$ is convex, lower-semicontinuous, and proper $\}$

Our setting

Let $\mathcal{H}$ be a real Hilbert space with inner product $\langle \cdot \mid \cdot \rangle$ (e.g. $\mathbb{R}^n$ with the dot product $\langle x \mid y \rangle = x^T y$).

Let $\Gamma_0(\mathcal{H}) = \{f : \mathcal{H} \to \mathbb{R} \cup \{+\infty\} |\ f$ is convex, lower-semicontinuous, and proper $\}$
e.g. e^x, $-\ln(x)$, $\| \cdot \|^2$,

Our setting

Let $\mathcal{H}$ be a real Hilbert space with inner product $\langle \cdot \mid \cdot \rangle$ (e.g. $\mathbb{R}^n$ with the dot product $\langle x \mid y \rangle = x^T y$).

Let $\Gamma_0(\mathcal{H}) = \{f : \mathcal{H} \to \mathbb{R} \cup \{+\infty\} |$ f is convex, lower-semicontinuous, and proper $\}$
e.g. e^x, $-\ln(x)$, $\| \cdot \|^2$, ReLU, Hinge loss, $\|\|Ax + b\|\|$, $| \cdot |_1$, $\sup\{f_i | i \in I\}$,

Our setting

Let $\mathcal{H}$ be a real Hilbert space with inner product $\langle \cdot \mid \cdot \rangle$
(e.g. $\mathbb{R}^n$ with the dot product $\langle x \mid y \rangle = x^T y$).

Let $\Gamma_0(\mathcal{H}) = \{ f : \mathcal{H} \to \mathbb{R} \cup \{+\infty\} |$ f is convex,
lower-semicontinuous, and proper $\}$
e.g. e^x, $-\ln(x)$, $\| \cdot \|^2$, ReLU, Hinge loss, $|||Ax + b|||$, $| \cdot |_1$,
$\sup\{f_i | i \in I\}$, affine composition, positive linear combinations, ...

Our setting

Let $\mathcal{H}$ be a real Hilbert space with inner product $\langle \cdot \mid \cdot \rangle$
(e.g. $\mathbb{R}^n$ with the dot product $\langle x \mid y \rangle = x^T y$).

Let $\Gamma_0(\mathcal{H}) = \{f : \mathcal{H} \to \mathbb{R} \cup \{+\infty\} | \ f$ is convex, lower-semicontinuous, and proper $\}$
e.g. e^x, $-\ln(x)$, $\| \cdot \|^2$, ReLU, Hinge loss, $|||Ax + b|||$, $| \cdot |_1$, $\sup\{f_i | i \in I\}$, affine composition, positive linear combinations, ...

This does not always include compositions of nonlinear operators, e.g., $\|\mathcal{N}(x) - d\|$ where $\mathcal{N}$ is a multilayer neural network.

For $f \in \Gamma_0(\mathcal{H})$, let's find a minimizer

$$\underset{x\in\mathcal{H}}{\operatorname{Argmin}}\ f(x) \qquad\qquad (\text{☺})$$

For $f \in \Gamma_0(\mathcal{H})$, let's find a minimizer

$$\operatorname*{Argmin}_{x\in\mathcal{H}} f(x) \qquad\qquad (\text{☺})$$

This includes constrained optimization. For a closed convex set C,

$$\iota_C(x) = \begin{cases} +\infty \text{ if } x \notin C \\ 0 \text{ if } x \in C. \end{cases}$$

For $f \in \Gamma_0(\mathcal{H})$, let's find a minimizer

$$\operatorname*{Argmin}_{x \in \mathcal{H}} f(x) \qquad\qquad (☺)$$

This includes constrained optimization. For a closed convex set C,

$$\iota_C(x) = \begin{cases} +\infty \text{ if } x \notin C \\ 0 \text{ if } x \in C. \end{cases} \qquad \text{Then } \inf_{\substack{x \in \mathcal{H} \\ x \in C}} f(x) = \inf_{x \in \mathcal{H}} f(x) + \iota_C(x).$$

For $f \in \Gamma_0(\mathcal{H})$, let's find a minimizer

$$\operatorname*{Argmin}_{x\in\mathcal{H}} f(x) \qquad\qquad (☺)$$

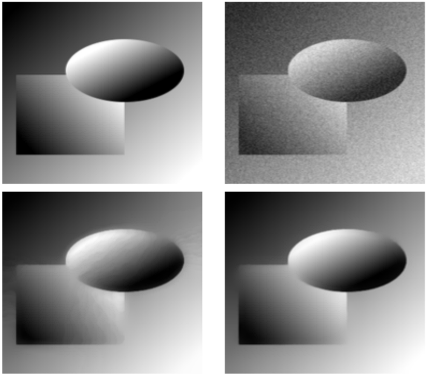
This includes constrained optimization. For a closed convex set C,

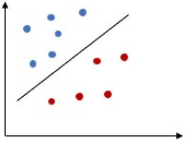
$$\iota_C(x) = \begin{cases} +\infty \text{ if } x \notin C \\ 0 \text{ if } x \in C. \end{cases} \qquad \text{Then } \inf_{\substack{x\in\mathcal{H}\\ x\in C}} f(x) = \inf_{x\in\mathcal{H}} f(x) + \iota_C(x).$$

Applications: signal processing, inverse problems, approximation theory, image processing, statistics, and machine learning.

Applications

- [Image processing: Stetzer et al. (2011)] Image recovery: Given a regularizing seminorm $|\cdot|_R$, solve an optimization problem involving $f(x) = \inf_{y \in L_2} \frac{1}{2}\|x-y\|_{L_2}^2 + |y|_R$.
- [Statistics: Square root LASSO] For $A \in \mathbb{R}^{M \times N}$ and $x \in \mathbb{R}^M$: $\textbf{minimize}_{y \in \mathbb{R}^N} \ \|Ay - x\|_2 + \|y\|_1$.
- [Machine Learning: Pontil et al., 2019] Sparse linear classifiers



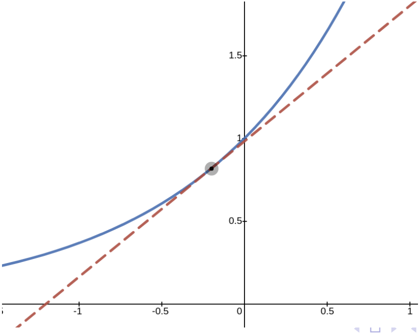


How do we solve $\nabla f = 0$ when ∇f doesn't exist?

How do we solve $\nabla f = 0$ when ∇f doesn't exist?

A **subgradient** $g \in \mathcal{H}$ of $f\colon \mathcal{H} \to]-\infty, +\infty]$ at $x \in \mathcal{H}$ satisfies

$$(\forall y \in \mathcal{H}) \qquad \langle y - x \mid g \rangle + f(x) \leq f(y),$$

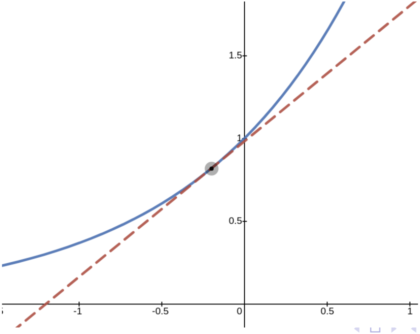


How do we solve $\nabla f = 0$ when ∇f doesn't exist?

A **subgradient** $g \in \mathcal{H}$ of $f : \mathcal{H} \to]-\infty, +\infty]$ at $x \in \mathcal{H}$ satisfies

$$(\forall y \in \mathcal{H}) \qquad \langle y - x \mid g \rangle + f(x) \leq f(y),$$

and the **subdifferential** $\partial f(x) \subset \mathcal{H}$ is the set containing all subgradients of f at x. This defines $\partial f : \mathcal{H} \to 2^{\mathcal{H}}$.



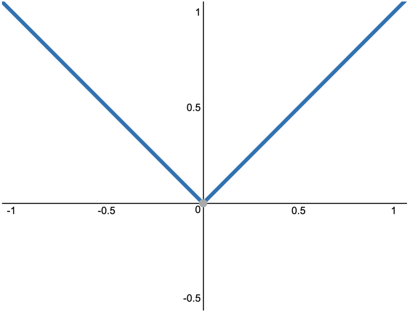
How do we solve $\nabla f = 0$ when ∇f doesn't exist?

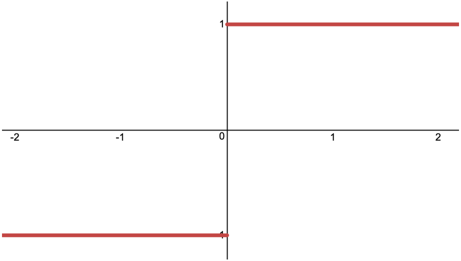
A **subgradient** $g \in \mathcal{H}$ of $f : \mathcal{H} \to]-\infty, +\infty]$ at $x \in \mathcal{H}$ satisfies

$$(\forall y \in \mathcal{H}) \qquad \langle y - x \mid g \rangle + f(x) \leq f(y),$$

and the **subdifferential** $\partial f(x) \subset \mathcal{H}$ is the set containing all subgradients of f at x. This defines $\partial f : \mathcal{H} \to 2^{\mathcal{H}}$.

Example: $f = |\cdot|$: What do we do at zero?





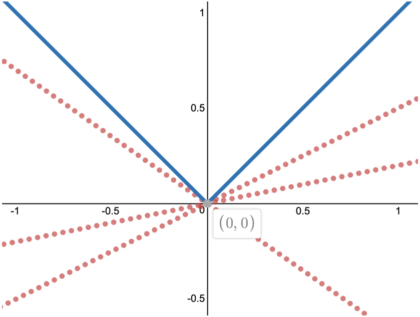
How do we solve $\nabla f = 0$ when ∇f doesn't exist?

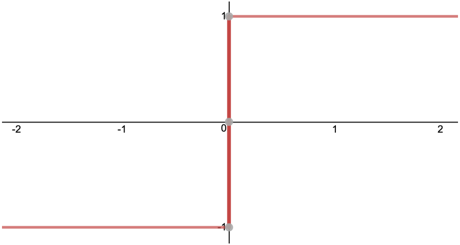
A **subgradient** $g \in \mathcal{H}$ of $f\colon \mathcal{H} \to]-\infty,+\infty]$ at $x \in \mathcal{H}$ satisfies

$$(\forall y \in \mathcal{H}) \qquad \langle y - x \mid g \rangle + f(x) \leq f(y),$$

and the **subdifferential** $\partial f(x) \subset \mathcal{H}$ is the set containing all subgradients of f at x. This defines $\partial f\colon \mathcal{H} \to 2^{\mathcal{H}}$.

Example: $f = |\cdot|$: What do we do at zero?





How do we solve $\nabla f = 0$ when ∇f doesn't exist?

A **subgradient** $g \in \mathcal{H}$ of $f : \mathcal{H} \to]-\infty, +\infty]$ at $x \in \mathcal{H}$ satisfies

$$(\forall y \in \mathcal{H}) \qquad \langle y - x \mid g \rangle + f(x) \leq f(y),$$

and the **subdifferential** $\partial f(x) \subset \mathcal{H}$ is the set containing all subgradients of f at x. This defines $\partial f : \mathcal{H} \to 2^{\mathcal{H}}$.

Fermat's Rule

For $f \in \Gamma_0(\mathcal{H})$, $x \in \operatorname{Argmin} f \Leftrightarrow 0 \in \partial f(x)$.

How do we solve $\nabla f = 0$ when ∇f doesn't exist?

A **subgradient** $g \in \mathcal{H}$ of $f : \mathcal{H} \to]-\infty, +\infty]$ at $x \in \mathcal{H}$ satisfies

$$(\forall y \in \mathcal{H}) \qquad \langle y - x \mid g \rangle + f(x) \leq f(y),$$

and the **subdifferential** $\partial f(x) \subset \mathcal{H}$ is the set containing all subgradients of f at x. This defines $\partial f : \mathcal{H} \to 2^{\mathcal{H}}$.

Fermat's Rule

For $f \in \Gamma_0(\mathcal{H})$, $x \in \operatorname{Argmin} f \Leftrightarrow 0 \in \partial f(x)$.

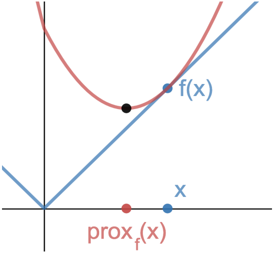
Proof:

$$\begin{aligned} 0 \in \partial f(x) &\Leftrightarrow (\forall y \in \mathcal{H}) \quad \langle y - x | 0 \rangle + f(x) \leq f(y) \\ &\Leftrightarrow (\forall y \in \mathcal{H}) \quad f(x) \leq f(y) \\ &\Leftrightarrow x \in \operatorname{Argmin} f \end{aligned}$$

Proximity operators: a new hope

The **proximity operator of** f at $x \in \mathcal{H}$ is

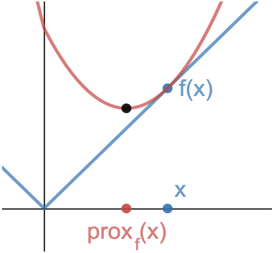
$$\operatorname{prox}_f(x) = \underset{u \in \mathcal{H}}{\operatorname{Argmin}}\; f(u) + \frac{1}{2}\|x - u\|^2$$



Proximity operators: a new hope

The **proximity operator of** f at $x \in \mathcal{H}$ is

$$\mathrm{prox}_f(x) = \underset{u \in \mathcal{H}}{\mathrm{Argmin}}\ f(u) + \frac{1}{2}\|x - u\|^2$$

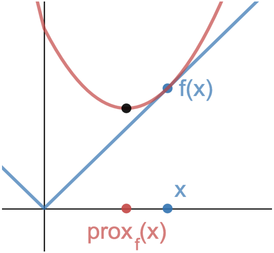


- For $f \in \Gamma_0(\mathcal{H})$ and $x \in \mathcal{H}$, $\mathrm{prox}_f(x)$ is unique.

Proximity operators: a new hope

The **proximity operator of** f at $x \in \mathcal{H}$ is

$$\operatorname{prox}_f(x) = \operatorname*{Argmin}_{u \in \mathcal{H}} f(u) + \frac{1}{2}\|x - u\|^2$$

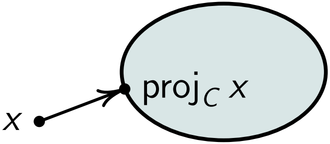


- For $f \in \Gamma_0(\mathcal{H})$ and $x \in \mathcal{H}$, $\operatorname{prox}_f(x)$ is unique. This defines an operator $\operatorname{prox}_f : \mathcal{H} \to \mathcal{H}$.

Proximity operators: a new hope

The **proximity operator of** f at $x \in \mathcal{H}$ is

$$\operatorname{prox}_f(x) = \underset{u\in\mathcal{H}}{\operatorname{Argmin}}\ f(u) + \frac{1}{2}\|x-u\|^2$$

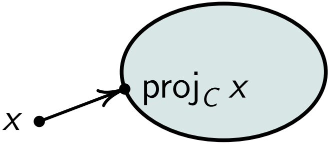


- For $f \in \Gamma_0(\mathcal{H})$ and $x \in \mathcal{H}$, $\operatorname{prox}_f(x)$ is unique. This defines an operator $\operatorname{prox}_f : \mathcal{H} \to \mathcal{H}$.
- Projections: $\operatorname{prox}_{\iota_C}(x) = \operatorname{Argmin}_{u\in C}\|x-u\|^2 = \operatorname{proj}_C x$.

Proximity operators: a new hope

The **proximity operator of** f at $x \in \mathcal{H}$ is

$$\operatorname{prox}_f(x) = \operatorname*{Argmin}_{u \in \mathcal{H}} f(u) + \frac{1}{2}\|x - u\|^2$$



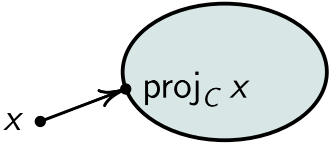
- For $f \in \Gamma_0(\mathcal{H})$ and $x \in \mathcal{H}$, $\operatorname{prox}_f(x)$ is unique. This defines an operator $\operatorname{prox}_f : \mathcal{H} \to \mathcal{H}$.
- Projections: $\operatorname{prox}_{\iota_C}(x) = \operatorname{Argmin}_{u \in C}\|x - u\|^2 = \operatorname{proj}_C x$.
- **Fixed points** of prox_f are minimizers:

$$x = \operatorname{prox}_f x \quad \Leftrightarrow \quad x \in \operatorname{Argmin} f$$

Proximity operators: a new hope

The **proximity operator of** f at $x \in \mathcal{H}$ is

$$\operatorname{prox}_f(x) = \operatorname*{Argmin}_{u \in \mathcal{H}} f(u) + \frac{1}{2}\|x-u\|^2$$



- For $f \in \Gamma_0(\mathcal{H})$ and $x \in \mathcal{H}$, $\operatorname{prox}_f(x)$ is unique. This defines an operator $\operatorname{prox}_f : \mathcal{H} \to \mathcal{H}$.
- Projections: $\operatorname{prox}_{\iota_C}(x) = \operatorname{Argmin}_{u \in C}\|x-u\|^2 = \operatorname{proj}_C x$.
- **Fixed points** of prox_f are minimizers:

$$x = \operatorname{prox}_f x \quad \Leftrightarrow \quad x \in \operatorname{Argmin} f$$

spoiler alert: $$x_{n+1} = \operatorname{prox}_f x_n \;\Rightarrow\; x_n \to x^* \in \operatorname{Argmin} f$$

What does a proximal step do?

Let $x \in \mathcal{H}$ and $\gamma > 0$.

$$x_+ = \operatorname{prox}_{\gamma f} x$$

$$\begin{aligned} x_+ = \operatorname*{Argmin}_{u \in \mathcal{H}} \gamma f(u) + \frac{1}{2}\|x - u\|^2 &\Leftrightarrow 0 \in \partial\left(\gamma f + \frac{1}{2}\|x - \cdot\|^2\right)(x_+) \\ &\Leftrightarrow 0 \in \gamma \partial f(x_+) + x_+ - x \\ &\Leftrightarrow x \in \gamma \partial f(x_+) + x_+ \end{aligned}$$

What does a proximal step do?

Let $x \in \mathcal{H}$ and $\gamma > 0$.

$$x_+ = \operatorname{prox}_{\gamma f} x$$

$$\Leftrightarrow x \in \gamma \partial f(x_+) + x_+$$

Differentiable setting: $\partial f(x_+) = \{\nabla f(x_+)\}$

What does a proximal step do?

Let $x \in \mathcal{H}$ and $\gamma > 0$.

$$x_+ = \operatorname{prox}_{\gamma f} x$$

$$\Leftrightarrow x \in \gamma \partial f(x_+) + x_+$$

Differentiable setting: $\partial f(x_+) = \{\nabla f(x_+)\}$

$$x_+ \overline{= x -}^{\text{Prox step}} \underset{\gamma > 0}{\gamma} \nabla f(x_+)$$

What does a proximal step do?

Let $x \in \mathcal{H}$ and $\gamma > 0$.

$$x_+ = \operatorname{prox}_{\gamma f} x$$

$$\Leftrightarrow x \in \gamma \partial f(x_+) + x_+$$

Differentiable setting: $\partial f(x_+) = \{\nabla f(x_+)\}$

$$x_+ \overset{\underline{\text{Prox step}}}{\underset{\gamma > 0}{=}} x - \gamma \nabla f(x_+)$$

$$x_+ \overset{\underline{\text{Gradient step}}}{=} x - \lambda \nabla f(x)$$

What does a proximal step do?

Let $x \in \mathcal{H}$ and $\gamma > 0$.

$$x_+ = \operatorname{prox}_{\gamma f} x$$

$$\Leftrightarrow x \in \gamma \partial f(x_+) + x_+$$

Differentiable setting: $\partial f(x_+) = \{\nabla f(x_+)\}$

Prox step

$$x_+ = x - \gamma \nabla f(x_+)$$

$\gamma > 0$

Gradient step

$$x_+ = x - \lambda \nabla f(x)$$

Extra restrictions on $\lambda > 0$

What does a proximal step do?

Let $x \in \mathcal{H}$ and $\gamma > 0$.

$$x_+ = \operatorname{prox}_{\gamma f} x$$

$$\Leftrightarrow x \in \gamma \partial f(x_+) + x_+$$

Differentiable setting: $\partial f(x_+) = \{\nabla f(x_+)\}$

$$x_+ \overset{\underline{\text{Prox step}}}{=} x - \gamma \nabla f(x_+)$$

$\gamma > 0$

$$x_+ \overset{\underline{\text{Gradient step}}}{=} x - \lambda \nabla f(x)$$

Extra restrictions on $\lambda > 0$

$$\operatorname{prox}_{\gamma \|\cdot\|^2/2} x = x/(\gamma + 1)$$

$$x - \lambda \nabla(\|\cdot\|^2/2)x = (1 - \lambda)x$$

Computing Proxes

[Convex Analysis and Monotone Operator Theory, 2nd ed., Bauschke & Combettes]

- Let $F(x_1, x_2) = f_1(x_1) + f_2(x_2)$, then $\mathrm{prox}_F(x_1, x_2) = (\mathrm{prox}_{f_1}(x_1), \mathrm{prox}_{f_2}(x_2))$.

Computing Proxes

[Convex Analysis and Monotone Operator Theory, 2nd ed., Bauschke & Combettes]

- Let $F(x_1, x_2) = f_1(x_1) + f_2(x_2)$, then $\text{prox}_F(x_1, x_2) = (\text{prox}_{f_1}(x_1), \text{prox}_{f_2}(x_2))$.
- Let $f \in \Gamma_0(\mathcal{H})$, $\alpha \geq 0$, $u \in \mathcal{H}$, $\beta \in \mathbb{R}$, and $\gamma > 0$ and set

$$h = f + (\alpha/2)\|\cdot - z\|^2 + \beta$$

Then, for every $x \in \mathcal{H}$,

$$\text{prox}_{\gamma h}x = \text{prox}_{\gamma(\gamma\alpha+1)^{-1}f}\left((\gamma\alpha + 1)^{-1}(x + \gamma(\alpha z \qquad))\right).$$

Computing Proxes

[Convex Analysis and Monotone Operator Theory, 2^{nd} ed., Bauschke & Combettes]

- Let $F(x_1, x_2) = f_1(x_1) + f_2(x_2)$, then $\operatorname{prox}_F(x_1, x_2) = (\operatorname{prox}_{f_1}(x_1), \operatorname{prox}_{f_2}(x_2))$.
- Let $f \in \Gamma_0(\mathcal{H})$, $\alpha \geq 0$, $u \in \mathcal{H}$, $\beta \in \mathbb{R}$, and $\gamma > 0$ and set

$$h = f + (\alpha/2)\| \cdot - z\|^2 + \beta + \langle \cdot \mid u \rangle$$

Then, for every $x \in \mathcal{H}$,

$$\operatorname{prox}_{\gamma h} x = \operatorname{prox}_{\gamma(\gamma\alpha+1)^{-1} f} \left((\gamma\alpha + 1)^{-1} (x + \gamma(\alpha z - u)) \right).$$

Computing Proxes

[Convex Analysis and Monotone Operator Theory, 2nd ed., Bauschke & Combettes]

- Let $F(x_1, x_2) = f_1(x_1) + f_2(x_2)$, then $\mathrm{prox}_F(x_1, x_2) = (\mathrm{prox}_{f_1}(x_1), \mathrm{prox}_{f_2}(x_2))$.
- Let $f \in \Gamma_0(\mathcal{H})$, $\alpha \geq 0$, $u \in \mathcal{H}$, $\beta \in \mathbb{R}$, and $\gamma > 0$ and set

$$h = f + (\alpha/2)\|\cdot - z\|^2 + \beta + \langle \cdot \mid u \rangle$$

Then, for every $x \in \mathcal{H}$,

$$\mathrm{prox}_{\gamma h} x = \mathrm{prox}_{\gamma(\gamma\alpha+1)^{-1} f}\left((\gamma\alpha + 1)^{-1}(x + \gamma(\alpha z - u))\right).$$

- If L is a bounded linear operator such that $LL^* = \mu \mathrm{Id}$ for $\mu > 0$, then for every $x \in \mathcal{H}$, $\mathrm{prox}_{f \circ L} x = x + \mu^{-1} L^* \left(\mathrm{prox}_{\mu f}(Lx) - Lx\right)$

Computing Proxes

[Convex Analysis and Monotone Operator Theory, 2nd ed., Bauschke & Combettes]

- Let $F(x_1, x_2) = f_1(x_1) + f_2(x_2)$, then $\operatorname{prox}_F(x_1, x_2) = (\operatorname{prox}_{f_1}(x_1), \operatorname{prox}_{f_2}(x_2))$.
- Let $f \in \Gamma_0(\mathcal{H})$, $\alpha \geq 0$, $u \in \mathcal{H}$, $\beta \in \mathbb{R}$, and $\gamma > 0$ and set

$$h = f + (\alpha/2)\| \cdot - z\|^2 + \beta + \langle \cdot \mid u \rangle$$

Then, for every $x \in \mathcal{H}$,

$$\operatorname{prox}_{\gamma h} x = \operatorname{prox}_{\gamma(\gamma\alpha+1)^{-1} f} \left((\gamma\alpha + 1)^{-1} (x + \gamma(\alpha z - u)) \right) .$$

- If L is a bounded linear operator such that $LL^* = \mu \mathrm{Id}$ for $\mu > 0$, then for every $x \in \mathcal{H}$,
  $\operatorname{prox}_{f \circ L} x = x + \mu^{-1} L^* \left(\operatorname{prox}_{\mu f}(Lx) - Lx \right)$
- Translation, Fenchel-Legendre conjugation, Moreau envelopes, ...

Computing Proxes

Example: Let $L\colon \mathbb{R}^n \to \mathbb{R}^n$ be a wavelet basis transform and let $b \in \mathbb{R}^n$.

$$f(x) = \|Lx - b\|_1$$

Do not solve the proximal subproblem directly!

Computing Proxes

Example: Let $L\colon \mathbb{R}^n \to \mathbb{R}^n$ be a wavelet basis transform and let $b \in \mathbb{R}^n$.

$$f(x) = \|Lx - b\|_1$$

$$\operatorname{prox}_{\|\cdot\|_1}(x) = \operatorname{soft}(x)$$

Do not solve the proximal subproblem directly!

- $\operatorname{prox}_{\|\cdot\|_1} = \operatorname{soft}$ **is the soft thresholder.** (known and easy to compute)

Computing Proxes

Example: Let $L\colon \mathbb{R}^n \to \mathbb{R}^n$ be a wavelet basis transform and let $b \in \mathbb{R}^n$.

$$f(x) = \|Lx - b\|_1$$

$$\operatorname{prox}_{\|\cdot - b\|_1}(x) = b + \operatorname{soft}(x - b)$$

Do not solve the proximal subproblem directly!

- $\operatorname{prox}_{\|\cdot\|_1} = \operatorname{soft}$ is the soft thresholder. (known and easy to compute)
- **Known results about translation.**

Computing Proxes

Example: Let $L\colon \mathbb{R}^n \to \mathbb{R}^n$ be a wavelet basis transform and let $b \in \mathbb{R}^n$.

$$f(x) = \|Lx - b\|_1$$

$$\operatorname{prox}_{\|L\cdot - b\|_1}(x) = L^*\left(b + \operatorname{soft}\left(Lx - b\right)\right)$$

Do not solve the proximal subproblem directly!

- $\operatorname{prox}_{\|\cdot\|_1} = \operatorname{soft}$ is the soft thresholder. (known and easy to compute)
- Known results about translation.
- **Known results about linear operators (note $L^*L = \mathrm{Id}$).**

Computing Proxes

Example: Let $L\colon \mathbb{R}^n \to \mathbb{R}^n$ be a wavelet basis transform and let $b \in \mathbb{R}^n$.

$$f(x) = \|Lx - b\|_1$$

$$\operatorname{prox}_{\|L\cdot - b\|_1}(x) = L^* \left(b + \operatorname{soft}\left(Lx - b\right)\right)$$

Do not solve the proximal subproblem directly!

- $\operatorname{prox}_{\|\cdot\|_1} = \operatorname{soft}$ is the soft thresholder. (known and easy to compute)
- Known results about translation.
- Known results about linear operators (note $L^* L = \mathrm{Id}$).

If we can compute the prox of the central nonlinearity, we can often figure out the rest.

Commentary

- Yields provenly-convergent algorithms on nonsmooth problems (both $f(x) \to \inf f(\mathcal{H})$ and $x_n \to x^* \in \operatorname{Argmin} f$)

Commentary

- Yields provenly-convergent algorithms on nonsmooth problems (both $f(x) \to \inf f(\mathcal{H})$ and $x_n \to x^* \in \operatorname{Argmin} f$)
- [Combettes & Glaudin, 2019] Even if f is differentiable, gradient-based algorithms do not always win against proximal algorithms.

Commentary

- Yields provenly-convergent algorithms on nonsmooth problems (both $f(x) \to \inf f(\mathcal{H})$ and $x_n \to x^* \in \operatorname{Argmin} f$)
- [Combettes & Glaudin, 2019] Even if f is differentiable, gradient-based algorithms do not always win against proximal algorithms.
- If the central nonlinearity is expensive, this can be taxing. E.g., $\operatorname{prox}_{\|\cdot\|_{\text{nuc}}}$ requires SVD.

Commentary

- Yields provenly-convergent algorithms on nonsmooth problems (both $f(x) \to \inf f(\mathcal{H})$ and $x_n \to x^* \in \operatorname{Argmin} f$)
- [Combettes & Glaudin, 2019] Even if f is differentiable, gradient-based algorithms do not always win against proximal algorithms.
- If the central nonlinearity is expensive, this can be taxing. E.g., $\mathrm{prox}_{\|\cdot\|_{\mathrm{nuc}}}$ requires SVD.
- Need a prox? Check out `proximity-operator.net`.

Proximal Point Algorithm

For every $f \in \Gamma_0(\mathcal{H})$,

$$(\forall x \in \mathcal{H}) \quad \mathrm{prox}_f x = x \quad \Leftrightarrow \quad x \in \ \mathrm{Argmin} f$$

Proximal Point Algorithm (Martinet, 1970)

Let $\gamma \in]0, +\infty[$ and $f \in \Gamma_0(\mathcal{H})$ such that $\mathrm{Argmin} f \neq \varnothing$. For any initial point $x_0 \in \mathcal{H}$, the sequence

$$x_{n+1} = \mathrm{prox}_{\gamma f}(x_n)$$

converges weakly to a point in $\mathrm{Argmin} f$.

What about 2 functions?

$$\textbf{minimize } f + g \textbf{ over } \mathcal{H} \qquad (\star)$$

What about 2 functions?

$$\textbf{minimize } f + g \textbf{ over } \mathcal{H} \qquad (\star)$$

Question: Given prox_f and prox_g, can we compute prox_{f+g} in closed form?

What about 2 functions?

$$\textbf{minimize } f + g \textbf{ over } \mathcal{H} \qquad (\star)$$

Question: Given prox_f and prox_g, can we compute prox_{f+g} in closed form?
Some restrictive examples exist, but usually no.

What about 2 functions?

$$\textbf{minimize } f + g \textbf{ over } \mathcal{H} \qquad (\star)$$

Question: Given prox_f and prox_g, can we compute prox_{f+g} in closed form?

Some restrictive examples exist, but usually no.

Solution: Splitting algorithms: algorithms which only use prox_f and prox_g to solve $(\star)$.

- Forward-Backward algorithm
- Douglas-Rachford algorithm
- The method of parallel projections
- The method of alternating projections
- Extrapolated Method of Parallel Subgradient Projections (EMOPSP)
- Tseng's Algorithm

$$\textbf{minimize } f+g \textbf{ over } \mathcal{H} \qquad (\star)$$

Forward-Backward Algorithm

Let $f \in \Gamma_0(\mathcal{H})$ and let $g\colon \mathcal{H} \to \mathbb{R}$ be convex and β-Lipschitz differentiable (for $\beta > 0$). Then if $\operatorname{Argmin}(f+g) \neq \varnothing$, the sequence

$$(\forall n \in \mathbb{N}) \left\lfloor \begin{array}{l} y_n = x_n - \gamma \nabla g(x_n) \\ x_{n+1} = \operatorname{prox}_{\gamma f} y_n \end{array} \right.$$

converges weakly to a point in $\operatorname{Argmin}(f+g)$, provided $\gamma \in \left]0, 2/\beta\right[$.

$f \equiv 0 \Rightarrow \operatorname{prox}_{\gamma f} = \operatorname{Id}$, i.e., Gradient descent
$f = \iota_C \Rightarrow \operatorname{prox}_{\gamma f} = \operatorname{proj}_C$, i.e., projected gradient descent

What if neither are differentiable?

Assume $\operatorname{Argmin}(f+g) \neq \varnothing$ and c.q. $0 \in \operatorname{int}(\operatorname{dom} f - \operatorname{dom} g)$.

Douglas-Rachford Splitting Algorithm

Let $y_0 \in \mathcal{H}$, $\gamma \in]0, +\infty[$, and $(\lambda_n)_{n\in\mathbb{N}}$ be a sequence in $[0, 2]$ such that $\sum_{n\in\mathbb{N}} \lambda_n(2-\lambda_n) = +\infty$

$$(\forall n \in \mathbb{N}) \left\lfloor \begin{array}{l} x_n = \operatorname{prox}_{\gamma g} y_n \\ u_n = \gamma^{-1}(y_n - x_n) \\ z_n = \operatorname{prox}_{\gamma f}(2x_n - y_n) \\ y_{n+1} = y_n + \lambda_n(z_n - x_n) \end{array} \right.$$

Then y_n converges weakly to $y \in \mathcal{H}$, and

- $x = \operatorname{prox}_{\gamma g} y$ is an optimal primal solution
- $\gamma^{-1}(y - x)$ is an optimal (Fenchel-Rockafellar) dual solution

Commentary on algorithms in this class

There are many variants:

- > 2 functions
- Parallel
- Block-iterative
- Accelerated
- Asynchronous

For special cases, linear convergence rates are possible (e.g., DR on closed subspaces in finite dimensions [Bauschke et al., 2014]).

Commentary on algorithms in this class

There are many variants:

- > 2 functions
- Parallel
- Block-iterative
- Accelerated
- Asynchronous

For special cases, linear convergence rates are possible (e.g., DR on closed subspaces in finite dimensions [Bauschke et al., 2014]). Convergence can be slow for particular examples.

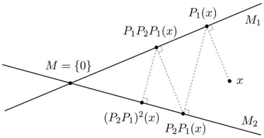


Figure 2: Alternating projections in $\mathbb{R}^2$ for two lines M_1, M_2.

credit: maths.ox.ac.uk

Commentary on algorithms in this class

There are many variants:

- > 2 functions
- Parallel
- Block-iterative
- Accelerated
- Asynchronous

For special cases, linear convergence rates are possible (e.g., DR on closed subspaces in finite dimensions [Bauschke et al., 2014]). Convergence can be slow for particular examples.

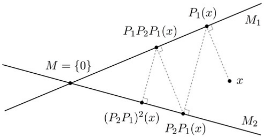


Figure 2: Alternating projections in $\mathbb{R}^2$ for two lines M_1, M_2.

credit: maths.ox.ac.uk

Usually, storage requirements increase linearly with the number of functions.

An example: product space technique

Suppose you want to

$$\text{minimize}_{x\in\mathcal{H}} \quad \sum_{i=1}^{m} f_i(x). \tag{1}$$

An example: product space technique

Suppose you want to

$$\text{minimize}_{x\in\mathcal{H}} \quad \sum_{i=1}^{m} f_i(x). \tag{1}$$

Rewrite in the Hilbert space $\mathcal{H}^m$: $\begin{cases} F(x_i)_{i\in[m]} = \sum_{i=1}^m f_i(x_i) \\ G(x_i)_{i\in[m]} = \iota_D(x_i)_{i\in[m]}, \end{cases}$
where $D = \{(x_i)_{i\in[m]} \in \mathcal{H}^m \mid x_1 = x_2 = \cdots = x_m\}$.

An example: product space technique

Suppose you want to

$$\text{minimize}_{x\in\mathcal{H}} \quad \sum_{i=1}^{m} f_i(x). \tag{1}$$

Rewrite in the Hilbert space $\mathcal{H}^m$: $\begin{cases} F(x_i)_{i\in[m]} = \sum_{i=1}^m f_i(x_i) \\ G(x_i)_{i\in[m]} = \iota_D(x_i)_{i\in[m]}, \end{cases}$ where $D = \{(x_i)_{i\in[m]} \in \mathcal{H}^m \mid x_1 = x_2 = \cdots = x_m\}$. Then (1) is equivalent to

$$\text{minimize}_{\mathrm{x}\in\mathcal{H}^m} F(\mathrm{x}) + G(\mathrm{x}).$$

An example: product space technique

Suppose you want to

$$\text{minimize}_{x\in\mathcal{H}} \quad \sum_{i=1}^{m} f_i(x). \tag{1}$$

Rewrite in the Hilbert space $\mathcal{H}^m$: $\begin{cases} F(x_i)_{i\in[m]} = \sum_{i=1}^{m} f_i(x_i) \\ G(x_i)_{i\in[m]} = \iota_D(x_i)_{i\in[m]}, \end{cases}$

where $D = \{(x_i)_{i\in[m]} \in \mathcal{H}^m \mid x_1 = x_2 = \cdots = x_m\}$. Then (1) is equivalent to

$$\text{minimize}_{\mathrm{x}\in\mathcal{H}^m} F(\mathrm{x}) + G(\mathrm{x}). \tag{2}$$

Useful facts about the prox:

$$\begin{cases} \operatorname{prox}_F(x_i)_{i\in[m]} = (\operatorname{prox}_{f_1} x_1, \operatorname{prox}_{f_2} x_2, \cdots, \operatorname{prox}_{f_m} x_m) \\ \end{cases}$$

An example: product space technique

Suppose you want to

$$\text{minimize}_{x\in\mathcal{H}} \quad \sum_{i=1}^{m} f_i(x). \tag{1}$$

Rewrite in the Hilbert space $\mathcal{H}^m$: $\begin{cases} F(x_i)_{i\in[m]} = \sum_{i=1}^m f_i(x_i) \\ G(x_i)_{i\in[m]} = \iota_D(x_i)_{i\in[m]}, \end{cases}$ where $D = \{(x_i)_{i\in[m]} \in \mathcal{H}^m \mid x_1 = x_2 = \cdots = x_m\}$. Then (1) is equivalent to

$$\text{minimize}_{\mathrm{x}\in\mathcal{H}^m} F(\mathrm{x}) + G(\mathrm{x}). \tag{2}$$

Useful facts about the prox:

$$\begin{cases} \mathrm{prox}_F(x_i)_{i\in[m]} = (\mathrm{prox}_{f_1}x_1, \mathrm{prox}_{f_2}x_2, \cdots, \mathrm{prox}_{f_m}x_m) \\ \mathrm{prox}_G(x_i)_{i\in[m]} = \mathrm{proj}_D(x_i)_{i\in[m]} = \frac{1}{m}\sum_{i=1}^m x_i \end{cases}$$

An example: product space technique

Suppose you want to

$$\text{minimize}_{x\in\mathcal{H}} \quad \sum_{i=1}^{m} f_i(x). \qquad (1)$$

Rewrite in the Hilbert space $\mathcal{H}^m$: $\begin{cases} F(x_i)_{i\in[m]} = \sum_{i=1}^m f_i(x_i) \\ G(x_i)_{i\in[m]} = \iota_D(x_i)_{i\in[m]}, \end{cases}$
where $D = \{(x_i)_{i\in[m]} \in \mathcal{H}^m \,|\, x_1 = x_2 = \cdots = x_m\}$. Then (1) is equivalent to

$$\text{minimize}_{\mathrm{x}\in\mathcal{H}^m} F(\mathrm{x}) + G(\mathrm{x}). \qquad (2)$$

Useful facts about the prox:
$\begin{cases} \mathrm{prox}_F(x_i)_{i\in[m]} = (\mathrm{prox}_{f_1}x_1, \mathrm{prox}_{f_2}x_2, \cdots, \mathrm{prox}_{f_m}x_m) \\ \mathrm{prox}_G(x_i)_{i\in[m]} = \mathrm{proj}_D(x_i)_{i\in[m]} = \frac{1}{m}\sum_{i=1}^m x_i \end{cases}$
Then use Douglas Rachford.

Thank you for your time!