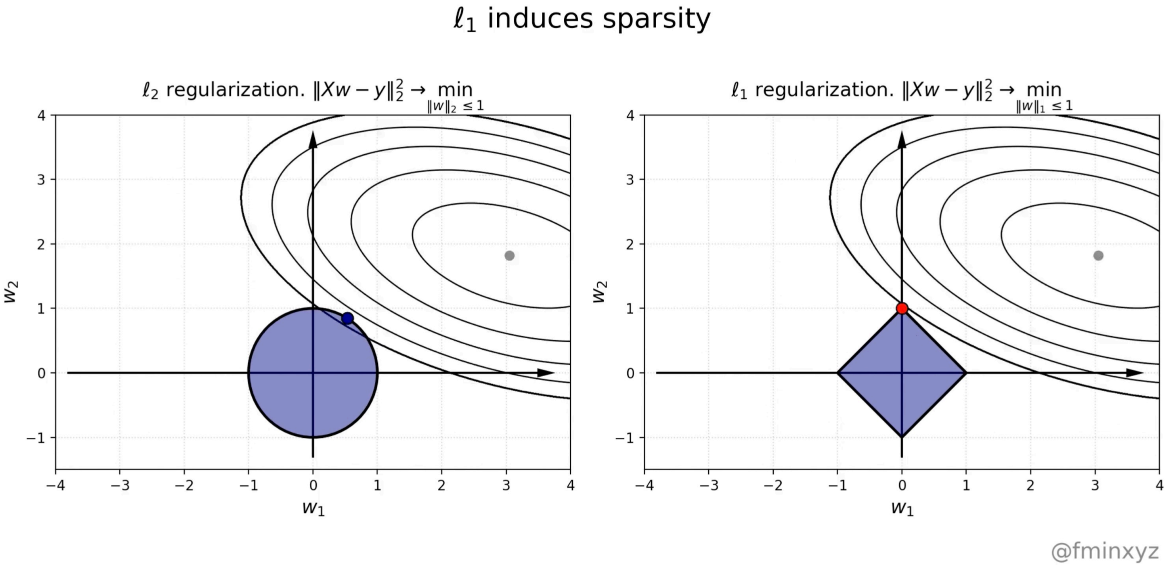
Proximal method

Daniil Merkulov

Optimization methods. MIPT

Non-smooth problems



Subgradient method

Subgradient Method: $\min_{x \in \mathbb{R}^n} f(x)$ $x_{k+1} = x_k - \alpha_k g_k, \quad g_k \in \partial f(x_k)$

Subgradient method

Subgradient Method: $\min_{x \in \mathbb{R}^n} f(x)$ $\qquad x_{k+1} = x_k - \alpha_k g_k, \quad g_k \in \partial f(x_k)$

convex (non-smooth)	strongly convex (non-smooth)
$f(x_k) - f^* \sim \mathcal{O}\left(\dfrac{1}{\sqrt{k}}\right)$ $k_\varepsilon \sim \mathcal{O}\left(\dfrac{1}{\varepsilon^2}\right)$	$f(x_k) - f^* \sim \mathcal{O}\left(\dfrac{1}{k}\right)$ $k_\varepsilon \sim \mathcal{O}\left(\dfrac{1}{\varepsilon}\right)$

Subgradient method

Subgradient Method: $\min_{x \in \mathbb{R}^n} f(x)$ $\quad x_{k+1} = x_k - \alpha_k g_k, \quad g_k \in \partial f(x_k)$

convex (non-smooth)	strongly convex (non-smooth)
$f(x_k) - f^* \sim \mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$ $k_\varepsilon \sim \mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	$f(x_k) - f^* \sim \mathcal{O}\left(\frac{1}{k}\right)$ $k_\varepsilon \sim \mathcal{O}\left(\frac{1}{\varepsilon}\right)$

Theorem

Assume that f is G-Lipschitz and convex, then Subgradient method converges as:

$$f(\overline{x}) - f^* \leq \frac{GR}{\sqrt{k}},$$

where

- $\alpha = \frac{R}{G\sqrt{k}}$

Subgradient method

Subgradient Method: $\min_{x \in \mathbb{R}^n} f(x)$ $\quad x_{k+1} = x_k - \alpha_k g_k, \quad g_k \in \partial f(x_k)$

convex (non-smooth)	strongly convex (non-smooth)
$f(x_k) - f^* \sim \mathcal{O}\left(\frac{1}{\sqrt{k}}\right)$ $k_\varepsilon \sim \mathcal{O}\left(\frac{1}{\varepsilon^2}\right)$	$f(x_k) - f^* \sim \mathcal{O}\left(\frac{1}{k}\right)$ $k_\varepsilon \sim \mathcal{O}\left(\frac{1}{\varepsilon}\right)$

Theorem

Assume that f is G-Lipschitz and convex, then Subgradient method converges as:

$$f(\overline{x}) - f^* \leq \frac{GR}{\sqrt{k}},$$

where

- $\alpha = \frac{R}{G\sqrt{k}}$
- $R = \|x_0 - x^*\|$

Subgradient method

Subgradient Method: $\min_{x \in \mathbb{R}^n} f(x)$ $\quad x_{k+1} = x_k - \alpha_k g_k, \quad g_k \in \partial f(x_k)$

convex (non-smooth)	strongly convex (non-smooth)
$f(x_k) - f^* \sim \mathcal{O}\left(\dfrac{1}{\sqrt{k}}\right)$ $k_\varepsilon \sim \mathcal{O}\left(\dfrac{1}{\varepsilon^2}\right)$	$f(x_k) - f^* \sim \mathcal{O}\left(\dfrac{1}{k}\right)$ $k_\varepsilon \sim \mathcal{O}\left(\dfrac{1}{\varepsilon}\right)$

Theorem

Assume that f is G-Lipschitz and convex, then Subgradient method converges as:

$$
f(\overline{x}) - f^* \leq \frac{GR}{\sqrt{k}},
$$

where

- $\alpha = \frac{R}{G\sqrt{k}}$
- $R = \|x_0 - x^*\|$
- $\overline{x} = \frac{1}{k}\sum_{i=0}^{k-1} x_i$

Non-smooth convex optimization lower bounds

convex (non-smooth)	strongly convex (non-smooth)
$f(x_k) - f^* \sim \mathcal{O}\left(\dfrac{1}{\sqrt{k}}\right)$ $k_\varepsilon \sim \mathcal{O}\left(\dfrac{1}{\varepsilon^2}\right)$	$f(x_k) - f^* \sim \mathcal{O}\left(\dfrac{1}{k}\right)$ $k_\varepsilon \sim \mathcal{O}\left(\dfrac{1}{\varepsilon}\right)$

Non-smooth convex optimization lower bounds

convex (non-smooth)	strongly convex (non-smooth)
$f(x_k) - f^* \sim \mathcal{O}\left(\dfrac{1}{\sqrt{k}}\right)$ $k_\varepsilon \sim \mathcal{O}\left(\dfrac{1}{\varepsilon^2}\right)$	$f(x_k) - f^* \sim \mathcal{O}\left(\dfrac{1}{k}\right)$ $k_\varepsilon \sim \mathcal{O}\left(\dfrac{1}{\varepsilon}\right)$

* Subgradient method is optimal for the problems above.

Non-smooth convex optimization lower bounds

convex (non-smooth)	strongly convex (non-smooth)
$f(x_k) - f^* \sim \mathcal{O}\left(\dfrac{1}{\sqrt{k}}\right)$ $k_\varepsilon \sim \mathcal{O}\left(\dfrac{1}{\varepsilon^2}\right)$	$f(x_k) - f^* \sim \mathcal{O}\left(\dfrac{1}{k}\right)$ $k_\varepsilon \sim \mathcal{O}\left(\dfrac{1}{\varepsilon}\right)$

- Subgradient method is optimal for the problems above.
- One can use Mirror Descent (a generalization of the subgradient method to a possiby non-Euclidian distance) with the same convergence rate to better fit the geometry of the problem.

Non-smooth convex optimization lower bounds

convex (non-smooth)	strongly convex (non-smooth)
$f(x_k) - f^* \sim \mathcal{O}\left(\dfrac{1}{\sqrt{k}}\right)$ $k_\varepsilon \sim \mathcal{O}\left(\dfrac{1}{\varepsilon^2}\right)$	$f(x_k) - f^* \sim \mathcal{O}\left(\dfrac{1}{k}\right)$ $k_\varepsilon \sim \mathcal{O}\left(\dfrac{1}{\varepsilon}\right)$

- Subgradient method is optimal for the problems above.
- One can use Mirror Descent (a generalization of the subgradient method to a possiby non-Euclidian distance) with the same convergence rate to better fit the geometry of the problem.
- However, we can achieve standard gradient descent rate $\mathcal{O}\left(\frac{1}{k}\right)$ (and even accelerated version $\mathcal{O}\left(\frac{1}{k^2}\right)$) if we will exploit the structure of the problem.

Proximal mapping intuition

Consider Gradient Flow ODE:

$$\frac{dx}{dt} = -\nabla f(x)$$

Explicit Euler discretization:

Proximal mapping intuition

Consider Gradient Flow ODE:

$$\frac{dx}{dt} = -\nabla f(x)$$

Explicit Euler discretization:

$$\frac{x_{k+1} - x_k}{\alpha} = -\nabla f(x_k)$$

Leads to ordinary Gradient Descent method

$$x_{k+1} = x_k - \alpha \nabla f(x_k)$$

Proximal mapping intuition

Consider Gradient Flow ODE:

$$\frac{dx}{dt} = -\nabla f(x)$$

Explicit Euler discretization:

$$\frac{x_{k+1} - x_k}{\alpha} = -\nabla f(x_k)$$

Leads to ordinary Gradient Descent method

Implicit Euler discretization:

$$\frac{x_{k+1} - x_k}{\alpha} = -\nabla f(x_{k+1})$$

Proximal mapping intuition

Consider Gradient Flow ODE:

$$
\frac{dx}{dt} = -\nabla f(x)
$$

Explicit Euler discretization:

$$
\frac{x_{k+1} - x_k}{\alpha} = -\nabla f(x_k)
$$

Leads to ordinary Gradient Descent method

Implicit Euler discretization:

$$
\frac{x_{k+1} - x_k}{\alpha} = -\nabla f(x_{k+1})
$$

$$
\frac{x_{k+1} - x_k}{\alpha} + \nabla f(x_{k+1}) = 0
$$

Proximal mapping intuition

Consider Gradient Flow ODE:

$$\frac{dx}{dt} = -\nabla f(x)$$

Explicit Euler discretization:

$$\frac{x_{k+1} - x_k}{\alpha} = -\nabla f(x_k)$$

Leads to ordinary Gradient Descent method

Implicit Euler discretization:

$$\frac{x_{k+1} - x_k}{\alpha} = -\nabla f(x_{k+1})$$

$$\frac{x_{k+1} - x_k}{\alpha} + \nabla f(x_{k+1}) = 0$$

$$\frac{x - x_k}{\alpha} + \nabla f(x)\Big|_{x=x_{k+1}} = 0$$

Proximal mapping intuition

Consider Gradient Flow ODE:

$$\frac{dx}{dt} = -\nabla f(x)$$

Explicit Euler discretization:

$$\frac{x_{k+1} - x_k}{\alpha} = -\nabla f(x_k)$$

Leads to ordinary Gradient Descent method

Implicit Euler discretization:

$$\frac{x_{k+1} - x_k}{\alpha} = -\nabla f(x_{k+1})$$

$$\frac{x_{k+1} - x_k}{\alpha} + \nabla f(x_{k+1}) = 0$$

$$\left.\frac{x - x_k}{\alpha} + \nabla f(x)\right|_{x = x_{k+1}} = 0$$

$$\left.\nabla \left[\frac{1}{2\alpha} \|x - x_k\|_2^2 + f(x) \right]\right|_{x = x_{k+1}} = 0$$

вып.

Proximal mapping intuition

Consider Gradient Flow ODE:

$$\frac{dx}{dt} = -\nabla f(x)$$

Explicit Euler discretization:

$$\frac{x_{k+1} - x_k}{\alpha} = -\nabla f(x_k)$$

Leads to ordinary Gradient Descent method

Implicit Euler discretization:

$$\frac{x_{k+1} - x_k}{\alpha} = -\nabla f(x_{k+1})$$

$$\frac{x_{k+1} - x_k}{\alpha} + \nabla f(x_{k+1}) = 0$$

$$\left.\frac{x - x_k}{\alpha} + \nabla f(x)\right|_{x = x_{k+1}} = 0$$

$$\left.\nabla \left[\frac{1}{2\alpha}\|x - x_k\|_2^2 + f(x)\right]\right|_{x = x_{k+1}} = 0$$

$$x_{k+1} = \text{arg}\min_{x \in \mathbb{R}^n} \left[f(x) + \frac{1}{2\alpha}\|x - x_k\|_2^2\right]$$

Proximal mapping intuition

Consider Gradient Flow ODE:

$$\frac{dx}{dt} = -\nabla f(x)$$

Explicit Euler discretization:

$$\frac{x_{k+1} - x_k}{\alpha} = -\nabla f(x_k)$$

Leads to ordinary Gradient Descent method

Implicit Euler discretization:

$$\frac{x_{k+1} - x_k}{\alpha} = -\nabla f(x_{k+1})$$

$$\frac{x_{k+1} - x_k}{\alpha} + \nabla f(x_{k+1}) = 0$$

$$\left.\frac{x - x_k}{\alpha} + \nabla f(x)\right|_{x = x_{k+1}} = 0$$

$$\left.\nabla \left[\frac{1}{2\alpha}\|x - x_k\|_2^2 + f(x)\right]\right|_{x = x_{k+1}} = 0$$

$$x_{k+1} = \text{arg}\min_{x \in \mathbb{R}^n} \left[f(x) + \frac{1}{2\alpha}\|x - x_k\|_2^2\right]$$

Proximal mapping intuition

Consider Gradient Flow ODE:

$$\frac{dx}{dt} = -\nabla f(x)$$

Explicit Euler discretization:

$$\frac{x_{k+1} - x_k}{\alpha} = -\nabla f(x_k)$$

Leads to ordinary Gradient Descent method

Implicit Euler discretization:

$$\frac{x_{k+1} - x_k}{\alpha} = -\nabla f(x_{k+1})$$

$$\frac{x_{k+1} - x_k}{\alpha} + \nabla f(x_{k+1}) = 0$$

$$\left.\frac{x - x_k}{\alpha} + \nabla f(x)\right|_{x = x_{k+1}} = 0$$

$$\left.\nabla \left[\frac{1}{2\alpha}\|x - x_k\|_2^2 + f(x)\right]\right|_{x = x_{k+1}} = 0$$

$$x_{k+1} = \text{arg}\min_{x \in \mathbb{R}^n} \left[f(x) + \frac{1}{2\alpha}\|x - x_k\|_2^2\right]$$

Proximal operator

$$x_{k+1} = \text{prox}_{f,\alpha}(x_k) = \text{arg}\min_{x \in \mathbb{R}^n} \left[f(x) + \frac{1}{2\alpha}\|x - x_k\|_2^2\right]$$

Proximal mapping intuition

- **GD from proximal method.** Back to the discretization:

Proximal mapping intuition

$$\frac{x_{k+1} - x_k}{\alpha} = -\nabla f(x_{k+1})$$

- **GD from proximal method.** Back to the discretization:

$$x_{k+1} + \alpha \nabla f(x_{k+1}) = x_k$$

Proximal mapping intuition

- **GD from proximal method.** Back to the discretization:

$$
\begin{aligned}
x_{k+1} + \alpha \nabla f(x_{k+1}) &= x_k \\
(I + \alpha \nabla f)(x_{k+1}) &= x_k
\end{aligned}
$$

Proximal mapping intuition

- **GD from proximal method.** Back to the discretization:

$$
\begin{aligned}
x_{k+1} + \alpha \nabla f(x_{k+1}) &= x_k \\
(I + \alpha \nabla f)(x_{k+1}) &= x_k \\
x_{k+1} = (I + \alpha \nabla f)^{-1} x_k &\overset{\alpha \to 0}{\approx} (I - \alpha \nabla f) x_k
\end{aligned}
$$

Proximal mapping intuition

- **GD from proximal method.** Back to the discretization:

$$
\begin{aligned}
x_{k+1} + \alpha \nabla f(x_{k+1}) &= x_k \\
(I + \alpha \nabla f)(x_{k+1}) &= x_k \\
x_{k+1} = (I + \alpha \nabla f)^{-1} x_k &\overset{\alpha \to 0}{\approx} (I - \alpha \nabla f) x_k = x_k - \alpha \nabla f(x_k)
\end{aligned}
$$

Proximal mapping intuition

- **GD from proximal method.** Back to the discretization:

$$
\begin{aligned}
x_{k+1} + \alpha \nabla f(x_{k+1}) &= x_k \\
(I + \alpha \nabla f)(x_{k+1}) &= x_k \\
x_{k+1} = (I + \alpha \nabla f)^{-1} x_k &\overset{\alpha \to 0}{\approx} (I - \alpha \nabla f) x_k
\end{aligned}
$$

  Thus, we have a usual gradient descent with $\alpha \to 0$: $x_{k+1} = x_k - \alpha \nabla f(x_k)$

- **Newton from proximal method.** Now let's consider proximal mapping of a second order Taylor approximation of the function $f^{II}_{x_k}(x)$:

Proximal mapping intuition

- **GD from proximal method.** Back to the discretization:

$$
\begin{aligned}
x_{k+1} + \alpha \nabla f(x_{k+1}) &= x_k \\
(I + \alpha \nabla f)(x_{k+1}) &= x_k \\
x_{k+1} = (I + \alpha \nabla f)^{-1} x_k &\overset{\alpha \to 0}{\approx} (I - \alpha \nabla f) x_k
\end{aligned}
$$

  Thus, we have a usual gradient descent with $\alpha \to 0$: $x_{k+1} = x_k - \alpha \nabla f(x_k)$

- **Newton from proximal method.** Now let's consider proximal mapping of a second order Taylor approximation of the function $f^{II}_{x_k}(x)$:

$$
x_{k+1} = \text{prox}_{f^{II}_{x_k}, \alpha}(x_k) = \text{arg}\min_{x \in \mathbb{R}^n} \left[f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{1}{2} \langle \nabla^2 f(x_k)(x - x_k), x - x_k \rangle + \frac{1}{2\alpha} \|x - x_k\|_2^2 \right]
$$

Proximal mapping intuition

- **GD from proximal method.** Back to the discretization:

$$
\begin{aligned}
x_{k+1} + \alpha \nabla f(x_{k+1}) &= x_k \\
(I + \alpha \nabla f)(x_{k+1}) &= x_k \\
x_{k+1} = (I + \alpha \nabla f)^{-1} x_k &\overset{\alpha \to 0}{\approx} (I - \alpha \nabla f) x_k
\end{aligned}
$$

  Thus, we have a usual gradient descent with $\alpha \to 0$: $x_{k+1} = x_k - \alpha \nabla f(x_k)$

- **Newton from proximal method.** Now let's consider proximal mapping of a second order Taylor approximation of the function $f^{II}_{x_k}(x)$:

$$
x_{k+1} = \text{prox}_{f^{II}_{x_k}, \alpha}(x_k) = \arg\min_{x \in \mathbb{R}^n} \left[f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{1}{2} \langle \nabla^2 f(x_k)(x - x_k), x - x_k \rangle + \frac{1}{2\alpha} \|x - x_k\|_2^2 \right]
$$

$$
\nabla f(x_k) + \nabla^2 f(x_k)(x - x_k) + \frac{1}{\alpha}(x - x_k) \Big|_{x = x_{k+1}} = 0
$$

Proximal mapping intuition

- **GD from proximal method.** Back to the discretization:

$$
\begin{aligned}
x_{k+1} + \alpha \nabla f(x_{k+1}) &= x_k \\
(I + \alpha \nabla f)(x_{k+1}) &= x_k \\
x_{k+1} = (I + \alpha \nabla f)^{-1} x_k &\overset{\alpha \to 0}{\approx} (I - \alpha \nabla f) x_k
\end{aligned}
$$

  Thus, we have a usual gradient descent with $\alpha \to 0$: $x_{k+1} = x_k - \alpha \nabla f(x_k)$

- **Newton from proximal method.** Now let's consider proximal mapping of a second order Taylor approximation of the function $f^{II}_{x_k}(x)$:

$$
x_{k+1} = \text{prox}_{f^{II}_{x_k}, \alpha}(x_k) = \text{arg}\min_{x \in \mathbb{R}^n} \left[f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \frac{1}{2} \langle \nabla^2 f(x_k)(x - x_k), x - x_k \rangle + \frac{1}{2\alpha} \|x - x_k\|_2^2 \right]
$$

$$
\left. \nabla f(x_k) + \nabla^2 f(x_k)(x - x_k) + \frac{1}{\alpha}(x - x_k) \right|_{x = x_{k+1}} = 0
$$

$$
x_{k+1} = x_k - \left[\nabla^2 f(x_k) + \frac{1}{\alpha} I \right]^{-1} \nabla f(x_k)
$$

$\alpha \to 0$ $\quad x_k - \alpha \nabla f(x_k)$

$\alpha \to \infty$ $\quad x_k - [\nabla^2 f(x_k)]^{-1} \nabla f(x_k)$

From projections to proximity

Let $\mathbb{I}_S$ be the indicator function for closed, convex S. Recall orthogonal projection $\pi_S(y)$

From projections to proximity

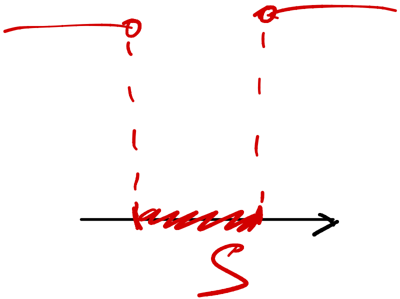
Let $\mathbb{I}_S$ be the indicator function for closed, convex S. Recall orthogonal projection $\pi_S(y)$

$$\pi_S(y) := \arg\min_{x \in S} \frac{1}{2}\|x - y\|_2^2.$$

From projections to proximity

Let $\mathbb{I}_S$ be the indicator function for closed, convex S. Recall orthogonal projection $\pi_S(y)$

$$\pi_S(y) := \arg\min_{x \in S} \frac{1}{2}\|x - y\|_2^2.$$



With the following notation of indicator function

$$\mathbb{I}_S(x) = \begin{cases} 0, & x \in S, \\ \infty, & x \notin S, \end{cases}$$

From projections to proximity

Let $\mathbb{I}_S$ be the indicator function for closed, convex S. Recall orthogonal projection $\pi_S(y)$

$$\pi_S(y) := \arg\min_{x \in S} \frac{1}{2}\|x - y\|_2^2.$$

With the following notation of indicator function

$$\mathbb{I}_S(x) = \begin{cases} 0, & x \in S, \\ \infty, & x \notin S, \end{cases}$$

Rewrite orthogonal projection $\pi_S(y)$ as

$$\pi_S(y) := \arg\min_{x \in \mathbb{R}^n} \left(\frac{1}{2}\|x - y\|^2 + \mathbb{I}_S(x) \right).$$

$+\infty, \; x \notin S$

$\frac{1}{2}\|x-y\|^2, \; x \in S$

$$\text{PROX}_f(y) = \arg\min_{x \in \mathbb{R}^n} \left(\frac{1}{2}\|x-y\|_2^2 + f(x) \right)$$

$\lambda = 1$

From projections to proximity

Let $\mathbb{I}_S$ be the indicator function for closed, convex S. Recall orthogonal projection $\pi_S(y)$

$$\pi_S(y) := \arg\min_{x \in S} \frac{1}{2}\|x - y\|_2^2.$$

With the following notation of indicator function

$$\mathbb{I}_S(x) = \begin{cases} 0, & x \in S, \\ \infty, & x \notin S, \end{cases}$$

$$x_{k+1} = \text{PROJ}_S(x_k - \alpha \nabla f(x_k))$$

$$x_{k+1} = \text{PROX}_r(x_k - \alpha \nabla f(x_k))$$

Rewrite orthogonal projection $\pi_S(y)$ as

$$\pi_S(y) := \arg\min_{x \in \mathbb{R}^n} \frac{1}{2}\|x - y\|^2 + \mathbb{I}_S(x).$$

Proximity: Replace $\mathbb{I}_S$ by some convex function!

$$\text{prox}_r(y) = \text{prox}_{r,1}(y) := \arg\min \frac{1}{2}\|x - y\|^2 + r(x)$$

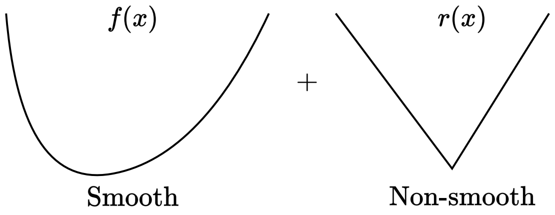
Regularized / Composite Objectives

Many nonsmooth problems take the form

$$\min_{x \in \mathbb{R}^n} \varphi(x) = f(x) + r(x)$$

- **Lasso, L1-LS, compressed sensing**

$$f(x) = \frac{1}{2}\|Ax - b\|_2^2, r(x) = \lambda\|x\|_1$$



Regularized / Composite Objectives

$\frac{1}{\sqrt{k}}$ $\frac{1}{k}$, $\frac{1}{k^2}$

Many nonsmooth problems take the form

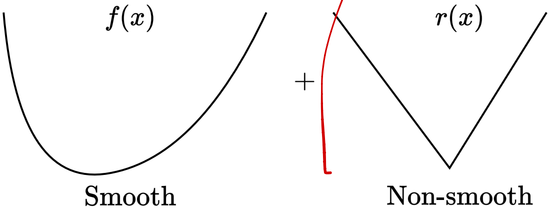
$$\min_{x \in \mathbb{R}^n} \varphi(x) = f(x) + r(x)$$

- **Lasso, L1-LS, compressed sensing**

$$f(x) = \frac{1}{2}\|Ax - b\|_2^2, r(x) = \lambda \|x\|_1$$

- **L1-Logistic regression, sparse LR**

$$f(x) = -y \log h(x) - (1-y) \log(1 - h(x)), r(x) = \lambda \|x\|_1$$



Если выпукл r(x)

PROX-friendly

Proximal mapping intuition

$\varphi(x) = f(x) + r(x)$

Optimality conditions:

$$0 \in \nabla f(x^*) + \partial r(x^*)$$

Proximal mapping intuition

Optimality conditions:

$$
\begin{aligned}
0 &\in \nabla f(x^*) + \partial r(x^*) \\
0 &\in \alpha \nabla f(x^*) + \alpha \partial r(x^*)
\end{aligned}
$$

Proximal mapping intuition

Optimality conditions:

$$
\begin{aligned}
0 &\in \nabla f(x^*) + \partial r(x^*) \\
0 &\in \alpha \nabla f(x^*) + \alpha \partial r(x^*) \\
x^* &\in \alpha \nabla f(x^*) + (I + \alpha \partial r)(x^*)
\end{aligned}
$$

Proximal mapping intuition

Optimality conditions:

$$
\begin{aligned}
0 &\in \nabla f(x^*) + \partial r(x^*) \\
0 &\in \alpha \nabla f(x^*) + \alpha \partial r(x^*) \\
x^* &\in \alpha \nabla f(x^*) + (I + \alpha \partial r)(x^*) \\
x^* - \alpha \nabla f(x^*) &\in (I + \alpha \partial r)(x^*)
\end{aligned}
$$

Proximal mapping intuition

Optimality conditions:

$$
\begin{aligned}
0 &\in \nabla f(x^*) + \partial r(x^*) \\
0 &\in \alpha \nabla f(x^*) + \alpha \partial r(x^*) \\
x^* &\in \alpha \nabla f(x^*) + (I + \alpha \partial r)(x^*) \\
x^* - \alpha \nabla f(x^*) &\in (I + \alpha \partial r)(x^*) \\
x^* &= (I + \alpha \partial r)^{-1}(x^* - \alpha \nabla f(x^*))
\end{aligned}
$$

Proximal mapping intuition

$$\text{prox}_{r,\alpha}(y) = (I + \alpha\partial r)^{-1}(y)$$

Optimality conditions:

$$
\begin{aligned}
0 &\in \nabla f(x^*) + \partial r(x^*) \\
0 &\in \alpha \nabla f(x^*) + \alpha \partial r(x^*) \\
x^* &\in \alpha \nabla f(x^*) + (I + \alpha \partial r)(x^*) \\
x^* - \alpha \nabla f(x^*) &\in (I + \alpha \partial r)(x^*) \\
x^* &= (I + \alpha \partial r)^{-1}(x^* - \alpha \nabla f(x^*)) \\
x^* &= \text{prox}_{r,\alpha}(x^* - \alpha \nabla f(x^*))
\end{aligned}
$$

Proximal mapping intuition

Optimality conditions:

$$
\begin{aligned}
0 &\in \nabla f(x^*) + \partial r(x^*) \\
0 &\in \alpha \nabla f(x^*) + \alpha \partial r(x^*) \\
x^* &\in \alpha \nabla f(x^*) + (I + \alpha \partial r)(x^*) \\
x^* - \alpha \nabla f(x^*) &\in (I + \alpha \partial r)(x^*) \\
x^* &= (I + \alpha \partial r)^{-1}(x^* - \alpha \nabla f(x^*)) \\
x^* &= \mathsf{prox}_{r,\alpha}(x^* - \alpha \nabla f(x^*))
\end{aligned}
$$

Proximal mapping intuition

Optimality conditions:

$$
\begin{aligned}
0 &\in \nabla f(x^*) + \partial r(x^*) \\
0 &\in \alpha \nabla f(x^*) + \alpha \partial r(x^*) \\
x^* &\in \alpha \nabla f(x^*) + (I + \alpha \partial r)(x^*) \\
x^* - \alpha \nabla f(x^*) &\in (I + \alpha \partial r)(x^*) \\
x^* &= (I + \alpha \partial r)^{-1}(x^* - \alpha \nabla f(x^*)) \\
x^* &= \text{prox}_{r,\alpha}(x^* - \alpha \nabla f(x^*))
\end{aligned}
$$

Which leads to the proximal gradient method:

$$
x_{k+1} = \text{prox}_{r,\alpha}(x_k - \alpha \nabla f(x_k))
$$

And this method converges at a rate of $\mathcal{O}(\frac{1}{k})$!

Proximal mapping intuition

Optimality conditions:

$$
\begin{aligned}
0 &\in \nabla f(x^*) + \partial r(x^*) \\
0 &\in \alpha \nabla f(x^*) + \alpha \partial r(x^*) \\
x^* &\in \alpha \nabla f(x^*) + (I + \alpha \partial r)(x^*) \\
x^* - \alpha \nabla f(x^*) &\in (I + \alpha \partial r)(x^*) \\
x^* &= (I + \alpha \partial r)^{-1}(x^* - \alpha \nabla f(x^*)) \\
x^* &= \text{prox}_{r,\alpha}(x^* - \alpha \nabla f(x^*))
\end{aligned}
$$

Which leads to the proximal gradient method:

$$
x_{k+1} = \text{prox}_{r,\alpha}(x_k - \alpha \nabla f(x_k))
$$

And this method converges at a rate of $\mathcal{O}(\frac{1}{k})$!

Another form of proximal operator

$$
\text{prox}_{f,\alpha}(x_k) = \text{prox}_{\alpha f}(x_k) = \text{arg}\min_{x \in \mathbb{R}^n} \left[\alpha f(x) + \frac{1}{2} \|x - x_k\|_2^2 \right] \qquad \text{prox}_f(x_k) = \text{arg}\min_{x \in \mathbb{R}^n} \left[f(x) + \frac{1}{2} \|x - x_k\|_2^2 \right]
$$

Proximal operators examples

упр. $\lambda \cdot \text{sign}(t) + t - x = 0$ решить отн. t

$t>0$: $\lambda + t - x = 0 \Rightarrow t = x - \lambda$

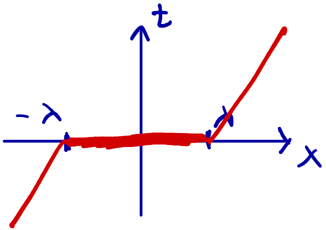
$t<0$: $-\lambda + t - x = 0 \rightarrow t = x + \lambda$

$t=0$: $[-\lambda, \lambda] + t - x = 0 \Rightarrow x \in [-\lambda; \lambda]$

- $r(x) = \lambda \|x\|_1, \; \lambda > 0$

$$
[\text{prox}_r(x)]_i = [|x_i| - \lambda]_+ \cdot \text{sign}(x_i),
$$

which is also known as soft-thresholding operator.



по опр. $\text{prox}_r(x) = \underset{\tilde{x} \in \mathbb{R}^n}{\text{argmin}} \left[r(\tilde{x}) + \frac{1}{2} \|x - \tilde{x}\|_2^2 \right] =$

$= \underset{\tilde{x} \in \mathbb{R}^n}{\text{argmin}} \left[\lambda \sum_{i=1}^n |\tilde{x}_i| + \frac{1}{2} \sum_{i=1}^n (\tilde{x}_i - x_i)^2 \right] = \text{argmin} \left(\sum_{i=1}^n \lambda |\tilde{x}_i| + \frac{1}{2} (\tilde{x}_i - x_i)^2 \right)$

$\Rightarrow \underbrace{\lambda |\tilde{x}_k| + \frac{1}{2} (\tilde{x}_k - x_k)^2}_{p(\tilde{x}_k)} \rightarrow \min_{\tilde{x}_k}$

$0 \in \partial p$

$\lambda \cdot \text{sign}(\tilde{x}_k) + \tilde{x}_k - x_k = 0$

Proximal operators examples

- $r(x) = \lambda \|x\|_1$, $\lambda > 0$

$$[\text{prox}_r(x)]_i = [|x_i| - \lambda]_+ \cdot \text{sign}(x_i),$$

which is also known as soft-thresholding operator.

- $r(x) = \frac{\lambda}{2}\|x\|_2^2$, $\lambda > 0$

$$\text{prox}_r(x) = \frac{x}{1+\lambda}.$$

$$\text{PROX}_r(x) = \underset{\tilde{x} \in \mathbb{R}^n}{\text{argmin}} \left[r(\tilde{x}) + \frac{1}{2}\|\tilde{x} - x\|_2^2 \right] =$$

$$= \underset{\tilde{x} \in \mathbb{R}^n}{\text{argmin}} \left[\frac{\lambda}{2}\|\tilde{x}\|_2^2 + \frac{1}{2}\|\tilde{x} - x\|_2^2 \right]$$

$$\nabla \ldots = 0$$

$$2\frac{\lambda}{2}\tilde{x} + 2\frac{1}{2}(\tilde{x} - x) = 0$$

$$(\lambda + 1)\tilde{x} - x = 0$$

$$\boxed{\tilde{x} = \frac{x}{1+\lambda}}$$

Proximal operators examples

- $r(x) = \lambda \|x\|_1$, $\lambda > 0$

$$[\text{prox}_r(x)]_i = [|x_i| - \lambda]_+ \cdot \text{sign}(x_i),$$

  which is also known as soft-thresholding operator.

- $r(x) = \frac{\lambda}{2}\|x\|_2^2$, $\lambda > 0$

$$\text{prox}_r(x) = \frac{x}{1+\lambda}.$$

- $r(x) = \mathbb{I}_S(x)$.

$$\text{prox}_r(x_k - \alpha \nabla f(x_k)) = \text{proj}_r(x_k - \alpha \nabla f(x_k))$$

Proximal operator properties

Theorem

Let $r : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a convex function for which prox_r is defined. If there exists such an $\hat{x} \in \mathbb{R}^n$ that $r(x) < +\infty$. Then, the proximal operator is uniquely defined (i.e., it always returns a single unique value).

Proof:

Proximal operator properties

$$\text{PROX}_r(x) = \arg\min_{\tilde{x} \in \mathbb{R}^n} \left[r(\tilde{x}) + \frac{1}{2}\|\tilde{x} - x\|_2^2 \right]$$

Theorem

Let $r : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a convex function for which prox_r is defined. If there exists such an $\hat{x} \in \mathbb{R}^n$ that $r(x) < +\infty$. Then, the proximal operator is uniquely defined (i.e., it always returns a single unique value).

Proof:

The proximal operator returns the minimum of some optimization problem.

Proximal operator properties

Theorem

Let $r : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a convex function for which prox_r is defined. If there exists such an $\hat{x} \in \mathbb{R}^n$ that $r(x) < +\infty$. Then, the proximal operator is uniquely defined (i.e., it always returns a single unique value).

Proof:

The proximal operator returns the minimum of some optimization problem.

Question: What can be said about this problem?

Proximal operator properties

Theorem

Let $r : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a convex function for which prox_r is defined. If there exists such an $\hat{x} \in \mathbb{R}^n$ that $r(x) < +\infty$. Then, the proximal operator is uniquely defined (i.e., it always returns a single unique value).

Proof:

The proximal operator returns the minimum of some optimization problem.

Question: What can be said about this problem?

It is strongly convex, meaning it has exactly one unique minimum (the existence of $\hat{x}$ is necessary for $r(\tilde{x}) + \frac{1}{2}\|x - \tilde{x}\|_2^2$ to take a finite value somewhere).

Proximal operator properties

Theorem

Let $r : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a convex function for which prox_r is defined. Then, for any $x, y \in \mathbb{R}^n$, the following three conditions are equivalent:

- $\text{prox}_r(x) = y$,

Proof

Proximal operator properties

Theorem

Let $r : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a convex function for which prox_r is defined. Then, for any $x, y \in \mathbb{R}^n$, the following three conditions are equivalent:

- $\text{prox}_r(x) = y$,
- $x - y \in \partial r(y)$,

Proof

Proximal operator properties

Theorem

Let $r : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a convex function for which prox_r is defined. Then, for any $x, y \in \mathbb{R}^n$, the following three conditions are equivalent:

- $\text{prox}_r(x) = y$,
- $x - y \in \partial r(y)$,
- $\langle x - y, z - y \rangle \leq r(z) - r(y)$ for any $z \in \mathbb{R}^n$.

Proof

Proximal operator properties

Theorem

Let $r : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a convex function for which prox_r is defined. Then, for any $x, y \in \mathbb{R}^n$, the following three conditions are equivalent:

- $\text{prox}_r(x) = y$,
- $x - y \in \partial r(y)$,
- $\langle x - y, z - y \rangle \leq r(z) - r(y)$ for any $z \in \mathbb{R}^n$.

Proof

1. Let's establish the equivalence between the first and second conditions.The first condition can be rewritten as

$$y = \arg\min_{\tilde{x} \in \mathbb{R}^d} \left(r(\tilde{x}) + \frac{1}{2}\|x - \tilde{x}\|^2 \right).$$

From the optimality condition for the convex function r, this is equivalent to:

$$0 \in \partial \left(r(\tilde{x}) + \frac{1}{2}\|x - \tilde{x}\|^2 \right)\Big|_{\tilde{x} = y} = \partial r(y) + y - x.$$

$0 \in \partial r + y - x$

$x - y \in \partial r$

Proximal operator properties

Theorem

Let $r : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be a convex function for which prox_r is defined. Then, for any $x, y \in \mathbb{R}^n$, the following three conditions are equivalent:

- $\text{prox}_r(x) = y$,
- $x - y \in \partial r(y)$
- $\langle x - y, z - y\rangle \leq r(z) - r(y)$ for any $z \in \mathbb{R}^n$.

Proof

1. Let's establish the equivalence between the first and second conditions.The first condition can be rewritten as

$$y = \arg\min_{\tilde{x} \in \mathbb{R}^d} \left(r(\tilde{x}) + \frac{1}{2}\|x - \tilde{x}\|^2 \right).$$

From the optimality condition for the convex function r, this is equivalent to:

$$0 \in \partial \left(r(\tilde{x}) + \frac{1}{2}\|x - \tilde{x}\|^2 \right)\Big|_{\tilde{x}=y} = \partial r(y) + y - x.$$

2. From the definition of the subdifferential, for any subgradient $g \in \partial f(y)$ and for any $z \in \mathbb{R}^d$:

$$\langle g, z - y\rangle \leq r(z) - r(y).$$

In particular, this holds true for $g = x - y$. Conversely, it is also clear: for $g = x - y$, the above relationship holds, which means $g \in \partial r(y)$.

Proximal operator properties

Theorem

The operator $\text{prox}_r(x)$ is firmly nonexpansive (FNE)

$$\|\text{prox}_r(x) - \text{prox}_r(y)\|_2^2 \leq \langle \text{prox}_r(x) - \text{prox}_r(y), x - y \rangle$$

and nonexpansive:

$$\|\text{prox}_r(x) - \text{prox}_r(y)\|_2 \leq \|x - y\|_2$$

Proof

1. Let $u = \text{prox}_r(x)$, and $v = \text{prox}_r(y)$. Then, from the previous property:

$$\langle x - u, z_1 - u \rangle \leq r(z_1) - r(u)$$
$$\langle y - v, z_2 - v \rangle \leq r(z_2) - r(v).$$

Proximal operator properties

Theorem

The operator $\text{prox}_r(x)$ is firmly nonexpansive (FNE)

$$\|\text{prox}_r(x) - \text{prox}_r(y)\|_2^2 \leq \langle \text{prox}_r(x) - \text{prox}_r(y), x - y \rangle$$

and nonexpansive:

$$\|\text{prox}_r(x) - \text{prox}_r(y)\|_2 \leq \|x - y\|_2$$

Proof

1. Let $u = \text{prox}_r(x)$, and $v = \text{prox}_r(y)$. Then, from the previous property:

$$\langle x - u, z_1 - u \rangle \leq r(z_1) - r(u)$$
$$\langle y - v, z_2 - v \rangle \leq r(z_2) - r(v).$$

2. Substitute $z_1 = v$ and $z_2 = u$. Summing up, we get:

$$\langle x - u, v - u \rangle + \langle y - v, u - v \rangle \leq 0,$$
$$\langle x - y, v - u \rangle + \|v - u\|^2 \leq 0.$$

Proximal operator properties

Theorem

The operator $\text{prox}_r(x)$ is firmly nonexpansive (FNE)

$$\|\text{prox}_r(x) - \text{prox}_r(y)\|_2^2 \leq \langle \text{prox}_r(x) - \text{prox}_r(y), x - y \rangle$$

and nonexpansive:

$$\|\text{prox}_r(x) - \text{prox}_r(y)\|_2 \leq \|x - y\|_2$$

Proof

1. Let $u = \text{prox}_r(x)$, and $v = \text{prox}_r(y)$. Then, from the previous property:

$$\langle x - u, z_1 - u \rangle \leq r(z_1) - r(u)$$
$$\langle y - v, z_2 - v \rangle \leq r(z_2) - r(v).$$

2. Substitute $z_1 = v$ and $z_2 = u$. Summing up, we get:

$$\langle x - u, v - u \rangle + \langle y - v, u - v \rangle \leq 0,$$
$$\langle x - y, v - u \rangle + \|v - u\|^2 \leq 0.$$

3. Which is exactly what we need to prove after substitution of u, v.

$$\|u - v\|_2 \leq \langle x - y, u - v \rangle$$

Proximal operator properties

Theorem

The operator $\text{prox}_r(x)$ is firmly nonexpansive (FNE)

$$\|\text{prox}_r(x) - \text{prox}_r(y)\|_2^2 \leq \langle \text{prox}_r(x) - \text{prox}_r(y), x - y \rangle$$

$$\leq \|\text{prox}_r(x) - \text{prox}_r(y)\| \cdot \|x - y\|$$

and nonexpansive:

$$\|\text{prox}_r(x) - \text{prox}_r(y)\|_2 \leq \|x - y\|_2$$

Proof

1. Let $u = \text{prox}_r(x)$, and $v = \text{prox}_r(y)$. Then, from the previous property:

$$\langle x - u, z_1 - u \rangle \leq r(z_1) - r(u)$$
$$\langle y - v, z_2 - v \rangle \leq r(z_2) - r(v).$$

2. Substitute $z_1 = v$ and $z_2 = u$. Summing up, we get:

$$\langle x - u, v - u \rangle + \langle y - v, u - v \rangle \leq 0,$$
$$\langle x - y, v - u \rangle + \|v - u\|_2^2 \leq 0.$$

3. Which is exactly what we need to prove after substitution of u, v.

$$\|u - v\|_2^2 \leq \langle x - y, u - v \rangle$$

4. The last point comes from simple Cauchy-Bunyakovsky-Schwarz for the last inequality.

Proximal operator properties

Theorem

Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ and $r : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be convex functions. Additionally, assume that f is continuously differentiable and L-smooth, and for r, prox_r is defined. Then, x^* is a solution to the composite optimization problem if and only if, for any $\alpha > 0$, it satisfies:

$$x^* = \text{prox}_{r,\alpha}(x^* - \alpha \nabla f(x^*))$$

Proof

1. Optimality conditions:

Proximal operator properties

Theorem

Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ and $r : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be convex functions. Additionally, assume that f is continuously differentiable and L-smooth, and for r, prox_r is defined. Then, x^* is a solution to the composite optimization problem if and only if, for any $\alpha > 0$, it satisfies:

$$x^* = \text{prox}_{r,\alpha}(x^* - \alpha \nabla f(x^*))$$

Proof

1. Optimality conditions:

$$0 \in \nabla f(x^*) + \partial r(x^*)$$

Proximal operator properties

Theorem

Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ and $r : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be convex functions. Additionally, assume that f is continuously differentiable and L-smooth, and for r, prox_r is defined. Then, x^* is a solution to the composite optimization problem if and only if, for any $\alpha > 0$, it satisfies:

$$x^* = \text{prox}_{r,\alpha}(x^* - \alpha \nabla f(x^*))$$

Proof

1. Optimality conditions:

$$\begin{aligned} 0 \in& \nabla f(x^*) + \partial r(x^*) \\ -\alpha \nabla f(x^*) \in& \alpha \partial r(x^*) \end{aligned}$$

Proximal operator properties

Theorem

Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ and $r : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be convex functions. Additionally, assume that f is continuously differentiable and L-smooth, and for r, prox_r is defined. Then, x^* is a solution to the composite optimization problem if and only if, for any $\alpha > 0$, it satisfies:

$$x^* = \text{prox}_{r,\alpha}(x^* - \alpha \nabla f(x^*))$$

Proof

1. Optimality conditions:

$$\begin{aligned} 0 \in& \nabla f(x^*) + \partial r(x^*) \\ -\alpha \nabla f(x^*) \in& \alpha \partial r(x^*) \\ x^* - \alpha \nabla f(x^*) - x^* \in& \alpha \partial r(x^*) \end{aligned}$$

УМНЫЙ НОЛЬ

Proximal operator properties

Theorem

Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ and $r : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be convex functions. Additionally, assume that f is continuously differentiable and L-smooth, and for r, prox_r is defined. Then, x^* is a solution to the composite optimization problem if and only if, for any $\alpha > 0$, it satisfies:

$$x^* = \text{prox}_{r,\alpha}(x^* - \alpha \nabla f(x^*))$$

Proof

1. Optimality conditions:

$$\begin{aligned} 0 \in& \nabla f(x^*) + \partial r(x^*) \\ - \alpha \nabla f(x^*) \in& \alpha \partial r(x^*) \\ x^* - \alpha \nabla f(x^*) - x^* \in& \alpha \partial r(x^*) \end{aligned}$$

2. Recall from the previous lemma:

$$\text{prox}_r(x) = y \Leftrightarrow x - y \in \partial r(y)$$

$x - y \in \partial r(x^*)$

Proximal operator properties

Theorem

Let $f : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ and $r : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ be convex functions. Additionally, assume that f is continuously differentiable and L-smooth, and for r, prox_r is defined. Then, x^* is a solution to the composite optimization problem if and only if, for any $\alpha > 0$, it satisfies:

$$x^* = \text{prox}_{r,\alpha}(x^* - \alpha\nabla f(x^*))$$

Proof

$\partial\varphi(x^*)$ $\quad \varphi(x) = f(x) + r(x)$

1. Optimality conditions:

$$\begin{aligned} 0 &\in \nabla f(x^*) + \partial r(x^*) \\ -\alpha\nabla f(x^*) &\in \alpha\partial r(x^*) \\ x^* - \alpha\nabla f(x^*) - x^* &\in \alpha\partial r(x^*) \end{aligned}$$

2. Recall from the previous lemma:

$$\text{prox}_r(x) = y \Leftrightarrow x - y \in \partial r(y)$$

3. Finally,

$$x^* = \text{prox}_{\alpha r}(x^* - \alpha\nabla f(x^*)) = \text{prox}_{r,\alpha}(x^* - \alpha\nabla f(x^*))$$

Convergence

Theorem

Consider the proximal gradient method

$$x_{k+1} = \mathrm{prox}_{\alpha r}\left(x_k - \alpha \nabla f(x_k)\right)$$

For the criterion $\varphi(x) = f(x) + r(x)$, we assume:

- f is convex, differentiable, $\mathrm{dom}(f) = \mathbb{R}^n$, and ∇f is Lipschitz continuous with constant $L > 0$.

Proximal gradient descent has a convergence rate of $O(1/k)$ or $O(1/\varepsilon)$. This matches the gradient descent rate! (But remember the proximal operation cost)

Convergence

Theorem

Consider the proximal gradient method

$$x_{k+1} = \text{prox}_{\alpha r}\left(x_k - \alpha \nabla f(x_k)\right)$$

For the criterion $\varphi(x) = f(x) + r(x)$, we assume:

- f is convex, differentiable, $\text{dom}(f) = \mathbb{R}^n$, and ∇f is Lipschitz continuous with constant $L > 0$.
- r is convex, and $\text{prox}_{\alpha r}(x_k) = \arg\min_{x \in \mathbb{R}^n} \left[\alpha r(x) + \frac{1}{2}\|x - x_k\|_2^2\right]$ can be evaluated.

Proximal gradient descent has a convergence rate of $O(1/k)$ or $O(1/\varepsilon)$. This matches the gradient descent rate! (But remember the proximal operation cost)

Convergence

Theorem

Consider the proximal gradient method

$$x_{k+1} = \text{prox}_{\alpha r}\left(x_k - \alpha \nabla f(x_k)\right)$$

For the criterion $\varphi(x) = f(x) + r(x)$, we assume:

- f is convex, differentiable, $\text{dom}(f) = \mathbb{R}^n$, and ∇f is Lipschitz continuous with constant $L > 0$.
- r is convex, and $\text{prox}_{\alpha r}(x_k) = \text{arg} \min_{x \in \mathbb{R}^n} \left[\alpha r(x) + \frac{1}{2}\|x - x_k\|_2^2\right]$ can be evaluated.

Proximal gradient descent has a convergence rate of $O(1/k)$ or $O(1/\varepsilon)$. This matches the gradient descent rate! (But remember the proximal operation cost)

Convergence

Theorem

проксимальный градиент метод

Consider the proximal gradient method

$$x_{k+1} = \text{prox}_{\alpha r}\left(x_k - \alpha \nabla f(x_k)\right)$$

For the criterion $\varphi(x) = f(x) + r(x)$, we assume:

- f is convex, differentiable, $\text{dom}(f) = \mathbb{R}^n$, and ∇f is Lipschitz continuous with constant $L > 0$.
- r is convex, and $\text{prox}_{\alpha r}(x_k) = \text{arg}\min\limits_{x \in \mathbb{R}^n} \left[\alpha r(x) + \frac{1}{2}\|x - x_k\|_2^2\right]$ can be evaluated.

Proximal gradient descent with fixed step size $\alpha = 1/L$ satisfies

$$\varphi(x^{(k)}) - \varphi^* \leq \frac{L\|x^{(0)} - x^*\|^2}{2k},$$

Convergence

Theorem

Consider the proximal gradient method

$$x_{k+1} = \text{prox}_{\alpha r}\left(x_k - \alpha \nabla f(x_k)\right)$$

For the criterion $\varphi(x) = f(x) + r(x)$, we assume:

- f is convex, differentiable, $\text{dom}(f) = \mathbb{R}^n$, and ∇f is Lipschitz continuous with constant $L > 0$.
- r is convex, and $\text{prox}_{\alpha r}(x_k) = \text{arg}\min\limits_{x \in \mathbb{R}^n} \left[\alpha r(x) + \frac{1}{2}\|x - x_k\|_2^2\right]$ can be evaluated.

Proximal gradient descent with fixed step size $\alpha = 1/L$ satisfies

$$\varphi(x^{(k)}) - \varphi^* \leq \frac{L\|x^{(0)} - x^*\|^2}{2k},$$

Proximal gradient descent has a convergence rate of $O(1/k)$ or $O(1/\varepsilon)$. This matches the gradient descent rate! (But remember the proximal operation cost)

Convergence

Accelerated Proximal Method

Accelerated Proximal Method

Let $x_0 = y_0 \in \text{dom}(r)$. For $k \geq 1$:

$$
\begin{aligned}
x_k &= \text{prox}_{\alpha_k h}(y_{k-1} - \alpha_k \nabla f(y_{k-1})) \\
y_k &= x_k + \frac{k-1}{k+2}(x_k - x_{k-1})
\end{aligned}
$$

Achieves

$$
\varphi(x_k) - \varphi^* \leq \frac{2L\|x_0 - x^*\|^2}{k^2}.
$$

Accelerated Proximal Method

Accelerated Proximal Method

Let $x_0 = y_0 \in \text{dom}(r)$. For $k \geq 1$:

$$x_k = \text{prox}_{\alpha_k h}(y_{k-1} - \alpha_k \nabla f(y_{k-1}))$$
$$y_k = x_k + \frac{k-1}{k+2}(x_k - x_{k-1})$$

Achieves

$$\varphi(x_k) - \varphi^* \leq \frac{2L\|x_0 - x^*\|^2}{k^2}.$$

Framework due to: Nesterov (1983, 2004); also Beck, Teboulle (2009). Simplified analysis: Tseng (2008).

- Uses extra "memory" for interpolation

Accelerated Proximal Method

Accelerated Proximal Method

Let $x_0 = y_0 \in \text{dom}(r)$. For $k \geq 1$:

$$x_k = \text{prox}_{\alpha_k h}(y_{k-1} - \alpha_k \nabla f(y_{k-1}))$$
$$y_k = x_k + \frac{k-1}{k+2}(x_k - x_{k-1})$$

Achieves

$$\varphi(x_k) - \varphi^* \leq \frac{2L\|x_0 - x^*\|^2}{k^2}.$$

Framework due to: Nesterov (1983, 2004); also Beck, Teboulle (2009). Simplified analysis: Tseng (2008).

- Uses extra "memory" for interpolation
- Same computational cost as ordinary prox-grad

Accelerated Proximal Method

Accelerated Proximal Method

Let $x_0 = y_0 \in \text{dom}(r)$. For $k \geq 1$:

$$
\begin{aligned}
x_k &= \text{prox}_{\alpha_k h}(y_{k-1} - \alpha_k \nabla f(y_{k-1})) \\
y_k &= x_k + \frac{k-1}{k+2}(x_k - x_{k-1})
\end{aligned}
$$

Achieves

$$
\varphi(x_k) - \varphi^* \leq \frac{2L\|x_0 - x^*\|^2}{k^2}.
$$

Framework due to: Nesterov (1983, 2004); also Beck, Teboulle (2009). Simplified analysis: Tseng (2008).

- Uses extra "memory" for interpolation
- Same computational cost as ordinary prox-grad
- Convergence rate theoretically optimal

Example: ISTA

Iterative Shrinkage-Thresholding Algorithm (ISTA)

ISTA is a popular method for solving optimization problems involving L1 regularization, such as Lasso. It combines gradient descent with a shrinkage operator to handle the non-smooth L1 penalty effectively.

- **Algorithm**:

Example: ISTA

Iterative Shrinkage-Thresholding Algorithm (ISTA)

ISTA is a popular method for solving optimization problems involving L1 regularization, such as Lasso. It combines gradient descent with a shrinkage operator to handle the non-smooth L1 penalty effectively.

- **Algorithm**:
  - Given x_0, for $k \geq 0$, repeat:

$$x_{k+1} = \text{prox}_{\lambda \alpha \|\cdot\|_1} \left(x_k - \alpha \nabla f(x_k) \right),$$

where $\text{prox}_{\lambda \alpha \|\cdot\|_1}(v)$ applies soft thresholding to each component of v.

Example: ISTA

Iterative Shrinkage-Thresholding Algorithm (ISTA)

ISTA is a popular method for solving optimization problems involving L1 regularization, such as Lasso. It combines gradient descent with a shrinkage operator to handle the non-smooth L1 penalty effectively.

- **Algorithm**:
  - Given x_0, for $k \geq 0$, repeat:

$$x_{k+1} = \text{prox}_{\lambda\alpha\|\cdot\|_1}\left(x_k - \alpha \nabla f(x_k)\right),$$

  where $\text{prox}_{\lambda\alpha\|\cdot\|_1}(v)$ applies soft thresholding to each component of v.
- **Convergence**:

Example: ISTA

Iterative Shrinkage-Thresholding Algorithm (ISTA)

ISTA is a popular method for solving optimization problems involving L1 regularization, such as Lasso. It combines gradient descent with a shrinkage operator to handle the non-smooth L1 penalty effectively.

- **Algorithm**:
  - Given x_0, for $k \geq 0$, repeat:

$$x_{k+1} = \text{prox}_{\lambda\alpha\|\cdot\|_1}\left(x_k - \alpha\nabla f(x_k)\right),$$

  where $\text{prox}_{\lambda\alpha\|\cdot\|_1}(v)$ applies soft thresholding to each component of v.
- **Convergence**:
  - Converges at a rate of $O(1/k)$ for suitable step size α.

Example: ISTA

Iterative Shrinkage-Thresholding Algorithm (ISTA)

ISTA is a popular method for solving optimization problems involving L1 regularization, such as Lasso. It combines gradient descent with a shrinkage operator to handle the non-smooth L1 penalty effectively.

- **Algorithm**:
  - Given x_0, for $k \geq 0$, repeat:

$$x_{k+1} = \text{prox}_{\lambda\alpha\|\cdot\|_1}\left(x_k - \alpha\nabla f(x_k)\right),$$

    where $\text{prox}_{\lambda\alpha\|\cdot\|_1}(v)$ applies soft thresholding to each component of v.
- **Convergence**:
  - Converges at a rate of $O(1/k)$ for suitable step size α.
- **Application**:

Example: ISTA

Iterative Shrinkage-Thresholding Algorithm (ISTA)

ISTA is a popular method for solving optimization problems involving L1 regularization, such as Lasso. It combines gradient descent with a shrinkage operator to handle the non-smooth L1 penalty effectively.

- **Algorithm**:
  - Given x_0, for $k \geq 0$, repeat:
$$x_{k+1} = \text{prox}_{\lambda\alpha\|\cdot\|_1}\left(x_k - \alpha \nabla f(x_k)\right),$$
    where $\text{prox}_{\lambda\alpha\|\cdot\|_1}(v)$ applies soft thresholding to each component of v.
- **Convergence**:
  - Converges at a rate of $O(1/k)$ for suitable step size α.
- **Application**:
  - Efficient for sparse signal recovery, image processing, and compressed sensing.

Example: FISTA

Fast Iterative Shrinkage-Thresholding Algorithm (FISTA)

FISTA improves upon ISTA's convergence rate by incorporating a momentum term, inspired by Nesterov's accelerated gradient method.

- **Algorithm**:

Example: FISTA

Fast Iterative Shrinkage-Thresholding Algorithm (FISTA)

FISTA improves upon ISTA's convergence rate by incorporating a momentum term, inspired by Nesterov's accelerated gradient method.

- **Algorithm**:
  - Initialize $x_0 = y_0$, $t_0 = 1$.

Example: FISTA

Fast Iterative Shrinkage-Thresholding Algorithm (FISTA)

FISTA improves upon ISTA's convergence rate by incorporating a momentum term, inspired by Nesterov's accelerated gradient method.

- **Algorithm**:
  - Initialize $x_0 = y_0$, $t_0 = 1$.
  - For $k \geq 1$, update:

$$
\begin{aligned}
x_k &= \text{prox}_{\lambda\alpha\|\cdot\|_1}\left(y_{k-1} - \alpha\nabla f(y_{k-1})\right), \\
t_k &= \frac{1 + \sqrt{1 + 4t_{k-1}^2}}{2}, \\
y_k &= x_k + \frac{t_{k-1} - 1}{t_k}(x_k - x_{k-1}).
\end{aligned}
$$

Example: FISTA

Fast Iterative Shrinkage-Thresholding Algorithm (FISTA)

FISTA improves upon ISTA's convergence rate by incorporating a momentum term, inspired by Nesterov's accelerated gradient method.

- **Algorithm**:
  - Initialize $x_0 = y_0$, $t_0 = 1$.
  - For $k \geq 1$, update:

$$
\begin{aligned}
x_k &= \text{prox}_{\lambda\alpha\|\cdot\|_1}\left(y_{k-1} - \alpha\nabla f(y_{k-1})\right), \\
t_k &= \frac{1 + \sqrt{1 + 4t_{k-1}^2}}{2}, \\
y_k &= x_k + \frac{t_{k-1} - 1}{t_k}(x_k - x_{k-1}).
\end{aligned}
$$

- **Convergence**:

Example: FISTA

Fast Iterative Shrinkage-Thresholding Algorithm (FISTA)

FISTA improves upon ISTA's convergence rate by incorporating a momentum term, inspired by Nesterov's accelerated gradient method.

- **Algorithm**:
    - Initialize $x_0 = y_0$, $t_0 = 1$.
    - For $k \geq 1$, update:

$$
\begin{aligned}
x_k &= \text{prox}_{\lambda\alpha\|\cdot\|_1}\left(y_{k-1} - \alpha\nabla f(y_{k-1})\right), \\
t_k &= \frac{1 + \sqrt{1 + 4t_{k-1}^2}}{2}, \\
y_k &= x_k + \frac{t_{k-1} - 1}{t_k}(x_k - x_{k-1}).
\end{aligned}
$$

- **Convergence**:
    - Improves the convergence rate to $O(1/k^2)$.

Example: FISTA

Fast Iterative Shrinkage-Thresholding Algorithm (FISTA)

FISTA improves upon ISTA's convergence rate by incorporating a momentum term, inspired by Nesterov's accelerated gradient method.

- **Algorithm**:
  - Initialize $x_0 = y_0$, $t_0 = 1$.
  - For $k \geq 1$, update:

$$
\begin{aligned}
x_k &= \text{prox}_{\lambda\alpha\|\cdot\|_1}\left(y_{k-1} - \alpha\nabla f(y_{k-1})\right), \\
t_k &= \frac{1 + \sqrt{1 + 4t_{k-1}^2}}{2}, \\
y_k &= x_k + \frac{t_{k-1} - 1}{t_k}(x_k - x_{k-1}).
\end{aligned}
$$

- **Convergence**:
  - Improves the convergence rate to $O(1/k^2)$.
- **Application**:

Example: FISTA

Fast Iterative Shrinkage-Thresholding Algorithm (FISTA)

FISTA improves upon ISTA's convergence rate by incorporating a momentum term, inspired by Nesterov's accelerated gradient method.

- **Algorithm**:
  - Initialize $x_0 = y_0$, $t_0 = 1$.
  - For $k \geq 1$, update:

$$
\begin{aligned}
x_k &= \text{prox}_{\lambda\alpha\|\cdot\|_1}\left(y_{k-1} - \alpha\nabla f(y_{k-1})\right), \\
t_k &= \frac{1 + \sqrt{1 + 4t_{k-1}^2}}{2}, \\
y_k &= x_k + \frac{t_{k-1} - 1}{t_k}(x_k - x_{k-1}).
\end{aligned}
$$

- **Convergence**:
  - Improves the convergence rate to $O(1/k^2)$.
- **Application**:
  - Especially useful for large-scale problems in machine learning and signal processing where the L1 penalty induces sparsity.

Example: Matrix Completion

$$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \qquad P_\Omega(A) = \begin{pmatrix} 0 & 2 \\ 0 & 0 \end{pmatrix}$$

Solving the Matrix Completion Problem

Matrix completion problems seek to fill in the missing entries of a partially observed matrix under certain assumptions, typically low-rank. This can be formulated as a minimization problem involving the nuclear norm (sum of singular values), which promotes low-rank solutions.

- **Problem Formulation**:

$$\min_X \frac{1}{2}\|P_\Omega(X) - P_\Omega(M)\|_F^2 + \lambda \|X\|_*,$$

where P_Ω projects onto the observed set Ω, and $\|\cdot\|_*$ denotes the nuclear norm.

$$\|X\|_* = \sup_{\|y\|_1 \neq 0} \frac{\|Xy\|_1}{\|y\|_1} = \ldots = \sum_{i=1}^n \sigma_i(X)$$

Example: Matrix Completion

Solving the Matrix Completion Problem

Matrix completion problems seek to fill in the missing entries of a partially observed matrix under certain assumptions, typically low-rank. This can be formulated as a minimization problem involving the nuclear norm (sum of singular values), which promotes low-rank solutions.

- **Problem Formulation**:

$$\min_X \frac{1}{2}\|P_\Omega(X) - P_\Omega(M)\|_F^2 + \lambda\|X\|_*,$$

  where P_Ω projects onto the observed set Ω, and $\|\cdot\|_*$ denotes the nuclear norm.
- **Proximal Operator**:

Example: Matrix Completion

Solving the Matrix Completion Problem

Matrix completion problems seek to fill in the missing entries of a partially observed matrix under certain assumptions, typically low-rank. This can be formulated as a minimization problem involving the nuclear norm (sum of singular values), which promotes low-rank solutions.

- **Problem Formulation**:

$$\min_X \frac{1}{2}\|P_\Omega(X) - P_\Omega(M)\|_F^2 + \lambda\|X\|_*,$$

  where P_Ω projects onto the observed set Ω, and $\|\cdot\|_*$ denotes the nuclear norm.
- **Proximal Operator**:
  - The proximal operator for the nuclear norm involves singular value decomposition (SVD) and soft-thresholding of the singular values.

Example: Matrix Completion

Solving the Matrix Completion Problem

Matrix completion problems seek to fill in the missing entries of a partially observed matrix under certain assumptions, typically low-rank. This can be formulated as a minimization problem involving the nuclear norm (sum of singular values), which promotes low-rank solutions.

- **Problem Formulation**:

$$
\min_X \frac{1}{2}\|P_\Omega(X) - P_\Omega(M)\|_F^2 + \lambda\|X\|_*,
$$

  where P_Ω projects onto the observed set Ω, and $\|\cdot\|_*$ denotes the nuclear norm.
- **Proximal Operator**:
  - The proximal operator for the nuclear norm involves singular value decomposition (SVD) and soft-thresholding of the singular values.
- **Algorithm**:

Example: Matrix Completion

Solving the Matrix Completion Problem

Matrix completion problems seek to fill in the missing entries of a partially observed matrix under certain assumptions, typically low-rank. This can be formulated as a minimization problem involving the nuclear norm (sum of singular values), which promotes low-rank solutions.

- **Problem Formulation**:

$$\min_X \frac{1}{2}\|P_\Omega(X) - P_\Omega(M)\|_F^2 + \lambda\|X\|_*,$$

  where P_Ω projects onto the observed set Ω, and $\|\cdot\|_*$ denotes the nuclear norm.
- **Proximal Operator**:
  - The proximal operator for the nuclear norm involves singular value decomposition (SVD) and soft-thresholding of the singular values.
- **Algorithm**:
  - Similar proximal gradient or accelerated proximal gradient methods can be applied, where the main computational effort lies in performing partial SVDs.

Example: Matrix Completion

Solving the Matrix Completion Problem

Matrix completion problems seek to fill in the missing entries of a partially observed matrix under certain assumptions, typically low-rank. This can be formulated as a minimization problem involving the nuclear norm (sum of singular values), which promotes low-rank solutions.

- **Problem Formulation**:

$$\min_X \frac{1}{2}\|P_\Omega(X) - P_\Omega(M)\|_F^2 + \lambda\|X\|_*,$$

  where P_Ω projects onto the observed set Ω, and $\|\cdot\|_*$ denotes the nuclear norm.
- **Proximal Operator**:
  - The proximal operator for the nuclear norm involves singular value decomposition (SVD) and soft-thresholding of the singular values.
- **Algorithm**:
  - Similar proximal gradient or accelerated proximal gradient methods can be applied, where the main computational effort lies in performing partial SVDs.
- **Application**:

Example: Matrix Completion

Solving the Matrix Completion Problem

Matrix completion problems seek to fill in the missing entries of a partially observed matrix under certain assumptions, typically low-rank. This can be formulated as a minimization problem involving the nuclear norm (sum of singular values), which promotes low-rank solutions.

- **Problem Formulation**:

$$\min_X \frac{1}{2}\|P_\Omega(X) - P_\Omega(M)\|_F^2 + \lambda\|X\|_*,$$

  where P_Ω projects onto the observed set Ω, and $\|\cdot\|_*$ denotes the nuclear norm.
- **Proximal Operator**:
  - The proximal operator for the nuclear norm involves singular value decomposition (SVD) and soft-thresholding of the singular values.
- **Algorithm**:
  - Similar proximal gradient or accelerated proximal gradient methods can be applied, where the main computational effort lies in performing partial SVDs.
- **Application**:
  - Widely used in recommender systems, image recovery, and other domains where data is naturally matrix-formed but partially observed.

Summary

- If we exploit the structure of the problem, we may beat the lower bounds for the unstructured problem.

Summary

- If we exploit the structure of the problem, we may beat the lower bounds for the unstructured problem.
- Proximal gradient method for a composite problem with an L-smooth convex function f and a convex proximal friendly function r has the same convergence as the gradient descent method for the function f. The smoothness/non-smoothness properties of r do not affect convergence.

Summary

- If we exploit the structure of the problem, we may beat the lower bounds for the unstructured problem.
- Proximal gradient method for a composite problem with an L-smooth convex function f and a convex proximal friendly function r has the same convergence as the gradient descent method for the function f. The smoothness/non-smoothness properties of r do not affect convergence.
- It seems that by putting $f = 0$, any nonsmooth problem can be solved using such a method. Question: is this true?

$$\min_x \left(\underbrace{\max(x)}_{r(x)} + 0 \right)$$

$$r(\tilde{x}) + \frac{1}{2}\|x - \tilde{x}\|_2^2$$

Summary

- If we exploit the structure of the problem, we may beat the lower bounds for the unstructured problem.
- Proximal gradient method for a composite problem with an L-smooth convex function f and a convex proximal friendly function r has the same convergence as the gradient descent method for the function f. The smoothness/non-smoothness properties of r do not affect convergence.
- It seems that by putting $f = 0$, any nonsmooth problem can be solved using such a method. Question: is this true?

Summary

- If we exploit the structure of the problem, we may beat the lower bounds for the unstructured problem.
- Proximal gradient method for a composite problem with an L-smooth convex function f and a convex proximal friendly function r has the same convergence as the gradient descent method for the function f. The smoothness/non-smoothness properties of r do not affect convergence.
- It seems that by putting $f = 0$, any nonsmooth problem can be solved using such a method. Question: is this true?

  If we allow the proximal operator to be inexact (numerically), then it is true that we can solve any nonsmooth optimization problem. But this is not better from the point of view of theory than solving the problem by subgradient descent, because some auxiliary method (for example, the same subgradient descent) is used to solve the proximal subproblem.
- Proximal method is a general modern framework for many numerical methods. Further development includes accelerated, stochastic, primal-dual modifications and etc.

Summary

- If we exploit the structure of the problem, we may beat the lower bounds for the unstructured problem.
- Proximal gradient method for a composite problem with an L-smooth convex function f and a convex proximal friendly function r has the same convergence as the gradient descent method for the function f. The smoothness/non-smoothness properties of r do not affect convergence.
- It seems that by putting $f = 0$, any nonsmooth problem can be solved using such a method. Question: is this true?

  If we allow the proximal operator to be inexact (numerically), then it is true that we can solve any nonsmooth optimization problem. But this is not better from the point of view of theory than solving the problem by subgradient descent, because some auxiliary method (for example, the same subgradient descent) is used to solve the proximal subproblem.
- Proximal method is a general modern framework for many numerical methods. Further development includes accelerated, stochastic, primal-dual modifications and etc.
- Further reading: Proximal operator splitting, Douglas-Rachford splitting, Best approximation problem, Three operator splitting.