Proximal Gradient Method

Pontus Giselsson
Learning goals

• Know the difference between first and second order methods
• Know the proximal gradient method:
  • Know that it is (sometimes) a majorization-minimization method
  • Understand its relation to the descent lemma
  • Understand the conditions for convergence and convergence proof
  • Understand what it converges to in nonconvex and convex settings
  • Able to show that the fixed-points solves the problem if convex
Optimization algorithm overview

Algorithms can roughly be divided into the following classes:

- Second-order methods
- Quasi second-order methods
- First-order methods
- Stochastic and coordinate-wise first-order methods
Second-order methods

• Solves problems using second-order (Hessian) information
• Requires smooth (twice continuously differentiable) functions
• Constraints can be incorporated via barrier functions
• Examples:
  • Newton’s method to minimize smooth function $f$:

$$x_{k+1} = x_k - \gamma_k (\nabla^2 f(x_k))^{-1} \nabla f(x_k)$$

• Interior points methods for smooth constrained problems:
  • Use sequence of smooth constraint barrier functions
  • For each barrier, solve smooth problem using Newton’s method
  • Make barriers increasingly well approximate constraint set
  • (Can be applied to directly solve primal-dual optimality condition)

• Computational backbone: solving linear systems $O(n^3)$
• Often restricted to small to medium scale problems
Quasi second-order methods

- Estimates second-order information from first-order
- Solves problems using estimated second-order information
- Requires smooth (twice continuously differentiable) functions
- Quasi-Newton method for smooth $f$

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

where $B_k$ is:
- estimate of Hessian inverse (not Hessian to avoid later inverse)
- cheaply computed from gradient information

- Computational backbone: forming $B_k$ and matrix multiplication
- Can solve large-scale smooth problems
First-order methods

- Solves problems using first-order (sub-gradient) information
- Computational primitives: gradients and proximal operators
- Use gradient if function differentiable, prox if nondifferentiable
- Examples for solving \( \min_x f(x) + g(x) \)
  - Proximal gradient method (requires smooth \( f \) since gradient used)
    \[
    x_{k+1} = \text{prox}_{\gamma g}(x_k - \gamma \nabla f(x_k))
    \]
  - Douglas-Rachford splitting (no smoothness requirement)
    \[
    z_{k+1} = \frac{1}{2} z_k + \frac{1}{2} (2\text{prox}_{\gamma g} - I)(2\text{prox}_{\gamma f} - I)z_k
    \]
    and \( x_k = \text{prox}_{\gamma f}(z_k) \) converges to solution
- Iteration often cheaper than second-order if function split wisely
- Can solve large scale problems
Stochastic and coordinate-wise first-order methods

• Sometimes first-order methods computationally too expensive
• Stochastic gradient methods:
  • Use stochastic approximation of gradient
  • For finite sum problems, cheaply computed approximation exists
• Coordinate-wise updates:
  • Update only one (or block of) coordinates in every iteration:
    • via direct minimization
    • via proximal gradient step
  • Can update coordinates in cyclic fashion
  • Stronger convergence results if random selection of block
  • Efficiently evaluated, e.g., if one function separable
• Can solve huge scale problems
Our focus

Proximal gradient method, stochastic and coordinate-wise versions

Lectures will cover:

• Proximal gradient method
• Coordinate and stochastic proximal gradient method
• Line search, acceleration, and scaling
• Newton prox method, early termination, quasi-Newton
Notation

• Will go back to optimization variable notation: $x, y, z$
• For learning examples, use machine learning notation: $\theta = (w, b)$
Proximal Gradient Method
Majorization Minimization

- Proximal gradient is (often) majorization minimization algorithm
- Majorization minimization for solving $\min_{x} f(x)$:
  - Let iterate be $x_k$
  - Find at $x_k$ majorizing function $\bar{f}_{x_k}$ such that
    
    \[
    \bar{f}_{x_k} \geq f \quad \text{and} \quad \bar{f}_{x_k}(x_k) = f(x_k)
    \]
  - Minimize $\bar{f}$ (easier than minimizing $f$) to get next iterate
    
    \[
    x_{k+1} = \arg\min_{x} \bar{f}_{x_k}(x)
    \]
  - Majorizer should ensure $x_{k+1} = x_k$ if and only if $x_k$ minimizes $f$
  - Guarantees function decrease (maybe not $x_k \to x \in \arg\min f$)
Majorization Minimization

• Proximal gradient is (often) majorization minimization algorithm
• Majorization minimization for solving \( \min_x f(x) \):
  • Let iterate be \( x_k \)
  • Find at \( x_k \) majorizing function \( \bar{f}_{x_k} \) such that \( \bar{f}_{x_k} \geq f \) and \( \bar{f}_{x_k}(x_k) = f(x_k) \)
  • Minimize \( \bar{f} \) (easier than minimizing \( f \)) to get next iterate \( x_{k+1} = \arg\min_x \bar{f}_{x_k}(x) \)
  • Majorizer should ensure \( x_{k+1} = x_k \) if and only if \( x_k \) minimizes \( f \)
  • Guarantees function decrease (maybe not \( x_k \to x \in \arg\min f \)
Majorization Minimization

- Proximal gradient is (often) majorization minimization algorithm
- Majorization minimization for solving $\min_x f(x)$:
  - Let iterate be $x_k$
  - Find at $x_k$ majorizing function $\bar{f}_{x_k}$ such that
    \[ \bar{f}_{x_k} \geq f \quad \text{and} \quad \bar{f}_{x_k}(x_k) = f(x_k) \]
  - Minimize $\bar{f}$ (easier than minimizing $f$) to get next iterate
    \[ x_{k+1} = \arg\min_x \bar{f}_{x_k}(x) \]
  - Majorizer should ensure $x_{k+1} = x_k$ if and only if $x_k$ minimizes $f$
  - Guarantees function decrease (maybe not $x_k \to x \in \arg\min f$)
Majorization Minimization

• Proximal gradient is (often) majorization minimization algorithm
• Majorization minimization for solving \( \min_x f(x) \):
  • Let iterate be \( x_k \)
  • Find at \( x_k \) majorizing function \( \overline{f}_{x_k} \) such that
    \[
    \overline{f}_{x_k} \geq f \quad \text{and} \quad \overline{f}_{x_k}(x_k) = f(x_k)
    \]
  • Minimize \( \overline{f} \) (easier than minimizing \( f \)) to get next iterate
    \[
    x_{k+1} = \arg\min_x \overline{f}_{x_k}(x)
    \]
  • Majorizer should ensure \( x_{k+1} = x_k \) if and only if \( x_k \) minimizes \( f \)
  • Guarantees function decrease (maybe not \( x_k \to x \in \arg\min f \))

![Graph showing the majorization minimization process](image-url)
Majorization Minimization

- Proximal gradient is (often) majorization minimization algorithm
- Majorization minimization for solving minimize $f(x)$:
  - Let iterate be $x_k$
  - Find at $x_k$ majorizing function $\bar{f}_{x_k}$ such that
    $$\bar{f}_{x_k} \geq f \quad \text{and} \quad \bar{f}_{x_k}(x_k) = f(x_k)$$
  - Minimize $\bar{f}$ (easier than minimizing $f$) to get next iterate
    $$x_{k+1} = \arg\min_x \bar{f}_{x_k}(x)$$
  - Majorizer should ensure $x_{k+1} = x_k$ if and only if $x_k$ minimizes $f$
  - Guarantees function decrease (maybe not $x_k \rightarrow x \in \arg\min f$)
Majorization Minimization

• Proximal gradient is (often) majorization minimization algorithm
• Majorization minimization for solving minimize $f(x)$:
  • Let iterate be $x_k$
  • Find at $x_k$ majorizing function $\tilde{f}_{x_k}$ such that
    $$\tilde{f}_{x_k} \geq f \quad \text{and} \quad \tilde{f}_{x_k}(x_k) = f(x_k)$$
  • Minimize $\tilde{f}$ (easier than minimizing $f$) to get next iterate
    $$x_{k+1} = \arg\min_x \tilde{f}_{x_k}(x)$$
  • Majorizer should ensure $x_{k+1} = x_k$ if and only if $x_k$ minimizes $f$
  • Guarantees function decrease (maybe not $x_k \to x \in \arg\min f$)
Majorization Minimization

- Proximal gradient is (often) majorization minimization algorithm
- Majorization minimization for solving \( \min_x f(x) \):
  - Let iterate be \( x_k \)
  - Find at \( x_k \) majorizing function \( \bar{f}_{x_k} \) such that
    \[
    \bar{f}_{x_k} \geq f \quad \text{and} \quad \bar{f}_{x_k}(x_k) = f(x_k)
    \]
  - Minimize \( \bar{f} \) (easier than minimizing \( f \)) to get next iterate
    \[
    x_{k+1} = \arg\min_x \bar{f}_{x_k}(x)
    \]
  - Majorizer should ensure \( x_{k+1} = x_k \) if and only if \( x_k \) minimizes \( f \)
  - Guarantees function decrease (maybe not \( x_k \to x \in \arg\min f \))
Composite optimization problems

• We will consider composite optimization problems of the form

\[
\minimize_x f(x) + g(x)
\]

where

• \( f : \mathbb{R}^n \rightarrow \mathbb{R} \) is \( \beta \)-smooth (not necessarily convex)
• \( g : \mathbb{R}^n \rightarrow \mathbb{R} \cup \{ \infty \} \) is closed convex
• Solution set is nonempty, i.e., a solution exists

• Model includes minimization problems of the form

\[
\minimize_x f(Lx) + g(x)
\]

with differentiable \( f : \mathbb{R}^m \rightarrow \mathbb{R} \) and \( L \in \mathbb{R}^{m \times n} \) where

• gradient \( \nabla(f \circ L)(x) = L^T \nabla f(Lx) \)
• \( f \circ L \) is \( \beta \|L\|_2^2 \)-smooth for \( \beta \)-smooth \( f \), \( (\|L\|_2 \) is operator norm)

• The latter is form of most supervised training problems
• The former is used here since lighter notation
Gradient method

• Consider minimize $\beta$-smooth $f : \mathbb{R}^n \to \mathbb{R}$ (i.e., $g = 0$)

• Recall that $\beta$-smoothness implies that

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

for all $x, y \in \mathbb{R}^n$, i.e., r.h.s. is majorizing function for fixed $x$

• Majorization minimization with majorizer if $\gamma_k \in [\epsilon, \beta^{-1}]$, $\epsilon > 0$:

$$x_{k+1} = \arg\min_y \left( f(x_k) + \nabla f(x_k)^T(y - x_k) + \frac{1}{2\gamma_k} \| y - x_k \|_2^2 \right)$$

$$= \arg\min_y \frac{1}{2\gamma_k} \| y - x_k + \gamma_k \nabla f(x_k) \|_2^2$$

$$= x_k - \gamma_k \nabla f(x_k)$$

• Gives gradient method, $\gamma_k$ (bounded above by $\beta^{-1}$) is step length
Longer steps

• The requirement $\gamma_k \in [\epsilon, \frac{1}{\beta}]$ guarantees a majorizer is minimized
• Analysis will say: Possible to have $\gamma_k \in [\epsilon, \frac{2}{\beta} - \epsilon]$: 

\[ x_1, x_2, x_3, x_4, x_5 \]
Longer steps

- The requirement \( \gamma_k \in [\epsilon, \frac{1}{\beta}] \) guarantees a majorizer is minimized
- Analysis will say: Possible to have \( \gamma_k \in [\epsilon, \frac{2}{\beta} - \epsilon] \):
Longer steps

• The requirement $\gamma_k \in [\epsilon, \frac{1}{\beta}]$ guarantees a majorizer is minimized.

• Analysis will say: Possible to have $\gamma_k \in [\epsilon, \frac{2}{\beta} - \epsilon]$.
Longer steps

- The requirement $\gamma_k \in [\epsilon, \frac{1}{\beta}]$ guarantees a majorizer is minimized
- Analysis will say: Possible to have $\gamma_k \in [\epsilon, \frac{2}{\beta} - \epsilon]$:


Longer steps

• The requirement $\gamma_k \in [\epsilon, \frac{1}{\beta}]$ guarantees a majorizer is minimized

• Analysis will say: Possible to have $\gamma_k \in [\epsilon, \frac{2}{\beta} - \epsilon]$:
Longer steps

• The requirement $\gamma_k \in [\epsilon, \frac{1}{\beta}]$ guarantees a majorizer is minimized

• Analysis will say: Possible to have $\gamma_k \in [\epsilon, \frac{2}{\beta} - \epsilon]$: 
Longer steps

• The requirement $\gamma_k \in [\epsilon, \frac{1}{\beta}]$ guarantees a majorizer is minimized
• Analysis will say: Possible to have $\gamma_k \in [\epsilon, \frac{2}{\beta} - \epsilon]$:
Proximal gradient method

• Consider minimize $f(x) + g(x)$ where
  • $f$ is $\beta$-smooth $f : \mathbb{R}^n \rightarrow \mathbb{R}$ (not necessarily convex)
  • $g$ is closed convex
• Due to $\beta$-smoothness of $f$, we have
  \[
  f(y) + g(y) \leq f(x) + \nabla f(x)^T (y - x) + \frac{\beta}{2} \| y - x \|^2_2 + g(y)
  \]
  for all $x, y \in \mathbb{R}^n$, i.e., r.h.s. is majorizing function for fixed $x$
• Majorization minimization with majorizer if $\gamma_k \in [\epsilon, \beta^{-1}]$, $\epsilon > 0$:
  \[
  x_{k+1} = \arg\min_y \left( f(x_k) + \nabla f(x_k)^T (y - x) + \frac{1}{2\gamma_k} \| y - x_k \|^2_2 + g(y) \right)
  \]
  \[
  = \arg\min_y \left( g(y) + \frac{1}{2\gamma_k} \| y - (x_k - \gamma_k \nabla f(x_k)) \|^2_2 \right)
  \]
  \[
  = \text{prox}_{\gamma_k g}(x_k - \gamma_k \nabla f(x_k))
  \]
gives proximal gradient method
Proximal gradient – Example

• Proximal gradient iterations for problem minimize $\frac{1}{2}(x - a)^2 + |x|$
• $f(x) = \frac{1}{2}(x - a)^2$ is smooth term and $g(x) = |x|$ is nonsmooth
• Iteration: $x_{k+1} = \text{prox}_{\gamma g}(x_k - \gamma \nabla f(x_k))$
• Note: convergence in finite number of iterations (not always)
Proximal gradient – Example

• Proximal gradient iterations for problem minimize $\frac{1}{2}(x - a)^2 + |x|$

• $f(x) = \frac{1}{2}(x - a)^2$ is smooth term and $g(x) = |x|$ is nonsmooth

• Iteration: $x_{k+1} = \text{prox}_{\gamma g}(x_k - \gamma \nabla f(x_k))$

• Note: convergence in finite number of iterations (not always)
Proximal gradient – Example

- Proximal gradient iterations for problem minimize $\frac{1}{2}(x-a)^2 + |x|
- $f(x) = \frac{1}{2}(x-a)^2$ is smooth term and $g(x) = |x|$ is nonsmooth
- Iteration: $x_{k+1} = \text{prox}_{\gamma g}(x_k - \gamma \nabla f(x_k))$
- Note: convergence in finite number of iterations (not always)
Proximal gradient – Example

• Proximal gradient iterations for problem minimize $\frac{1}{2}(x - a)^2 + |x|

• $f(x) = \frac{1}{2}(x - a)^2$ is smooth term and $g(x) = |x|$ is nonsmooth

• Iteration: $x_{k+1} = \text{prox}_{\gamma g}(x_k - \gamma \nabla f(x_k))$

• Note: convergence in finite number of iterations (not always)
Proximal gradient – Example

- Proximal gradient iterations for problem minimize $\frac{1}{2}(x - \alpha)^2 + |x|
- $f(x) = \frac{1}{2}(x - \alpha)^2$ is smooth term and $g(x) = |x|$ is nonsmooth
- Iteration: $x_{k+1} = \text{prox}_{\gamma g}(x_k - \gamma \nabla f(x_k))$
- Note: convergence in finite number of iterations (not always)
Proximal gradient – Example

• Proximal gradient iterations for problem minimize $x \frac{1}{2} (x - a)^2 + |x|

• $f(x) = \frac{1}{2} (x - a)^2$ is smooth term and $g(x) = |x|$ is nonsmooth

• Iteration: $x_{k+1} = \text{prox}_{\gamma g}(x_k - \gamma \nabla f(x_k))$

• Note: convergence in finite number of iterations (not always)
Proximal gradient – Special cases

- Proximal gradient method:
  - solves \( \min_x (f(x) + g(x)) \)
  - iteration: \( x_{k+1} = \text{prox}_{\gamma_k g}(x_k - \gamma_k \nabla f(x_k)) \)

- Proximal gradient method with \( g = 0 \):
  - solves \( \min_x f(x) \)
  - \( \text{prox}_{\gamma_k g}(z) = \arg\min_x (0 + \frac{1}{2\gamma} \|x - z\|_2^2) = z \)
  - iteration: \( x_{k+1} = \text{prox}_{\gamma_k g}(x_k - \gamma_k \nabla f(x_k)) = x_k - \gamma_k \nabla f(x_k) \)
  - reduces to gradient method

- Proximal gradient method with \( f = 0 \):
  - solves \( \min_x g(x) \)
  - \( \nabla f(x) = 0 \)
  - iteration: \( x_{k+1} = \text{prox}_{\gamma_k g}(x_k - \gamma_k \nabla f(x_k)) = \text{prox}_{\gamma_k g}(x_k) \)
  - reduces to proximal point method (which is not very useful)
Proximal gradient – Optimality condition

• Proximal gradient iteration:

\[ x_{k+1} = \text{prox}_{\gamma_k g}(x_k - \gamma_k \nabla f(x_k)) \]

\[ = \arg\min_y (g(y) + \frac{1}{2\gamma_k} \|y - (x_k - \gamma_k \nabla f(x_k))\|^2) \]

where \( x_{k+1} \) is unique due to strong convexity of \( h \)

• Fermat’s rule (and since CQ holds) gives optimality condition:

\[ 0 \in \partial g(x_{k+1}) + \partial h(x_{k+1}) \]

\[ = \partial g(x_{k+1}) + \gamma_k^{-1}(x_{k+1} - (x_k - \gamma_k \nabla f(x_k))) \]

\[ = \partial g(x_{k+1}) + \nabla f(x_k) + \gamma_k^{-1}(x_{k+1} - x_k) \]

since \( h \) differentiable

• A consequence: \( \partial g(x_{k+1}) \) is nonempty
To solve minimize $f(x) + g(x)$, an algorithm must:

• have fixed-points (output equals input) that solve problem
• converge to a fixed-point

Proximal gradient method:

• for convex problems, it satisfies both requirements
• for nonconvex, weaker (but still useful) results hold
Proximal gradient – Fixed-point set

• Denote $T_{PG}^{\gamma} := \text{prox}_{\gamma g}(I - \gamma \nabla f)$, gives algorithm $x_{k+1} = T_{PG}^{\gamma} x_k$

• Proximal gradient fixed-point set definition

$$\text{fix}T_{PG}^{\gamma} = \{x : x = T_{PG}^{\gamma} x\} = \{x : x = \text{prox}_{\gamma g}(x - \gamma \nabla f(x))\}$$

i.e., set of points for which $x_{k+1} = x_k$
Proximal gradient – Fixed-point characterization

Let $\gamma > 0$. Then $\bar{x} \in \text{fix}T_{\gamma}^{PGA}$ if and only if $0 \in \partial g(\bar{x}) + \nabla f(\bar{x})$.

- Proof: by proximal gradient step optimality condition

$\bar{x} \in \text{fix}T_{\gamma}^{PGA} \iff \bar{x} = \text{prox}_{\gamma g}(\bar{x} - \gamma \nabla f(\bar{x}))$

$\iff 0 \in \partial g(\bar{x}) + \gamma^{-1}(\bar{x} - (\bar{x} - \gamma \nabla f(\bar{x})))$

$\iff 0 \in \partial g(\bar{x}) + \nabla f(\bar{x})$

- Consequence: fixed-point set same for all $\gamma > 0$
- We call inclusion $0 \in \partial g(\bar{x}) + \nabla f(\bar{x})$ fixed-point characterization
Meaning of fixed-point characterization

• What does fixed-point characterization $0 \in \partial g(\bar{x}) + \nabla f(\bar{x})$ mean?

• For convex differentiable $f$, subdifferential $\partial f(x) = \{\nabla f(x)\}$ and

$$0 \in \partial f(\bar{x}) + \partial g(\bar{x}) = \partial (f + g)(\bar{x})$$

(subdifferential sum rule holds), i.e., fixed-points solve problem

• For nonconvex differentiable $f$, we might have $\partial f(\bar{x}) = \emptyset$
  • Fixed-point are not in general global solutions
  • Points $\bar{x}$ that satisfy $0 \in \partial g(\bar{x}) + \nabla f(\bar{x})$ are called critical points
  • If $g = 0$, the condition is $\nabla f(\bar{x}) = 0$, i.e., a stationary point

• Quality of fixed-points differs

• How about convergence to fixed-point?
Assumptions for convergence – Nonconvex case

• Proximal gradient method $x_{k+1} = \text{prox}_{\gamma_k g}(x_k - \gamma_k \nabla f(x_k))$

• Assumptions:
  
  (i) $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable (not necessarily convex)
  
  (ii) For every $x_k$ and $x_{k+1}$ there exists $\beta_k \in [\eta, \eta^{-1}]$, $\eta \in (0, 1)$:
  
  $$f(x_{k+1}) \leq f(x_k) + \nabla f(x_k)^T (x_{k+1} - x_k) + \frac{\beta_k}{2} \|x_k - x_{k+1}\|^2$$

  where $\beta_k$ is a sort of local Lipschitz constant
  
  (iii) $g : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is closed convex
  
  (iv) A minimizer exists (and $p^* = \min_x (f(x) + g(x))$ is optimal value)
  
  (v) Algorithm parameters $\gamma_k \in [\epsilon, \frac{2}{\beta_k} - \epsilon]$, where $\epsilon > 0$

• Assumption on $f$ satisfied with $\beta_k = \beta$ if $f$ $\beta$-smooth
A basic inequality

Using

(a) Upper bound assumption on \( f \), i.e., Assumption (ii)
(b) Prox optimality condition: There exists \( s_{k+1} \in \partial g(x_{k+1}) \)

\[
0 = s_{k+1} + \gamma_k^{-1}(x_{k+1} - (x_k - \gamma_k \nabla f(x_k)))
\]

(c) Subgradient definition: \( g(x_k) \geq g(x_{k+1}) + s_{k+1}^T (x_k - x_{k+1}) \)

\[
f(x_{k+1}) + g(x_{k+1}) \leq f(x_k) + \nabla f(x_k)^T (x_{k+1} - x_k) + \frac{\beta_k}{2} \| x_{k+1} - x_k \|^2 + g(x_{k+1})
\]

\[
\leq f(x_k) + \nabla f(x_k)^T (x_{k+1} - x_k) + \frac{\beta_k}{2} \| x_{k+1} - x_k \|^2 + g(x_k) - s_{k+1}^T (x_k - x_{k+1})
\]

\[
= f(x_k) + g(x_k) - (\gamma_k^{-1} - \frac{\beta_k}{2}) \| x_{k+1} - x_k \|^2
\]
Function value decrease

- What conclusions can we draw from

\[ f(x_{k+1}) + g(x_{k+1}) \leq f(x_k) + g(x_k) - (\gamma_k^{-1} - \frac{\beta_k}{2})\|x_{k+1} - x_k\|^2 \]

- The requirement on \( \gamma_k \in [\epsilon, \frac{2}{\beta_k} - \epsilon] \):
  - since \( \beta_k \in [\eta, \eta^{-1}] \) there is \( \epsilon > 0 \) such that \([\epsilon, \frac{2}{\beta_k} - \epsilon]\) nonempty
  - therefore \( \delta > 0 \) exists such that

\[ \gamma_k^{-1} \in \left[ \frac{\beta_k}{2} + \delta, \delta^{-1} \right] \quad \Rightarrow \quad \gamma_k^{-1} - \frac{\beta_k}{2} \geq \delta > 0 \]

which implies that function value decreases:

\[ f(x_{k+1}) + g(x_{k+1}) \leq f(x_k) + g(x_k) - \delta\|x_{k+1} - x_k\|^2 \]

- Not very useful!
Fixed-point residual converges

• Rearrange inequality from previous slide:
  \[ \delta \|x_{k+1} - x_k\|_2^2 \leq f(x_k) + g(x_k) - (f(x_{k+1}) + g(x_{k+1})) \]

• Telescope summation gives for all \( n \in \mathbb{N} \):
  \[
  \delta \sum_{k=1}^{n} \|x_{k+1} - x_k\|_2^2 \leq \sum_{k=1}^{n} (f(x_k) + g(x_k) - (f(x_{k+1}) + g(x_{k+1}))) \\
  = f(x_1) + g(x_1) - (f(x_{n+1}) + g(x_{n+1})) \\
  \leq f(x_1) + g(x_1) - p^* < \infty
  
\]
  where \( p^* = \min_x (f(x) + g(x)) \) and \(< \infty \) since \( x_1 \in \text{dom}g \)

• Since \( \delta > 0 \), this implies:
  \[ \|\text{prox}_{\gamma g}(x_k - \gamma \nabla f(x_k)) - x_k\|_2 = \|x_{k+1} - x_k\|_2 \to 0 \]
Residual convergence – Implication

What does $\|\text{prox}_{\gamma_k g}(x_k - \gamma_k \nabla f(x_k)) - x_k\|_2 \to 0$ mean and imply?

- That fixed-point equation will be satisfied in the limit
- By prox-grad optimality condition:

$$\partial g(x_{k+1}) + \nabla f(x_k) \ni \gamma_k^{-1}(x_k - x_{k+1}) \to 0$$

as $k \to \infty$ (since $\gamma_k \geq \epsilon$, i.e., $0 < \gamma_k^{-1} \leq \epsilon^{-1}$) or equivalently

$$\partial g(x_{k+1}) + \nabla f(x_{k+1}) \ni \gamma_k^{-1}(x_k - x_{k+1}) + \underbrace{\nabla f(x_{k+1}) - \nabla f(x_k)}_{u_k} \to 0$$

where $u_k \to 0$ is concluded by continuity of $\nabla f$, implications:

- Fixed-point characterization satisfied in the limit
- Nonconvex $f$: Critical point definition satisfied in the limit
- Convex $f$: Global optimality condition satisfied in the limit
- However, does not imply that $(x_k)$ converges to a fixed-point
Sequence convergence results

Nonconvex $f$:
  - convergent (sub)sequences (if exist), converge to fixed-point

Convex $f$:
  - sequence converges to fixed-point, hence to (global) solution
Sequence convergence – Convex case

- Assume, in addition to previous assumptions, that $f$ is convex
- The following result can be shown to hold

A sequence $(x_k)_{k \in \mathbb{N}}$ converges to a point in $\text{fix} T^{\gamma}_{\text{PG}}$ if:

(i) $\|\text{prox}_{\gamma g}(x_k - \gamma \nabla f(x_k)) - x_k\|_2 \to 0$ as $k \to \infty$

(ii) $(\|x_k - z\|_2)_{k \in \mathbb{N}}$ converges for all $z \in \text{fix} T^{\gamma}_{\text{PG}}$

- Condition (i) already shown to hold for prox-grad iteration
- Condition (ii) holds for convex problems (but not for nonconvex)
- A proof can be found in note on course webpage
Summary

Nonconvex $f$:

- Fixed-points $\bar{x}$ such that $0 \in \partial g(\bar{x}) + \nabla f(\bar{x})$ are critical points
- Generated sequence $u_k \to 0$ satisfies $u_k \in \partial g(x_{k+1}) + \nabla f(x_{k+1})$
- If convergent (sub)sequence exists, converges to fixed-point

Convex $f$:

- Fixed-points $\bar{x}$ such that $0 \in \partial g(\bar{x}) + \nabla f(\bar{x})$ are global solutions
- Generated sequence $u_k \to 0$ satisfies $u_k \in \partial g(x_{k+1}) + \nabla f(x_{k+1})$
- Sequence converges to fixed-point
Choose $\beta_k$ and $\gamma_k$

- Convergence based on assumption that $\beta_k$ known that satisfies
  \[
  f(x_{k+1}) \leq f(x_k) + \nabla f(x_k)^T (x_{k+1} - x_k) + \frac{\beta_k}{2} \|x_k - x_{k+1}\|^2
  \]
  call this descent condition (DC)
- If $f$ is $\beta$-smooth, then $\beta_k = \beta$ is valid choice since
  \[
  f(y) \leq f(x) + \nabla f(x)^T (y - x) + \frac{\beta}{2} \|x - y\|^2
  \]
  for all $x, y$, select $\gamma_k \in [\epsilon, \frac{2}{\beta} - \epsilon]$
Choose $\beta_k$ and $\gamma_k$ – Backtracking

• Backtracking, choose $\delta > 1$, $\beta_k \in [\eta, \eta^{-1}]$ and loop:
  1. choose $\gamma_k \in [\epsilon, \frac{2}{\beta_k} - \epsilon]$
  2. compute $x_{k+1} = \text{prox}_{\gamma_k g}(x_k - \gamma_k \nabla f(x_k))$
  3. if descent condition (DC) satisfied
     break
   else
     set $\beta_k \leftarrow \delta \beta_k$ and go to 1
   end
• Backtracking will terminate within finite number of backtracks if:
  • $f$ smooth ($\nabla f$ Lipschitz), constant unknown: initialize $\beta_k = \beta_{k-1}$
  • $\nabla f$ locally Lipschitz and sequence bounded: initialize $\beta_k = \beta$
When is problem solved?

• Consider minimize\( f(x) + g(x) \)

• Apply proximal gradient method \( x_{k+1} = \text{prox}_{\gamma_k g}(x_k - \gamma_k \nabla f(x_k)) \)

• Algorithm sequence satisfies

\[
\partial g(x_{k+1}) + \nabla f(x_{k+1}) \ni \gamma_k^{-1}(x_k - x_{k+1}) + \underbrace{\nabla f(x_{k+1}) - \nabla f(x_k)}_{u_k} \to 0
\]

is \( \|u_k\| \) small a good measure of being close to fixed-point?
When is problem solved?

Let $\delta > 0$ and solve equivalent problem $\minimize_x (\delta f(x) + \delta g(x))$:

- Denote algorithm parameter $\gamma_{\delta,k} = \frac{\gamma_k}{\delta}$
- Algorithm satisfies:

$$x_{k+1} = \prox_{\gamma_{\delta,k} \delta g}(x_k - \gamma_{\delta,k} \nabla \delta f(x_k)) = \prox_{\gamma_k g}(x_k - \gamma_k \nabla f(x_k))$$

i.e., the same algorithm as before

- However, $u_{\delta,k}$ in this setting satisfies

$$u_{\delta,k} = \gamma_{\delta,k}^{-1}(x_k - x_{k+1}) + \nabla \delta f(x_{k+1}) - \nabla \delta f(x_k)$$

$$= \delta \gamma_{\delta}^{-1}(x_k - x_{k+1}) + \nabla f(x_{k+1}) - \nabla f(x_k)$$

$$= \delta u_k$$

i.e., same algorithm but different optimality measure

- Optimality measure should be scaling invariant
Stopping condition

• For $\beta$ smooth $f$, use scaled condition $\beta^{-1}u_k$

$$\beta^{-1}u_k := \beta^{-1}(\gamma_k^{-1}(x_k - x_{k+1}) + \nabla f(x_{k+1}) - \nabla f(x_k))$$

which is scale invariant

• Stop algorithm when $\beta^{-1}u_k$ is small enough
  • absolute stopping conditions with small $\epsilon_{\text{abs}} > 0$
    • $\beta^{-1}\|u_k\|_2 \leq \epsilon_{\text{abs}}$
    • $\beta^{-1}(\gamma_k^{-1}\|x_k - x_{k+1}\|_2 + \|\nabla f(x_k) - \nabla f(x_{k+1})\|_2) \leq \epsilon_{\text{abs}}$
  • relative stopping condition with small $\epsilon_{\text{rel}}, \epsilon > 0$
    • $\beta^{-1}\frac{\|u_k\|_{x_k} + \epsilon}{\|x_k\|_2 + \epsilon} \leq \epsilon_{\text{rel}}$
    • $\beta^{-1}\gamma_k^{-1}\frac{\|x_k - x_{k+1}\|_2}{\|x_k\|_2 + \epsilon} + \frac{\|\nabla f(x_k) - \nabla f(x_{k+1})\|_2}{\|\nabla f(x_k)\|_2 + \epsilon} \leq \epsilon_{\text{rel}}$

• Problem considered solved to optimality if, say, $\epsilon_{\text{abs}} \leq 10^{-6}$

• Sometimes want to stop algorithm early, a form of regularization

• Other stopping conditions can be used, should be scaling invariant
Example – SVM

- Classification problem from SVM lecture, SVM with
  - polynomial features of degree 2
  - regularization parameter $\lambda = 0.00001$
Example – Fixed-point residual

- Plots $\beta^{-1}\|u_k\|_2 = \beta^{-1}\|\gamma_k^{-1}(x_k - x_{k+1}) + \nabla f(x_{k+1}) - \nabla f(x_k)\|_2$
- Shows residual up to 20,000 iterations
- Quite many iterations needed to converge
Example – SVM higher degree polynomial

- Classification problem from SVM lecture, SVM with
  - polynomial features of degree 6
  - regularization parameter $\lambda = 0.00001$
Example – Fixed-point residual

- Plots $\beta^{-1}\|u_k\|_2 = \beta^{-1}\|\gamma_k^{-1}(x_k - x_{k+1}) + \nabla f(x_{k+1}) - \nabla f(x_k)\|_2$
- Shows residual up to 200'000 iterations (10x more than before)
- Many iterations needed
Applying proximal gradient to primal problems

Problem \(\min_x f(x) + g(x)\):

- Assumptions:
  - \(f\) smooth
  - \(g\) closed convex and prox friendly\(^1\)
- Algorithm: \(x_{k+1} = \text{prox}_{\gamma_k g}(x_k - \gamma_k \nabla f(x_k))\)

Problem \(\min_x f(Lx) + g(x)\):

- Assumptions:
  - \(f\) smooth (implies \(f \circ L\) smooth)
  - \(g\) closed convex and prox friendly\(^1\)
- Gradient \(\nabla(f \circ L)(x) = L^T \nabla f(Lx)\)
- Algorithm: \(x_{k+1} = \text{prox}_{\gamma_k g}(x_k - \gamma_k L^T \nabla f(Lx_k))\)

\(^1\) Prox friendly: proximal operator cheap to evaluate, e.g., \(g\) separable
Applying proximal gradient to dual problem

Dual problem minimize $f^*(\nu) + g^*(-LT\nu)$:

- Assumptions:
  - $f$ closed convex and prox friendly
  - $g$ strongly convex (which implies $g^* \circ -LT$ smooth)
- Gradient: $\nabla (g^* \circ -LT)(\nu) = -L \nabla g^*(-LT \nu)$
- Prox (Moreau): $\text{prox}_{\gamma_k f^*}(\nu) = \nu - \gamma_k \text{prox}_{\gamma_k^{-1} f}(\gamma_k^{-1} \nu)$
- Algorithm:

$$
\nu_{k+1} = \text{prox}_{\gamma_k f^*}(\nu_k - \gamma_k \nabla (g^* \circ -LT)(\nu_k))
= (I - \gamma_k \text{prox}_{\gamma_k^{-1} f}(\gamma_k^{-1} \circ I))(\nu_k + \gamma_k L \nabla g^*(-LT \nu_k))
$$

- Problem must be convex to have dual!
- Enough to know prox of $f$
Primal recovery

- Fermat’s rule for dual proximal gradient method

\[ 0 \in \partial f^*(\nu_{k+1}) + \nabla (g^* \circ -L^T)(\nu_k) + \gamma_k^{-1}(\nu_{k+1} - \nu_k) \]

\[ = \partial f^*(\nu_{k+1}) - L\nabla (g^*(-L^T \nu_k)) + \gamma_k^{-1}(\nu_{k+1} - \nu_k) \]

- Now, let \( x_k = \nabla g^*(-L^T \nu_k) \), then

\[ 0 \in \begin{cases} \nabla g^*(-L^T \nu_k) - x_k \\ \partial f^*(\nu_{k+1}) - Lx_k + \gamma_k^{-1}(\nu_{k+1} - \nu_k) \end{cases} \]

and \( (x_k, \nu_k) \) satisfies optimality condition when \( \nu_{k+1} - \nu_k \to 0 \)
What problems cannot be solved (efficiently)?

Problem minimize \( f(x) + g(x) \)

- Assumptions: \( f \) and \( g \) convex and nonsmooth
- No term differentiable, another method must be used:
  - Subgradient method
  - Douglas-Rachford splitting
  - Primal-dual methods

Problem minimize \( f(x) + g(Lx) \)

- Assumptions:
  - \( f \) smooth
  - \( g \) nonsmooth convex
  - \( L \) arbitrary structured matrix
- Can apply proximal gradient method, but

\[
\text{prox}_{\gamma_k(g \circ L)}(z) = \arg\min_{x} g(Lx) + \frac{1}{2\gamma} \| x - z \|_2^2
\]

often not “prox friendly”, i.e., it is expensive to evaluate