
Laboratory for Information and Inference Systems (LIONS)
École Polytechnique Fédérale de Lausanne (EPFL)

EE-556 (Fall 2022)
License Information for Mathematics of Data Slides

- This work is released under a Creative Commons License with the following terms:
- **Attribution**
  - The licensor permits others to copy, distribute, display, and perform the work. In return, licensees must give the original authors credit.
- **Non-Commercial**
  - The licensor permits others to copy, distribute, display, and perform the work. In return, licensees may not use the work for commercial purposes – unless they get the licensor’s permission.
- **Share Alike**
  - The licensor permits others to distribute derivative works only under a license identical to the one that governs the licensor’s work.
- **Full Text of the License**
Outline

- Composite minimization
- Proximal gradient methods
- Introduction to Frank-Wolfe method
Recall sparse regression in generalized linear models (GLMs)

**Problem (Sparse regression in GLM)**

Our goal is to estimate \( x^\natural \in \mathbb{R}^p \) given \( \{b_i\}_{i=1}^n \) and \( \{a_i\}_{i=1}^n \), knowing that the likelihood function at \( y_i \) given \( a_i \) and \( x^\natural \) is given by \( L(\langle a_i, x^\natural \rangle, b_i) \), and that \( x^\natural \) is sparse.

**Optimization formulation**

\[
\min_{x \in \mathbb{R}^p} \left\{ -\sum_{i=1}^n \log L(\langle a_i, x^\natural \rangle, b_i) + \rho_n \|x\|_1 \right\}
\]

where \( \rho_n > 0 \) is a parameter which controls the strength of sparsity regularization.

**Theorem (cf. [13] for details)**

Under some technical conditions, there exists \( \{\rho_i\}_{i=1}^\infty \) such that with high probability,

\[
\| x^\star - x^\natural \|_2^2 = \mathcal{O} \left( \frac{s \log p}{n} \right), \quad \text{supp } x^\star = \text{supp } x^\natural.
\]

Recall ML: \( \| x_{ML} - x^\natural \|_2^2 = \mathcal{O} \left( \frac{p}{n} \right) \).
Sparse inverse covariance estimation

Problem (Graphical model selection)

Given a data set $\mathcal{D} := \{x_1, \cdots, x_n\}$, where $x_i$ is a Gaussian random variable. Let $\Sigma$ be the covariance matrix corresponding to the graphical model of the Gaussian Markov random field. Our goal is to learn a sparse precision matrix $X$ (i.e., the inverse covariance matrix $\Sigma^{-1}$) that captures the Markov random field structure.

**Optimization formulation** [16]

$$
\min_{X \succ 0} \left\{ \text{tr}(\Sigma X) - \log \det(X) + \rho_n \|\text{vec}(X)\|_1 \right\},
$$

where $X \succ 0$ means that $X$ is symmetric and positive definite and $\rho_n > 0$ is a regularization parameter and $\text{vec}$ is the vectorization operator. Let $X^\star$ be the minimizer of (1), under some technical conditions, there exists a $\rho_n$ such that $\|X^\star - \Sigma^{-1}\|_2^2 = O(\min\{d^2 \log p, (s + p) \log p\}/n)$ where $d$ is the maximum graph degree.
Composite convex minimization

Problem (Composite convex minimization)

\[
F^* := \min_{x \in \mathbb{R}^p} \{ F(x) := f(x) + g(x) \}
\] (2)

- \( f \) and \( g \) are both proper, closed, and convex.
- \( \text{dom}(F) := \text{dom}(f) \cap \text{dom}(g) \neq \emptyset \) and \(-\infty < F^* < +\infty\).
- The solution set \( S^* := \{ x^* \in \text{dom}(F) : F(x^*) = F^* \} \) is nonempty.

Remarks:
- Without loss of generality, \( f \) is smooth and \( g \) is non-smooth in the sequel.
- By Moreau-Rockafellar Theorem, we have \( \partial F = \partial (f + g) = \partial f + \partial g = \nabla f + \partial g \).
- Subgradient method attains a \( O\left(1/\sqrt{T}\right) \) rate.
- Without \( g \), accelerated gradient method attains a \( O\left(1/T^2\right) \) rate.
Composite convex minimization

Problem (Composite convex minimization)

\[
F^* := \min_{x \in \mathbb{R}^p} \{ F(x) := f(x) + g(x) \} \tag{2}
\]

- \( f \) and \( g \) are both proper, closed, and convex.
- \( \text{dom}(F) := \text{dom}(f) \cap \text{dom}(g) \neq \emptyset \) and \(-\infty < F^* < +\infty\).
- The solution set \( S^* := \{ x^* \in \text{dom}(F) : F(x^*) = F^* \} \) is nonempty.

Remarks:
- Without loss of generality, \( f \) is smooth and \( g \) is non-smooth in the sequel.
- By Moreau-Rockafellar Theorem, we have \( \partial F = \partial (f + g) = \partial f + \partial g = \nabla f + \partial g \).
- Subgradient method attains a \( \mathcal{O}\left(1/\sqrt{T}\right) \) rate.
- Without \( g \), accelerated gradient method attains a \( \mathcal{O}\left(1/T^2\right) \) rate.

Can we design algorithms that achieve a faster convergence rate for composite convex minimization?
Designing algorithms for finding a solution $x^*$

**Quadratic majorizer for $f$**

When $f$ has $L$-Lipschitz continuous gradient, we have, $\forall x, y \in \mathbb{R}^p$

$$f(x) \leq f(y) + \nabla f(y)^T(x - y) + \frac{L}{2} \|x - y\|^2$$
Designing algorithms for finding a solution $x^*$

**Quadratic majorizer for $f$**

When $f$ has $L$-Lipschitz continuous gradient, we have, $\forall x, y \in \mathbb{R}^p$

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

**Quadratic majorizer for $f + g$**

When $f$ has $L$-Lipschitz continuous gradient, we have, $\forall x, y \in \mathbb{R}^p$

$$f(x) + g(x) \leq f(y) + \nabla f(y)^T(x - y) + \frac{L}{2} \|x - y\|_2^2 + g(x) := P_L(x, y)$$
Designing algorithms for finding a solution $x^*$

**Quadratic majorizer for $f$**

When $f$ has $L$-Lipschitz continuous gradient, we have, $\forall x, y \in \mathbb{R}^p$

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

**Quadratic majorizer for $f + g$**

When $f$ has $L$-Lipschitz continuous gradient, we have, $\forall x, y \in \mathbb{R}^p$

$$f(x) + g(x) \leq f(y) + \nabla f(y)^T(x - y) + \frac{L}{2} \|x - y\|_2^2 + g(x) := P_L(x, y)$$

**Majorization-minimization for $f + g$**

$$x^{k+1} = \arg \min_{x \in \mathbb{R}^p} P_L(x, x^k) = \arg \min_{x \in \mathbb{R}^p} \left\{ g(x) + \frac{L}{2} \|x - \left(x^k - \frac{1}{L} \nabla f(x^k)\right)\|^2 \right\}$$
Geometric illustration

\[ P_L(x, x^k) := f(x^k) + \nabla f(x^k)^T (x - x^k) + \frac{L}{2} \|x - x^k\|_2^2 + g(x) \]

\[ F(x) = f(x) + g(x) \]
A short detour: Proximal-point operators

Definition (Proximal operator [18])
Let $g \in \mathcal{F}(\mathbb{R}^p)$, $x \in \mathbb{R}^p$ and $\lambda \geq 0$. The proximal operator (or prox-operator) of $g$ is defined as:

$$\text{prox}_\lambda g(x) \equiv \arg \min_{y \in \mathbb{R}^p} \left\{ g(y) + \frac{1}{2\lambda} \|y - x\|^2 \right\}. \tag{3}$$
A short detour: Proximal-point operators

**Definition (Proximal operator [18])**

Let $g \in \mathcal{F}(\mathbb{R}^p)$, $x \in \mathbb{R}^p$ and $\lambda \geq 0$. The proximal operator (or prox-operator) of $g$ is defined as:

$$
\text{prox}_{\lambda g}(x) \equiv \arg \min_{y \in \mathbb{R}^p} \left\{ g(y) + \frac{1}{2\lambda} \|y - x\|^2 \right\}.
$$

(3)

**Remarks:**

- The **proximal operator** of $\frac{1}{L} g$ evaluated at $(x^k - \frac{1}{L} \nabla f(x^k))$ is given by

$$
\text{prox}_{\frac{1}{L} g} \left(x^k - \frac{1}{L} \nabla f(x^k)\right) = \arg \min_{x \in \mathbb{R}^p} \left\{ g(x) + \frac{L}{2} \|x - \left(x^k - \frac{1}{L} \nabla f(x^k)\right)\|^2 \right\}.
$$

- This prox-operator minimizes the majorizing bound:

$$
f(x) + g(x) \leq f(x^k) + \nabla f(x^k)^T (x - x^k) + \frac{L}{2} \|x - x^k\|_2^2 + g(x)
$$

- Rule of thumb: Replace gradient steps with proximal gradient steps!
Tractable prox-operators

Processing non-smooth terms in (16)

▶ We handle the nonsmooth term $g$ in (16) using its proximal operator.
▶ However, computing proximal operator $\text{prox}_g$ of a general convex function $g$

$$\text{prox}_g(x) \equiv \arg \min_{y \in \mathbb{R}^p} \left\{ g(y) + \frac{1}{2}\|y - x\|_2^2 \right\}.$$ 

can be computationally demanding.

Definition (Tractable proximity)

▶ Given $g \in \mathcal{F}(\mathbb{R}^p)$. We say that $g$ is proximally tractable if $\text{prox}_g$ defined by (3) can be computed efficiently.
▶ "efficiently" = \{closed form solution, low-cost computation, polynomial time\}. 
Tractable prox-operators

Example

- For separable functions, the prox-operator can be efficient. When \( g(x) := \|x\|_1 = \sum_{i=1}^{n} |x_i| \), we have
  \[
  \text{prox}_\lambda g(x) = \text{sign}(x) \otimes \max\{|x| - \lambda, 0\}.
  \]

- Sometimes, we can compute the prox-operator via basic algebra. When \( g(x) := \frac{1}{2} \|Ax - b\|_2^2 \), we have
  \[
  \text{prox}_\lambda g(x) = \left( I + \lambda A^T A \right)^{-1} (x + \lambda Ab).
  \]

- For the indicator functions of simple sets, e.g., \( g(x) := \delta_{\mathcal{X}}(x) \), the prox-operator is the projection operator
  \[
  \text{prox}_\lambda g(x) := \pi_{\mathcal{X}}(x),
  \]
  where \( \pi_{\mathcal{X}}(x) \) denotes the projection of \( x \) onto \( \mathcal{X} \). For instance, when \( \mathcal{X} = \{x : \|x\|_1 \leq \lambda\} \), the projection can be obtained efficiently.
Computational efficiency - Example

Proximal operator of quadratic function

The proximal operator of a quadratic function \( g(x) := \frac{1}{2} \|Ax - b\|_2^2 \) is defined as

\[
\text{prox}_\lambda g(x) := \arg \min_{y \in \mathbb{R}^p} \left\{ \frac{1}{2} \|Ay - b\|_2^2 + \frac{1}{2\lambda} \|y - x\|_2^2 \right\}.
\] (4)

How do we compute \( \text{prox}_\lambda g(x) \)?

The derivation:

\( \circ \) The optimality condition implies that the solution of (4) should satisfy the following:

\[ A^T(Ay - b) + \lambda^{-1}(y - x) = 0. \]

\( \circ \) Setting \( y = \text{prox}_\lambda g(x) \), we obtain

\[ \text{prox}_\lambda g(x) = \left( I + \lambda A^T A \right)^{-1} (x + \lambda Ab). \]

Remarks:

\( \circ \) The Woodbury matrix identity can be useful: \( (I + \lambda A^T A)^{-1} = I - A^T(\lambda^{-1}I + AA^T)^{-1} A. \)

\( \circ \) When \( A^T A \) is efficiently diagonalizable, i.e., \( A^T A := U \Lambda U^T \), such that

- \( U \) is a unitary matrix, i.e., \( UU^T = U^TU = I \), and \( \Lambda \) is a diagonal matrix.

- \( \text{prox}_\lambda g(x) = U (I + \lambda \Lambda)^{-1} U^T (x + \lambda Ab) \).
A non-exhaustive list of proximal tractability functions

<table>
<thead>
<tr>
<th>Name</th>
<th>Function</th>
<th>Proximal operator</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>ℓ₁-norm</td>
<td>( f(x) := |x|_1 )</td>
<td>( \text{prox}_\lambda f(x) = \text{sign}(x) \otimes [|x|<em>1 - \lambda]</em>+ )</td>
<td>( O(p) )</td>
</tr>
<tr>
<td>ℓ₂-norm</td>
<td>( f(x) := |x|_2 )</td>
<td>( \text{prox}_\lambda f(x) = [1 - \lambda/|x|<em>2]</em>+ x )</td>
<td>( O(p) )</td>
</tr>
<tr>
<td>Support function</td>
<td>( f(x) := \max_{y \in C} x^T y )</td>
<td>( \text{prox}_\lambda f(x) = x - \lambda \pi_C(x) )</td>
<td>( O(p) )</td>
</tr>
<tr>
<td>Box indicator</td>
<td>( f(x) := \delta_{[a,b]}(x) )</td>
<td>( \text{prox}<em>\lambda f(x) = \pi</em>{[a,b]}(x) )</td>
<td>( O(p) )</td>
</tr>
<tr>
<td>Positive semidefinite cone indicator</td>
<td>( f(X) := \delta_{S^+}(X) )</td>
<td>( \text{prox}<em>\lambda f(X) = U[S^+]</em>+ U^T, \text{ where } X = U \Sigma U^T )</td>
<td>( O(p^3) )</td>
</tr>
<tr>
<td>Hyperplane indicator</td>
<td>( f(x) := \delta_{\mathcal{X}}(x), \mathcal{X} := {x : a^T x = b} )</td>
<td>( \text{prox}_\lambda f(x) = x + (b - a^T x)/|a|_2 ) ( a )</td>
<td>( O(p) )</td>
</tr>
<tr>
<td>Simplex indicator</td>
<td>( f(x) := \delta_{\mathcal{X}}(x), \mathcal{X} := {x : x \geq 0, 1^T x = 1} )</td>
<td>( \text{prox}_\lambda f(x) = (x - \nu 1) ) for some ( \nu \in \mathbb{R} ), which can be efficiently calculated</td>
<td>( \tilde{O}(p) )</td>
</tr>
<tr>
<td>Convex quadratic</td>
<td>( f(x) := (1/2)x^T Q x + q^T x )</td>
<td>( \text{prox}_\lambda f(x) = (\lambda I + Q)^{-1} x )</td>
<td>( O(p \log p) \rightarrow O(p^3) )</td>
</tr>
<tr>
<td>Square ℓ₂-norm</td>
<td>( f(x) := (1/2)|x|_2^2 )</td>
<td>( \text{prox}_\lambda f(x) = (1/(1 + \lambda)) x )</td>
<td>( O(p) )</td>
</tr>
<tr>
<td>log-function</td>
<td>( f(x) := -\log(x) )</td>
<td>( \text{prox}_\lambda f(x) = ((x^2 + 4\lambda)^{1/2} + x)/2 )</td>
<td>( O(1) )</td>
</tr>
<tr>
<td>log det-function</td>
<td>( f(x) := -\log \det(X) )</td>
<td>( \text{prox}_\lambda f(X) ) is the log-function prox applied to the individual eigenvalues of ( X )</td>
<td>( O(p^3) )</td>
</tr>
</tbody>
</table>

Here: \([x]_+ := \max\{0, x\}\) and \(\delta_{\mathcal{X}}\) is the indicator function of the convex set \(\mathcal{X}\), \(\text{sign}\) is the sign function, \(S^+_p\) is the cone of symmetric positive semidefinite matrices.

For more functions, see [5, 15].
Solution methods

Composite convex minimization

\[ F^* := \min_{x \in \mathbb{R}^p} \left\{ F(x) := f(x) + g(x) \right\}. \] (5)

Choice of numerical solution methods

- **Solve (5)** = Find \( x^k \in \mathbb{R}^p \) such that

\[ F(x^k) - F^* \leq \varepsilon \]

for a given tolerance \( \varepsilon > 0 \).

- **Oracles**: We can use one of the following configurations (oracles):
  1. \( \partial f(\cdot) \) and \( \partial g(\cdot) \) at any point \( x \in \mathbb{R}^p \).
  2. \( \nabla f(\cdot) \) and \( \text{prox}_{\lambda g}(\cdot) \) at any point \( x \in \mathbb{R}^p \).
  3. \( \text{prox}_{\lambda f} \) and \( \text{prox}_{\lambda g}(\cdot) \) at any point \( x \in \mathbb{R}^p \).
  4. \( \nabla f(\cdot) \), inverse of \( \nabla^2 f(\cdot) \) and \( \text{prox}_{\lambda g}(\cdot) \) at any point \( x \in \mathbb{R}^p \).

Using different oracle leads to different types of algorithms
## Proximal-gradient algorithm

<table>
<thead>
<tr>
<th>Basic proximal-gradient scheme (ISTA)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1.</strong> Choose $x^0 \in \text{dom}(F)$ arbitrarily as a starting point.</td>
</tr>
<tr>
<td><strong>2.</strong> For $k = 0, 1, \cdots$, generate a sequence ${x^k}_{k \geq 0}$ as:</td>
</tr>
<tr>
<td>$x^{k+1} := \text{prox}_{\alpha g} \left( x^k - \alpha \nabla f(x^k) \right)$,</td>
</tr>
<tr>
<td>where $\alpha := \frac{1}{L}$.</td>
</tr>
</tbody>
</table>
Proximal-gradient algorithm

<table>
<thead>
<tr>
<th>Basic proximal-gradient scheme (ISTA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Choose $x^0 \in \text{dom}(F)$ arbitrarily as a starting point.</td>
</tr>
<tr>
<td>2. For $k = 0, 1, \ldots$, generate a sequence ${x^k}_{k \geq 0}$ as:</td>
</tr>
<tr>
<td>$x^{k+1} := \text{prox}_{\alpha g}(x^k - \alpha \nabla f(x^k))$,</td>
</tr>
<tr>
<td>where $\alpha := \frac{1}{L}$.</td>
</tr>
</tbody>
</table>

Theorem (Convergence of ISTA [2])

Let $\{x^k\}$ be generated by ISTA. Then:

$$F(x^k) - F^* \leq \frac{L_f \|x^0 - x^*\|_2^2}{2(k + 1)}$$

The worst-case complexity to reach $F(x^k) - F^* \leq \varepsilon$ of (ISTA) is $O\left(\frac{L_f R_0^2}{\varepsilon}\right)$, where $R_0 := \max_{x^* \in S^*} \|x^0 - x^*\|_2$.

- **Oracles**: $\text{prox}_{\alpha g}(\cdot)$ and $\nabla f(\cdot)$.

- Compared to the subgradient gradient method, the rate improves at the cost of prox-computation.
Fast proximal-gradient algorithm

**Fast proximal-gradient scheme (FISTA)**

1. Choose $x^0 \in \text{dom}(F)$ arbitrarily as a starting point.
2. Set $y^0 := x^0$ and $t_0 := 1$, $\alpha := L^{-1}$.
3. For $k = 0, 1, \ldots$, generate two sequences $\{x^k\}_{k \geq 0}$ and $\{y^k\}_{k \geq 0}$ as:

\[
\begin{cases}
  x^{k+1} := \text{prox}_{\alpha g} \left( y^k - \alpha \nabla f(y^k) \right), \\
  t_{k+1} := (1 + \sqrt{4\frac{t_k^2}{t_{k+1}^2} + 1})/2, \\
  y^{k+1} := x^{k+1} + \frac{t_{k-1}}{t_{k+1}}(x^{k+1} - x^k).
\end{cases}
\]
Fast proximal-gradient algorithm

Fast proximal-gradient scheme (FISTA)

1. Choose $x^0 \in \text{dom}(F)$ arbitrarily as a starting point.
2. Set $y^0 := x^0$ and $t_0 := 1$, $\alpha := L^{-1}$.
3. For $k = 0, 1, \ldots$, generate two sequences $\{x^k\}_{k \geq 0}$ and $\{y^k\}_{k \geq 0}$ as:

\[
\begin{aligned}
x^{k+1} &= \text{prox}_{\alpha g} \left( y^k - \alpha \nabla f(y^k) \right), \\
t^{k+1} &= \left( 1 + \sqrt{4t^2_k + 1} \right) / 2, \\
y^{k+1} &= x^{k+1} + \frac{t_k - 1}{t_{k+1}} (x^{k+1} - x^k).
\end{aligned}
\]

Theorem (Convergence of FISTA [2])

Let $\{x^k\}$ be generated by FISTA. Then:

\[
F(x^k) - F^* \leq \frac{2L_f \|x^0 - x^*\|_2^2}{(k + 1)^2}
\]

The worst-case complexity to reach $F(x^k) - F^* \leq \varepsilon$ of (FISTA) is $O \left( R_0 \sqrt{\frac{L_f}{\varepsilon}} \right)$, $R_0 := \max_{x^* \in \mathcal{S}^*} \|x^0 - x^*\|_2$. 

Mathematics of Data | Prof. Volkan Cevher, volkan.cevher@epfl.ch

Slide 16 / 48
## Fast proximal-gradient algorithm

### Fast proximal-gradient scheme (FISTA)

1. Choose \( x^0 \in \text{dom} (F) \) arbitrarily as a starting point.
2. Set \( y^0 := x^0 \) and \( t_0 := 1, \alpha := L^{-1}. \)
3. For \( k = 0, 1, \ldots, \) generate two sequences \( \{x^k\}_{k \geq 0} \) and \( \{y^k\}_{k \geq 0} \) as:

\[
\begin{align*}
x^{k+1} &:= \text{prox}_{\alpha g} \left( y^k - \alpha \nabla f (y^k) \right), \\
t_{k+1} &:= \frac{1 + \sqrt{4t_k^2 + 1}}{2}, \\
y^{k+1} &:= x^{k+1} + \frac{t_k - 1}{t_{k+1}} (x^{k+1} - x^k).
\end{align*}
\]

**Remark:** From \( \mathcal{O} \left( \frac{L_f R_0^2}{\epsilon} \right) \) to \( \mathcal{O} \left( R_0 \sqrt{\frac{L_f}{\epsilon}} \right) \) iterations at **almost no additional cost!**

### Complexity per iteration

- **One** gradient \( \nabla f (y^k) \) and **one** prox-operator of \( g; \)
- **8** arithmetic operations for \( t_{k+1} \) and \( \gamma_{k+1}; \)
- **2** more vector additions, and **one** scalar-vector multiplication.

The **cost per iteration** is **almost the same** as in **gradient scheme** if proximal operator of \( g \) is efficient.
Example 1: $\ell_1$-regularized least squares

Problem ($\ell_1$-regularized least squares)

Given $A \in \mathbb{R}^{n \times p}$ and $b \in \mathbb{R}^n$, solve:

$$F^\star := \min_{x \in \mathbb{R}^p} \left\{ F(x) := \frac{1}{2} \|Ax - b\|_2^2 + \lambda \|x\|_1 \right\}, \quad (6)$$

where $\lambda > 0$ is a regularization parameter.

Complexity per iterations

- Evaluating $\nabla f(x^k) = A^T(Ax^k - b)$ requires one $Ax$ and one $A^Ty$.
- One soft-thresholding operator $\text{prox}_{\lambda g}(x) = \text{sign}(x) \otimes \max\{|x| - \lambda, 0\}$.
- **Optional**: Evaluating $L = \|A^T A\|$ (spectral norm) - via **power iterations**

Synthetic data generation

- $A := \text{randn}(n, p)$ - standard Gaussian $\mathcal{N}(0, I)$.
- $x^\star$ is a $k$-sparse vector generated randomly.
- $b := Ax^\star + \mathcal{N}(0, 10^{-3})$. 
Example 1: Theoretical bounds vs practical performance

Theoretical bounds

We have the following guarantees for $\text{FISTA} := \frac{2L_f R_0^2}{(k+2)^2}$ and for $\text{ISTA} := \frac{L_f R_0^2}{2(k+2)}$. 

![Graph showing theoretical bounds vs practical performance](image)

Remarks:

- $\ell_1$-regularized least squares formulation has restricted strong convexity.
- The proximal-gradient method can automatically exploit this structure.
Example 1: Theoretical bounds vs practical performance

Theoretical bounds

We have the following guarantees for \( \text{FISTA} := \frac{2L_f R_0^2}{(k+2)^2} \) and for \( \text{ISTA} := \frac{L_f R_0^2}{2(k+2)}. \)
Example 1: Theoretical bounds vs practical performance

Theoretical bounds

We have the following guarantees for \( \text{FISTA} := \frac{2L_f R_0^2}{(k+2)^2} \) and for \( \text{ISTA} := \frac{L_f R_0^2}{2(k+2)} \).

Remarks:

- \( \ell_1 \)-regularized least squares formulation has **restricted strong convexity**.
- The proximal-gradient method can automatically exploit this structure.
Example 2: Sparse logistic regression

Problem (Sparse logistic regression)

Given $A \in \mathbb{R}^{n \times p}$ and $b \in \{-1, +1\}^n$, solve:

$$F^* := \min_{x, \beta} \left\{ F(x) := \frac{1}{n} \sum_{j=1}^{n} \log \left( 1 + \exp \left( -b_j (a_j^T x + \beta) \right) \right) + \rho \|x\|_1 \right\}.$$ 

Real data

- Real data: w8a with $n = 49'749$ data points, $p = 300$ features

Parameters

- $\rho = 10^{-4}$.
- Number of iterations 5000, tolerance $10^{-7}$.
- Ground truth: Solve problem up to $10^{-9}$ accuracy by TFOCS to get a high accuracy approximation of $x^*$ and $F^*$. 
Example 2: Sparse logistic regression - numerical results

<table>
<thead>
<tr>
<th></th>
<th>ISTA</th>
<th>LS-ISTA</th>
<th>FISTA</th>
<th>FISTA-R</th>
<th>LS-FISTA</th>
<th>LS-FISTA-R</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of iterations</td>
<td>5000</td>
<td>5000</td>
<td>4046</td>
<td>2423</td>
<td>447</td>
<td>317</td>
</tr>
<tr>
<td>CPU time (s)</td>
<td>26.975</td>
<td>61.506</td>
<td>21.859</td>
<td>18.444</td>
<td>10.683</td>
<td>6.228</td>
</tr>
<tr>
<td>Solution error ($\times 10^{-7}$)</td>
<td>29370</td>
<td>2.774</td>
<td>1.000</td>
<td>0.998</td>
<td>0.961</td>
<td>0.985</td>
</tr>
</tbody>
</table>
When $f$ is strongly convex: Algorithms

<table>
<thead>
<tr>
<th>Proximal-gradient scheme (ISTA$_\mu$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Given $x^0 \in \mathbb{R}^p$ as a starting point.</td>
</tr>
<tr>
<td>2. For $k = 0, 1, \cdots$, generate a sequence ${x^k}_{k \geq 0}$ as:</td>
</tr>
<tr>
<td>$x^{k+1} := \text{prox}_{\alpha_k} g \left( x^k - \alpha_k \nabla f(x^k) \right)$,</td>
</tr>
<tr>
<td>where $\alpha_k := 2/(L_f + \mu)$ is the optimal step-size.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fast proximal-gradient scheme (FISTA$_\mu$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Given $x^0 \in \mathbb{R}^p$ as a starting point. Set $y^0 := x^0$.</td>
</tr>
<tr>
<td>2. For $k = 0, 1, \cdots$, generate sequences ${x^k}<em>{k \geq 0}$ and ${y^k}</em>{k \geq 0}$ as:</td>
</tr>
</tbody>
</table>
| $\begin{cases} 
  x^{k+1} := \text{prox}_{\alpha_k} g \left( y^k - \alpha_k \nabla f(y^k) \right), \\
  y^{k+1} := x^{k+1} + \left( \frac{\sqrt{c_f} - 1}{\sqrt{c_f} + 1} \right) (x^{k+1} - x^k), 
\end{cases}$ |
| where $c_f := L_f / \mu$ and $\alpha_k := L_f^{-1}$ is the optimal step-size. |
When $f$ is strongly convex: Convergence

**Assumption**

$f$ is strongly convex with parameter $\mu > 0$, i.e., $f \in F_{L,\mu}^{1,1}(\mathbb{R}^p)$.

**Condition number:** $c_f := \frac{L_f}{\mu} \geq 0$.

**Theorem (ISTA$_\mu$ [14])**

$$F(x^k) - F^* \leq \frac{L_f}{2} \left( \frac{c_f - 1}{c_f + 1} \right)^{2k} \|x^0 - x^*\|^2_2.$$  

**Convergence rate:** Linear with contraction factor: $\omega := \left( \frac{c_f - 1}{c_f + 1} \right)^2 = \left( \frac{L_f - \mu}{L_f + \mu} \right)^2$.

**Theorem (FISTA$_\mu$ [14])**

$$F(x^k) - F^* \leq \frac{L_f + \mu}{2} \left( 1 - \sqrt{\frac{\mu}{L_f}} \right)^k \|x^0 - x^*\|^2_2.$$  

**Convergence rate:** Linear with contraction factor: $\omega_f = \frac{\sqrt{L_f} - \sqrt{\mu}}{\sqrt{L_f}} < \omega$. 
### Summary of the worst-case complexities

#### Comparison

<table>
<thead>
<tr>
<th>Complexity</th>
<th>Proximal-gradient scheme</th>
<th>Fast proximal-gradient scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complexity $[\mu = 0]$</td>
<td>$O\left(R_0^2\left(L_f/\varepsilon\right)\right)$</td>
<td>$O\left(R_0 \sqrt{L_f/\varepsilon}\right)$</td>
</tr>
<tr>
<td>Per iteration</td>
<td>1-gradient, 1-prox, 1-sv, 1-v+</td>
<td>1-gradient, 1-prox, 2-sv, 3-v+</td>
</tr>
<tr>
<td>Complexity $[\mu &gt; 0]$</td>
<td>$O\left(\kappa \log(\varepsilon^{-1})\right)$</td>
<td>$O\left(\sqrt{\kappa} \log(\varepsilon^{-1})\right)$</td>
</tr>
<tr>
<td>Per iteration</td>
<td>1-gradient, 1-prox, 1-sv, 1-v+</td>
<td>1-gradient, 1-prox, 1-sv, 2-v+</td>
</tr>
</tbody>
</table>

Here: $sv =$ scalar-vector multiplication, $v+ =$ vector addition.

$R_0 := \max_{x^* \in S^*} \|x^0 - x^*\|$ and $\kappa = L_f/\mu_f$ is the condition number.
Summary of the worst-case complexities

Comparison

<table>
<thead>
<tr>
<th>Complexity</th>
<th>Proximal-gradient scheme</th>
<th>Fast proximal-gradient scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complexity $\mu = 0$</td>
<td>$O \left( R_0^2 \left( \frac{L_f}{\varepsilon} \right) \right)$</td>
<td>$O \left( R_0 \sqrt{L_f/\varepsilon} \right)$</td>
</tr>
<tr>
<td>Per iteration</td>
<td>1-gradient, 1-prox, 1-$sv$, 1-$v+$</td>
<td>1-gradient, 1-prox, 2-$sv$, 3-$v+$</td>
</tr>
<tr>
<td>Complexity $\mu &gt; 0$</td>
<td>$O \left( \kappa \log \left( \frac{1}{\varepsilon - 1} \right) \right)$</td>
<td>$O \left( \sqrt{\kappa} \log \left( \frac{1}{\varepsilon - 1} \right) \right)$</td>
</tr>
<tr>
<td>Per iteration</td>
<td>1-gradient, 1-prox, 1-$sv$, 1-$v+$</td>
<td>1-gradient, 1-prox, 1-$sv$, 2-$v+$</td>
</tr>
</tbody>
</table>

Here: $sv = \text{scalar-vector multiplication}$, $v+ = \text{vector addition}$.

$R_0 := \max_{x^* \in S^*} \| x^0 - x^* \|$ and $\kappa = \frac{L_f}{\mu_f}$ is the condition number.

Need alternatives when

- computing $\nabla f(x)$ is much costlier than computing $\text{prox}_g$
## Summary of the worst-case complexities

### Comparison

<table>
<thead>
<tr>
<th>Complexity</th>
<th>Proximal-gradient scheme</th>
<th>Fast proximal-gradient scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>Complexity $[\mu = 0]$</td>
<td>$O \left( R_0^2 \left( \frac{L_f}{\varepsilon} \right) \right)$</td>
<td>$O \left( R_0 \sqrt{L_f / \varepsilon} \right)$</td>
</tr>
<tr>
<td>Per iteration</td>
<td>1-gradient, 1-prox, 1-$sv$, 1-$v+$</td>
<td>1-gradient, 1-prox, 2-$sv$, 3-$v+$</td>
</tr>
<tr>
<td>Complexity $[\mu &gt; 0]$</td>
<td>$O \left( \kappa \log(\varepsilon^{-1}) \right)$</td>
<td>$O \left( \sqrt{\kappa \log(\varepsilon^{-1})} \right)$</td>
</tr>
<tr>
<td>Per iteration</td>
<td>1-gradient, 1-prox, 1-$sv$, 1-$v+$</td>
<td>1-gradient, 1-prox, 1-$sv$, 2-$v+$</td>
</tr>
</tbody>
</table>

Here: $sv = \text{scalar-vector multiplication}, v+ = \text{vector addition}$.

$R_0 := \max_{x^* \in S^*} \| x^0 - x^* \|$ and $\kappa = \frac{L_f}{\mu_f}$ is the condition number.

### Need alternatives when

- Computing $\nabla f(x)$ is much costlier than computing $\text{prox}_g$

### Software

**TFOCS** is a good software package to learn about first order methods.

http://cvxr.com/tfocs/
## Composite minimization: Non-convex case

### Problem (Unconstrained composite minimization)

\[ F^* := \min_{x \in \mathbb{R}^p} \{ F(x) := f(x) + g(x) \} \quad \text{(CM)} \]

- \( g : \mathbb{R}^p \to \mathbb{R} \cup \{\infty\} \) is proper, closed, convex, and (possibly) nonsmooth.
- \( f : \mathbb{R}^p \to \mathbb{R} \) is proper and closed, \( \text{dom}(f) \) is convex, and \( f \) is \( L_f \)-smooth.
- \( \text{dom}(F) := \text{dom}(f) \cap \text{dom}(g) \neq \emptyset \) and \( -\infty < F^* < +\infty \).
- The solution set \( S^* := \{ x^* \in \text{dom}(F) : F(x^*) = F^* \} \) is nonempty.
A different quantification of convergence: Gradient mapping

Definition (Gradient mapping)

Let \( \text{prox}_g \) denote the proximal operator of \( g \) and \( \lambda > 0 \) some real constant. Then, the gradient mapping operator is defined as

\[
G_\lambda(x) := \frac{1}{\lambda} \left( x - \text{prox}_\lambda g(x - \lambda \nabla f(x)) \right).
\]

Properties [1]

- \( \|G_\lambda(x)\| = 0 \iff x \) is a stationary point.
- Lipschitz continuity: \( \|G_{\lambda_L}(x) - G_{\lambda_L}(y)\| \leq (2L + L_f)\|x - y\| \)

Why do we care about gradient mapping?

- It is the generalization of the gradient of \( f \), \( \nabla f(x) \)
- Recall prox-gradient update: \( x^{t+1} = \text{prox}_\lambda g(x^t - \lambda \nabla f(x^t)) \), which is equivalent to \( x^{t+1} = x^t - \lambda G_\lambda(x^t) \).
- In fact, when \( \text{prox}_g = \mathbb{I} \), then, \( G_\lambda(x) = \frac{1}{\lambda} (x - (x - \lambda \nabla f(x))) = \nabla f(x) \).
Sufficient Decrease property for proximal-gradient

Assumption

- $f$ is $L_f$-smooth.
- $g$ is proper, closed, convex, and (possibly) nonsmooth. $g$ is proximally tractable.

$$x^{k+1} := \text{prox}_{\frac{1}{L}g} \left( x^k - \frac{1}{L} \nabla f(x^k) \right)$$

Lemma (Sufficient decrease [1])

For any $x \in \text{int}(\text{dom}(f))$ and $L \in (\frac{L_f}{2}, \infty)$, it holds that

$$F(x^{k+1}) \leq F(x^k) - \frac{L_f}{2L^2} \norm{\frac{1}{L} G_{\frac{1}{L}}(x^k)}_2^2,$$

(7)

Corollary

$$F(x^{k+1}) \leq F(x^k) - \frac{1}{2L_f} \norm{\frac{1}{L_f} G_{\frac{1}{L_f}}(x^k)}_2^2,$$

for $L = L_f$
Non-convex case: Convergence

<table>
<thead>
<tr>
<th>Basic proximal-gradient scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Choose $x^0 \in \text{dom } (F)$ arbitrarily as a starting point.</td>
</tr>
<tr>
<td>2. For $k = 0, 1, \cdots$, generate a sequence ${x^k}_{k \geq 0}$ as:</td>
</tr>
<tr>
<td>$x^{k+1} := \text{prox}_{\alpha g} \left( x^k - \alpha \nabla f(x^k) \right),$</td>
</tr>
<tr>
<td>where $\alpha := \left( 0, \frac{2}{L_f} \right).$</td>
</tr>
</tbody>
</table>

Theorem (Convergence of proximal-gradient method: Non-convex [1])
Let $\{x^k\}$ be generated by proximal-gradient scheme above. Then, we have

$$\min_{i=0, 1, \cdots, k} \| G_\alpha(x^i) \|_2^2 \leq \frac{F(x^0) - F(x^*)}{M(k + 1)},$$
where $M := \alpha^2 \left( \frac{1}{\alpha} - \frac{L_f}{2} \right)$

- When $\alpha = \frac{1}{L_f}$, $M = \frac{1}{2L_f}$.
- The worst-case complexity to reach $\min_{i=0, 1, \cdots, k} \| G_\alpha(x^i) \|_2^2 \leq \varepsilon$ is $O \left( \frac{1}{\varepsilon} \right)$. 
**Stochastic convex composite minimization**

**Problem (Mathematical formulation)**

Consider the following composite convex minimization problem:

\[
F^* = \min_{x \in \mathbb{R}^p} \left\{ F(x) := \mathbb{E}_\theta [F(x, \theta)] := \mathbb{E}_\theta [f(x, \theta) + g(x, \theta)] \right\}
\]

- $\theta$ is a random vector whose probability distribution is supported on set $\Theta$.
- The solution set $S^* := \{x^* \in \text{dom}(F) : F(x^*) = F^*\}$ is nonempty.
- Oracles: (sub)gradient of $f(\cdot, \theta)$, $\nabla f(x, \theta)$, and stochastic prox operator of $g(\cdot, \theta)$, $\text{prox}_{g(\cdot, \theta)}(x)$.

**Remark**

- In this setting, we replace $\nabla f(\cdot)$ with its stochastic estimates.
- It is possible to replace $\text{prox}_{g(\cdot)}$ with its stochastic estimate (advanced material).
Stochastic proximal gradient method

<table>
<thead>
<tr>
<th>Stochastic proximal gradient method (SPG)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1.</strong> Choose $x^0 \in \mathbb{R}^p$ and $(\gamma_k)_{k \in \mathbb{N}} \in ]0, +\infty[^{\mathbb{N}}$.</td>
</tr>
<tr>
<td><strong>2.</strong> For $k = 0, 1, \ldots$ perform:</td>
</tr>
<tr>
<td>$x^{k+1} = \text{prox}_{\gamma_k g(\cdot,\theta)}(x^k - \gamma_k G(x^k, \theta^k))$.</td>
</tr>
</tbody>
</table>

**Definitions:**

- $\text{prox}_{\lambda g(\cdot,\theta)} := \arg\min_{y \in \mathbb{R}^p} \left\{ g(y, \theta) + \frac{1}{2\lambda} \| y - x \|^2 \right\}$
- $\{\theta_k\}_{k=0,1,\ldots}$: sequence of independent random variables.
- $G(x^k, \theta^k) \in \partial f(x^k, \theta^k)$: an unbiased estimate of the deterministic (sub)gradient:
  \[ \mathbb{E}[G(x^k, \theta^k)] \in \partial f(x^k). \]
Stochastic proximal gradient method

<table>
<thead>
<tr>
<th>Stochastic proximal gradient method (SPG)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Choose $x^0 \in \mathbb{R}^p$ and $(\gamma_k)_{k \in \mathbb{N}} \in ]0, +\infty[^{\mathbb{N}}$.</td>
</tr>
<tr>
<td>2. For $k = 0, 1, \ldots$ perform:</td>
</tr>
<tr>
<td>$x^{k+1} = \text{prox}_{\gamma_k g(\cdot, \theta)}(x^k - \gamma_k G(x^k, \theta_k))$.</td>
</tr>
</tbody>
</table>

Definitions:

- $\text{prox}_{\lambda g(\cdot, \theta)} := \arg \min_{y \in \mathbb{R}^p} \left\{ g(y, \theta) + \frac{1}{2\lambda} \| y - x \|_2^2 \right\}$
- $\{\theta_k\}_{k=0,1,\ldots}$: sequence of independent random variables.
- $G(x^k, \theta_k) \in \partial f(x^k, \theta_k)$: an unbiased estimate of the deterministic (sub)gradient:
  \[ \mathbb{E}[G(x^k, \theta_k)] \in \partial f(x^k). \]

Remark

Cost of computing $G(x^k, \theta_k)$ is usually much cheaper than $\nabla f(x^k)$. 

Mathematics of Data | Prof. Volkan Cevher, volkan.cevher@epfl.ch
Convergence analysis

Assumptions for the problem setting

▶ $f(\cdot, \theta)$ and $g(\cdot, \theta)$ are convex functions in the first argument, $g$ is proximally-tractable.

▶ (Sub)gradients of $F$ satisfy stochastic bounded gradient condition: $\exists C \geq 0, B \geq 0$ such that

$$
\mathbb{E}_\theta[\| \partial F(x, \theta) \|^2] \leq B^2 + C(F(x) - F(x^*)) .
$$

▶ $\mathbb{E}[\| x^t - x^* \|^2] \leq R^2$ for all $t \geq 0$.

Implications of the assumptions

▶ None of the above assumptions enforce that $f$ is smooth.

▶ Stochastic bounded gradient condition holds with $C = 0$ when both $f(\cdot, \theta)$ and $g(\cdot, \theta)$ are Lipschitz continuous.

▶ The same condition holds when $f(\cdot, \theta)$ is $L_f$-smooth and $g(\cdot, \theta)$ is Lipschitz continuous.

▶ However, for the upcoming theorem, we will take $C > 0$, which rules out the case when both functions are only Lipschitz continuous.
Convergence analysis

Assumptions for the problem setting

▶ $f(\cdot, \theta)$ and $g(\cdot, \theta)$ are convex functions in the first argument, $g$ is proximally-tractable.

▶ (Sub)gradients of $F$ satisfy stochastic bounded gradient condition: \( \exists C \geq 0, B \geq 0 \) such that

\[
\mathbb{E}_\theta [\| \partial F(x, \theta) \|^2] \leq B^2 + C(F(x) - F(x^*)) .
\]

▶ \( \mathbb{E}[\| x^t - x^* \|^2] \leq R^2 \) for all \( t \geq 0 \).

Theorem (Ergodic convergence [12])

▶ Assume the above assumptions hold with $C > 0$.

▶ Let the sequence \( \{x^k\}_{k \geq 0} \) be generated by SPG.

▶ Set $\gamma_k = 1/(C\sqrt{k})$.

Conclusion:

▶ Define $\bar{x}^k = \frac{1}{k} \sum_{i=0}^{k-1} x^i$, then

\[
\mathbb{E}[F(\bar{x}^k) - F(x^*)] \leq \frac{1}{\sqrt{k}} \left( R^2 C + \frac{B^2}{C} \right) , \quad \forall k \geq 1 .
\]
Revisiting a special composite structure

A basic constrained problem setting

\[
\begin{align*}
    f^* &:= \min_{x \in \mathbb{R}^p} \left\{ f(x) + \delta_{\mathcal{X}}(x) \right\} := \min_{x \in \mathbb{R}^p} \left\{ f(x) : x \in \mathcal{X} \right\}, \\
\end{align*}
\]

(8)

Assumptions

- \(\mathcal{X}\) is nonempty, convex and compact (closed and bounded) where \(\delta_{\mathcal{X}}\) is its indicator function.
- \(f \in \mathcal{F}_{L,1}^1(\mathbb{R}^p)\) (i.e., convex with Lipschitz gradient).

Recall proximal gradient algorithm

Basic proximal-gradient scheme (ISTA)

1. Choose \(x^0 \in \text{dom}(F)\) arbitrarily as a starting point.
2. For \(k = 0, 1, \ldots\), generate a sequence \(\{x^k\}_{k \geq 0}\) as:

\[
x^{k+1} := \text{prox}_{\alpha g} \left( x^k - \alpha \nabla f(x^k) \right)
\]

where \(\alpha := 1/L\).

- Prox-operator of indicator of \(\mathcal{X}\) is projection onto \(\mathcal{X}\) \implies ensures feasibility

How else can we ensure feasibility?
Frank-Wolfe’s approach - I

\[ f^* := \min_{x \in \mathbb{R}^p} \left\{ f(x) : x \in \mathcal{X} \right\}, \]

**Conditional gradient method (CGM, see [10] for review)**

A plausible strategy which dates back to 1956 [6]. At iteration \( k \):

1. Consider the linear approximation of \( f \) at \( x^k \)

\[ \phi_k(x) := f(x^k) + \nabla f(x^k)^T(x - x^k) \]

2. Minimize this approximation within constraint set

\[ \hat{x}^k \in \min_{x \in \mathcal{X}} \phi_k(x) = \min_{x \in \mathcal{X}} \nabla f(x^k)^T x \]

3. Take a step towards \( \hat{x}^k \) with step-size \( \gamma_k \in [0, 1] \)

\[ x^{k+1} = x^k + \gamma_k(\hat{x}^k - x^k) \]

\[ x^{k+1} \text{ is feasible since it is convex combination of two other feasible points.} \]
Frank-Wolfe’s approach - II

\[ f^* := \min_{x \in \mathbb{R}^p} \left\{ f(x) : x \in \mathcal{X} \right\} \]

\[
\begin{aligned}
\dot{x}^k &:= \arg \min_{x \in \mathcal{X}} \nabla f(x^k)^T x \\
x^{k+1} &:= (1 - \gamma_k)x^k + \gamma_k \dot{x}^k,
\end{aligned}
\]

where \( \gamma_k := \frac{2}{k+2} \).

Conditional gradient method (CGM)

1. Choose \( x^0 \in \mathcal{X} \).
2. For \( k = 0, 1, \ldots \) perform:

\[
\begin{aligned}
\dot{x}^k &:= \arg \min_{x \in \mathcal{X}} \nabla f(x^k)^T x \\
x^{k+1} &:= (1 - \gamma_k)x^k + \gamma_k \dot{x}^k,
\end{aligned}
\]

where \( \gamma_k := \frac{2}{k+2} \).
On the linear minimization oracle

\[ f^* := \min_{x \in \mathbb{R}^p} \left\{ f(x) : x \in \mathcal{X} \right\} \]

Definition (Linear minimization oracle)

Let \( \mathcal{X} \) be a convex, closed and bounded set. Then, the linear minimization oracle of \( \mathcal{X} \) (\( \text{lmo}_{\mathcal{X}} \)) returns a vector \( \hat{x} \) such that

\[ \text{lmo}_{\mathcal{X}}(x) := \hat{x} \in \arg \min_{y \in \mathcal{X}} x^T y \]  

(9)

- \( \text{lmo}_{\mathcal{X}} \) returns an extreme point of \( \mathcal{X} \).
- \( \text{lmo}_{\mathcal{X}} \) is arguably cheaper than projection.
- \( \text{lmo}_{\mathcal{X}} \) is not single valued, note \( \in \) in the definition.
Convergence guarantees of CGM

**Problem setting**

\[
  f^* := \min_{x \in \mathbb{R}^p} \left\{ f(x) : x \in \mathcal{X} \right\},
\]

**Assumptions**

- $\mathcal{X}$ is nonempty, **convex**, closed and **bounded**.
- $f \in \mathcal{F}_L^1,1(\mathbb{R}^p)$ (i.e., convex with Lipschitz gradient).

**Theorem**

*Under assumptions listed above, CGM with step size $\gamma_k = \frac{2}{k+2}$ satisfies*

\[
  f(x^k) - f(x^*) \leq \frac{4LD^2}{k+1}
\]

where $D_{\mathcal{X}} := \max_{x, y \in \mathcal{X}} \|x - y\|_2$ is diameter of constraint set.
Convergence guarantees of CGM: A faster rate

Problem setting

\[ f^* := \min_{x \in \mathbb{R}^p} \left\{ f(x) : x \in \mathcal{X} \right\}, \]

Assumptions

- \( \mathcal{X} \) is nonempty, \( \alpha \)-strongly convex, closed and bounded.
- \( f \in \mathcal{F}_{L,\mu}^1(\mathbb{R}^p) \) (i.e., strongly convex with Lipschitz gradient).

Definition (\( \alpha \)-strongly convex set) [7]

A convex set \( \mathcal{X} \subseteq \mathbb{R}^{p \times p} \) is \( \alpha \)-strongly convex with respect to \( \| \cdot \| \) if for any \( x, y \in \mathcal{X} \), any \( \gamma \in [0, 1] \) and any vector \( z \in \mathbb{R}^{p \times p} \) such that \( \| z \| = 1 \), it holds that

\[ \gamma x + (1 - \gamma) y + \gamma (1 - \gamma) \frac{\alpha}{2} \| x - y \|^2 z \in \mathcal{X} \]

That is, for any \( x, y \in \mathcal{X} \), the ball centered at \( \gamma x + (1 - \gamma) y \) with radius \( \gamma (1 - \gamma) \frac{\alpha}{2} \| x - y \|^2 \) is contained in \( \mathcal{X} \).
CGM for strongly convex objective + strongly convex set

<table>
<thead>
<tr>
<th>Conditional gradient method - CGM2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Choose $x^0 \in X$.</td>
</tr>
<tr>
<td>2. For $k = 0, 1, \ldots$ perform:</td>
</tr>
<tr>
<td>$\hat{x}^k := \arg \min_{x \in X} \nabla f(x) \cdot x$</td>
</tr>
<tr>
<td>$\gamma_k := \arg \min_{\gamma \in [0, 1]} \gamma \langle \hat{x}^k - x^k, \nabla f(x^k) \rangle + \gamma^2 \frac{L}{2} | \hat{x}^k - x^k |^2$</td>
</tr>
<tr>
<td>$x^{k+1} := (1 - \gamma_k)x^k + \gamma_k \hat{x}^k$,</td>
</tr>
</tbody>
</table>

**Theorem ([7])**

*Under assumptions listed previously, CGM2 satisfies*

$$f(x^k) - f(x^*) = O \left( \frac{1}{k^2} \right).$$  \(11\)
Example: lmo of nuclear-norm ball

Consider $\delta_{\mathcal{X}}$, the indicator of nuclear-norm ball $\mathcal{X} := \{X : X \in \mathbb{R}^{P \times P}, \|X\|_* \leq \alpha\}$

$lmo$ of nuclear-norm ball

$$lmo_{\mathcal{X}}(X) := \hat{X} \in \arg \min_{Y \in \mathcal{X}} \langle Y, X \rangle$$

This can be computed as follows:

- Compute top singular vectors of $X$ $\implies (u_1, \sigma_1, v_1) = \text{svds}(X, 1)$.
- Form the rank-1 output $\implies X = -u_1 \alpha v_1^T$

We can efficiently approximate top singular vectors by power method!
**Proximal gradient vs. Frank-Wolfe**

**Definitions:**
- Here: $sv =$ scalar-vector multiplication, $v+ =$ vector addition.
- $R_0 := \max_{x^* \in S^*} \|x^0 - x^*\|$ is the maximum initial distance.
- $D_X := \max_{x,y \in X} \|x - y\|_2$ is diameter of constraint set $X$.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Proximal-gradient scheme</th>
<th>Frank-Wolfe method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rate</td>
<td>$O\left(\frac{(L_f R_0^2)}{k}\right)$</td>
<td>$O\left(\frac{(L_f D_X^2)}{k}\right)$</td>
</tr>
<tr>
<td>Complexity</td>
<td>$O\left(\frac{R_0^2 (L_f / \varepsilon)}{\varepsilon}\right)$</td>
<td>$O\left(\frac{D_X^2 (L_f / \varepsilon)}{\varepsilon}\right)$</td>
</tr>
<tr>
<td>Per iteration</td>
<td>1-gradient, 1-prox, 1-$sv$, 1-$v+$</td>
<td>1-gradient, 1-lmo, 2-$sv$, 1-$v+$</td>
</tr>
</tbody>
</table>

How do $prox$ operator and $lmo$ compare in practice?
An example with matrices

Problem Definition

\[ \min_{X \in \mathbb{R}^{p \times p}} f(X) + g(X) \]

- Define \( g(X) = \delta_{\mathcal{X}}(X) \), where \( \mathcal{X} := \{ X : X \in \mathbb{R}^{p \times p}, \| X \|_* \leq \alpha \} \) is nuclear norm ball.
- This problem is equivalent to:
  \[ \min_{X \in \mathcal{X}} f(X) \]

Observations

- \( \text{prox}_g = \pi_{\mathcal{X}} \). Projection requires full SVD, \( O(p^3) \).
- \( \text{lmom} \) computes (approximately) top singular vectors, roughly in \( \approx O(p^2) \) with Lanczos algorithm.
Example: Phase retrieval

Phase retrieval

Aim: Recover signal $x^\dagger \in \mathbb{C}^p$ from the measurements $b \in \mathbb{R}^n$:

$$b_i = \left| \langle a_i, x^\dagger \rangle \right|^2 + \omega_i.$$  

($a_i \in \mathbb{C}^p$ are known measurement vectors, $\omega_i$ models noise).

- Non-linear measurements $\rightarrow$ non-convex maximum likelihood estimators.

PhaseLift [4]

Phase retrieval can be solved as a convex matrix completion problem, following a combination of:

- semidefinite relaxation ($x^\dagger x^\dagger H = X^\dagger$)
- convex relaxation ($\text{rank} \rightarrow \| \cdot \|_*$)

albeit in terms of the lifted variable $X \in \mathbb{C}^{p \times p}$.
Problem formulation

We solve the following PhaseLift variant:

\[
    f^* := \min_{X \in \mathbb{C}^{p \times p}} \left\{ \frac{1}{2} \| A(X) - b \|_2^2 : \|X\|_* \leq \kappa, \ X \geq 0 \right\}.
\]  (12)

Experimental setup [19]

Coded diffraction pattern measurements, \( b = [b_1, \ldots, b_L] \) with \( L = 20 \) different masks

\[
b_\ell = |\text{fft}(d_\ell^H \odot x^\ell)|^2
\]

\( \odot \) denotes Hadamard product; \( | \cdot |^2 \) applies element-wise

\( d_\ell \) are randomly generated octonary masks (distributions as proposed in [4])

Parametric choices: \( \lambda^0 = 0^n; \quad \epsilon = 10^{-2}; \quad \kappa = \text{mean}(b) \).
Test with synthetic data: Prox vs sharp

→ Synthetic data: $x^\natural = \text{randn}(p, 1) + i \cdot \text{randn}(p, 1)$.

→ Stopping criteria: $\frac{\|x^k - x^\natural\|_2}{\|x^\natural\|_2} \leq 10^{-2}$.

→ Averaged over 10 Monte-Carlo iterations.

Note that the problem is $p \times p$ dimensional!
A basic constrained non-convex problem

Problem setting

\[ f^* := \min_{x \in \mathbb{R}^p} \left\{ f(x) : x \in \mathcal{X} \right\}, \]

Assumptions

- \( \mathcal{X} \) is nonempty, convex, closed and bounded.
- \( f \) has \( L \)-Lipschitz continuous gradients, but it is non-convex.

Stationary point

Due to constraints, \( \| \nabla f(x^*) \| = 0 \) may not hold!

Frank-Wolfe gap: Following measure, known as FW-gap, generalizes the definition of stationary point for constrained problems:

\[ g_{FW}(x) := \max_{y \in \mathcal{X}} (x - y)^T \nabla f(x) \]

- \( g_{FW}(x) \geq 0 \) for all \( x \in \mathcal{X} \).
- \( x \in \mathcal{X} \) is a stationary point if and only if \( g_{FW}(x) = 0 \).
CGM for non-convex problems

1. Choose \( x^0 \in \mathcal{X} \), \( K > 0 \) total number of iterations.
2. For \( k = 0, 1, \ldots, K - 1 \) perform:
   \[
   \begin{align*}
   \hat{x}^k & := \text{lmo}_\mathcal{X}(\nabla f(x^k)) \\
   x^{k+1} & := (1 - \gamma_k)x^k + \gamma_k \hat{x}^k,
   \end{align*}
   \]
   where \( \gamma_k := \frac{1}{\sqrt{K+1}} \).

Theorem

Denote \( \bar{x} \) chosen uniformly random from \( \{x^1, x^2, \ldots, x^K\} \). Then, CGM satisfies

\[
\min_{k=1,2,\ldots,K} g_{FW}(x^k) \leq \mathbb{E}[g_{FW}(\bar{x})] \leq \frac{1}{\sqrt{K}} \left( f(x^0) - f^* + \frac{LD^2}{2} \right).
\]

A basic constrained stochastic problem

Problem setting (Stochastic)

\[ f^* := \min_{x \in \mathbb{R}^p} \left\{ \mathbb{E}[f(x, \theta)] : x \in \mathcal{X} \right\}, \quad (13) \]

Assumptions

- \( \theta \) is a random vector whose probability distribution is supported on set \( \Theta \)
- \( \mathcal{X} \) is nonempty, convex, closed and bounded.
- \( f(\cdot, \theta) \in \mathcal{F}_{L^1}^{1,1}(\mathbb{R}^p) \) for all \( \theta \) (i.e., convex with Lipschitz gradient).

Example (Finite-sum model)

\[ \mathbb{E}[f(x, \theta)] = \frac{1}{n} \sum_{j=1}^{n} f_j(x) \]

- \( j = \theta \) is a drawn uniformly from \( \Theta = \{1, 2, \ldots, n\} \)
- \( f_j \in \mathcal{F}_{L^1}^{1,1}(\mathbb{R}^p) \) for all \( j \) (i.e., convex with Lipschitz gradient).
Stochastic conditional gradient method

1. Choose $x^0 \in \mathcal{X}$.
2. For $k = 0, 1, \ldots$ perform:

$$\begin{align*}
\hat{x}^k &:= \text{lm} \max \mathcal{X} (\tilde{\nabla} f (x^k, \theta_k)) \\
x^{k+1} &= (1 - \gamma_k) x^k + \gamma_k \hat{x}^k,
\end{align*}$$

where $\gamma_k := \frac{2}{k+2}$, and $\tilde{\nabla} f$ is an unbiased estimator of $\nabla f$.

Theorem [9]

Assume that the following variance condition holds

$$\mathbb{E} \| \nabla f(x^k) - \tilde{\nabla} f(x^k, \theta_k) \|^2 \leq \left( \frac{LD}{k+1} \right)^2. \quad (*)$$

Then, the iterates of SFW satisfies

$$\mathbb{E} [f(x^k, \theta)] - f^* \leq \frac{4LD^2}{k + 1}.$$

$(*) \rightarrow$ SFW requires decreasing variance!
Stochastic conditional gradient method

<table>
<thead>
<tr>
<th>Stochastic conditional gradient method (SFW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Choose $x^0 \in \mathcal{X}$.</td>
</tr>
<tr>
<td>2. For $k = 0, 1, \ldots$ perform:</td>
</tr>
</tbody>
</table>
| \[
\begin{align*}
\hat{x}^k & := \text{lmo}_\mathcal{X}(\tilde{\nabla} f(x^k, \theta_k)) \\
x^{k+1} & := (1 - \gamma_k)x^k + \gamma_k\hat{x}^k,
\end{align*}
\]
| where $\gamma_k := \frac{2}{k+2}$, and $\tilde{\nabla} f$ is an unbiased estimator of $\nabla f$. |

Example (Finite-sum model)

\[
\mathbb{E}[f(x, \theta)] = \frac{1}{n} \sum_{j=1}^{n} f_j(x)
\]

Assume $f_j$ is $G$-Lipschitz continuous for all $j$. Suppose that $S_k$ is a random sampling (with replacement) from $\Theta = \{1, 2, \ldots, n\}$. Then,

\[
\tilde{\nabla} f(x^k, \theta_k) := \frac{1}{|S_k|} \sum_{j \in S_k} f_j(x^k) \implies \mathbb{E}\|\nabla f(x) - \tilde{\nabla} f(x, \theta_k)\|^2 \leq \frac{G^2}{|S_k|}.
\]

Hence, by choosing $|S_k| = \left(\frac{G(k+1)}{LD}\right)^2$ we satisfy the variance condition for SFW.
Wrap up!

- Monday: Transition from variance reduction to deep learning...
By definition, \( g(y) + \frac{1}{2\lambda} \| y - x \|^2 \) attains its minimum when \( y = \text{prox}_{\lambda g}(x) \).

One can see that \( g(y) + \frac{1}{2\lambda} \| y - x \|^2 \) is convex, and prox operator computes its minimizer over \( \mathbb{R}^p \).

As a result, subdifferential of \( g(y) + \frac{1}{2\lambda} \| y - x \|^2 \) at the minimizer \( y = \text{prox}_{\lambda g}(x) \) should include 0.

Hence, \( 0 \in \partial g(\text{prox}_{\lambda g}(x)) + \frac{1}{\lambda} \left( \text{prox}_{\lambda g}(x) - x \right) \).

After rearranging the above inclusion we obtain: \( x \in \lambda \partial g(\text{prox}_{\lambda g}(x)) + \text{prox}_{\lambda g}(x) \).

We can rewrite the RHS as a single function: \( \lambda \partial g(\text{prox}_{\lambda g}(x)) + \text{prox}_{\lambda g}(x) = (\lambda \partial g + I)(\text{prox}_{\lambda g}(x)) \).

The inclusion becomes: \( x \in (\lambda \partial g + I)(\text{prox}_{\lambda g}(x)) \).

Finally, we compute the inverse of \( (\lambda \partial g + I)(\cdot) \) to conclude: \( \text{prox}_{\lambda g}(x) = (\lambda \partial g + I)^{-1}(x) \).

In the literature, \( (\lambda \partial g + I)^{-1} \) is called the resolvent of the subdifferential of \( g \) with parameter \( \lambda \).

This is just a technical term that stands for proximal operator of \( \lambda g \), as we have defined in this course.
*A short detour: Basic properties of prox-operator

**Property (Basic properties of prox-operator)**

1. $\text{prox}_g(x)$ is **well-defined** and **single-valued** (i.e., the prox-operator (3) has a unique solution since $g(\cdot) + (1/2)\|\cdot - x\|^2$ is strongly convex).

2. **Optimality condition:**
   
   $x \in \text{prox}_g(x) + \partial g(\text{prox}_g(x)), \ x \in \mathbb{R}^p.$

3. $x^*$ is a **fixed point** of $\text{prox}_g(\cdot)$:
   
   $0 \in \partial g(x^*) \iff x^* = \text{prox}_g(x^*).$

4. **Nonexpansiveness:**
   
   $\|\text{prox}_g(x) - \text{prox}_g(\tilde{x})\|_2 \leq \|x - \tilde{x}\|_2, \ \forall x, \tilde{x} \in \mathbb{R}^p.$

**Note:** An operator is called **non-expansive** if it is $L$-Lipschitz continuous with $L = 1.$
Adaptive Restart

It is possible to preserve $\mathcal{O}(1/k^2)$ convergence guarantee!

One needs to slightly modify the algorithm as below.

**Generalized fast proximal-gradient scheme**

1. Choose $x^0 = x^{-1} \in \text{dom}(F)$ arbitrarily as a starting point.
2. Set $\theta_0 = \theta_{-1} = 1$, $\lambda := L_f^{-1}$
3. For $k = 0, 1, \ldots$, generate two sequences $\{x^k\}_{k \geq 0}$ and $\{y^k\}_{k \geq 0}$ as:

\[
\begin{align*}
{y^k} & := x^k + \theta_k(\theta_{k-1}^{-1} - 1)(x^k - x^{k-1}) \\
{x^{k+1}} & := \text{prox}_{\lambda g}(y^k - \lambda \nabla f(y^k)), \quad \text{if restart test holds}
\end{align*}
\]

\[
\begin{align*}
\theta_{k-1} & = \theta_k = 1 \\
y^k & = x^k \\
x^{k+1} & := \text{prox}_{\lambda g}(y^k - \lambda \nabla f(y^k))
\end{align*}
\]

\[\theta_k\] is chosen so that it satisfies

\[
\theta_{k+1} = \frac{\sqrt{\theta_k^4 + 4\theta_k^2} - \theta_k^2}{2} < \frac{2}{k + 3}
\]

\[\text{Mathematics of Data | Prof. Volkan Cevher, volkan.cevher@epfl.ch}

Slide 3/25
Adaptive Restart: Guarantee

**Theorem (Global complexity [8])**

The sequence \( \{x^k\}_{k \geq 0} \) generated by the modified algorithm satisfies

\[
F(x^k) - F^* \leq \frac{2L_f}{(k + 2)^2} \left( R_0^2 + \sum_{k_i \leq k} \left( \|x^* - x^{k_i}\|_2^2 - \|x^* - z^{k_i}\|_2^2 \right) \right) \quad \forall k \geq 0.
\]

(15)

where \( R_0 := \min_{x^* \in S^*} \|x^0 - x^*\| \), \( z^k = x^{k-1} + \theta_{k-1}^{-1}(x^k - x^{k-1}) \) and \( k_i, i = 1... \) are the iterations for which the restart test holds.

Various restarts tests that might coincide with \( \|x^* - x^{k_i}\|_2^2 \leq \|x^* - z^{k_i}\|_2^2 \)

- Exact non-monotonicity test: \( F(x^{k+1}) - F(x^k) > 0 \)
- Non-monotonicity test: \( \langle (L_F(y^{k-1} - x^k), x^{k+1} - \frac{1}{2}(x^k + y^{k-1})) \rangle > 0 \) (implies exact non-monotonicity and it is advantageous when function evaluations are expensive)
- Gradient-mapping based test: \( \langle (L_f(y^k - x^{k+1}), x^{k+1} - x^k) \rangle > 0 \)
Recall: Composite convex minimization

Problem (Unconstrained composite convex minimization)

\[
F^\star := \min_{x \in \mathbb{R}^p} \{ F(x) := f(x) + g(x) \}
\]  

- \( f \) and \( g \) are both proper, closed, and convex.
- \( \operatorname{dom}(F) := \operatorname{dom}(f) \cap \operatorname{dom}(g) \neq \emptyset \) and \(-\infty < F^\star < +\infty\).
- The solution set \( S^\star := \{ x^\star \in \operatorname{dom}(F) : F(x^\star) = F^\star \} \) is nonempty.
*Recall: Composite convex minimization guarantees

Proximal gradient method (ISTA) vs. fast proximal gradient method (FISTA)

**Assumptions, step sizes and convergence rates**

Proximal gradient method:

\[
f \in \mathcal{F}^1_{1,L}, \quad \alpha = \frac{1}{L}, \quad F(x^k) - F(x^*) \leq \epsilon, \quad O\left(\frac{1}{\epsilon}\right).
\]

Fast proximal gradient method:

\[
f \in \mathcal{F}^1_{L}, \quad \alpha = \frac{1}{L}, \quad F(x^k) - F(x^*) \leq \epsilon, \quad O\left(\frac{1}{\sqrt{\epsilon}}\right).
\]
*Recall: Composite convex minimization guarantees

Proximal gradient method (ISTA) vs. fast proximal gradient method (FISTA)

### Assumptions, step sizes and convergence rates

#### Proximal gradient method:

\[ f \in \mathcal{F}_{L}^{1,1}, \quad \alpha = \frac{1}{L} \]

\[ F(x^k) - F(x^*) \leq \epsilon, \quad O\left(\frac{1}{\epsilon}\right). \]

#### Fast proximal gradient method:

\[ f \in \mathcal{F}_{L}^{1,1}, \quad \alpha = \frac{1}{L} \]

\[ F(x^k) - F(x^*) \leq \epsilon, \quad O\left(\frac{1}{\sqrt{\epsilon}}\right). \]

- We require \( \alpha_k \) to be a function of \( L \).
- It may not be possible to know exactly the Lipschitz constant. Line-search?
- Adaptation to local geometry → may lead to larger steps.
How can we better adapt to the local geometry?

Non-adaptive:

\[
\begin{align*}
\|\nabla f(x) - \nabla f(y)\| & \leq L \|y - x\| \\
L & \text{is a global worst-case constant}
\end{align*}
\]
*How can we better adapt to the local geometry?

**Line-search:**

\[
x^{k+1} = \arg\min_x \left\{ f(x^k) + \langle \nabla f(x^k), x - x^k \rangle + \frac{L_k}{2} ||x - x^k||^2_2 \right\}
\]

\[\|\nabla f(x) - \nabla f(y)\| \leq L\|y - x\|\]

L is a global worst-case constant

Local quadratic upper bound

\[Q_{L_k}(x, x^k)\]

\[f(x) \leq f(x^k) + \nabla f(x^k)^T(x - x^k) + \frac{L_k}{2} ||x - x^k||^2_2\]

applies only locally
*How can we better adapt to the local geometry?*

**Variable metric:**

\[ f(x) = f(x^k) + r f(x^k)^T (x - x^k) + \frac{L}{2} \|x - x^k\|^2 \]

\[ \|\nabla f(x) - \nabla f(y)\| \leq L \|y - x\| \]

\[ L \text{ is a global worst-case constant} \]

\[ x^{k+1} = \arg \min_x \left\{ f(x^k) + \langle \nabla f(x^k), x - x^k \rangle + \frac{L}{2} \|x - x^k\|^2 \right\} \]
The idea of the proximal-Newton method

Assumptions A.2

Assume that $f \in \mathcal{F}_{L,\mu}^{2,1}(\mathbb{R}^p)$ and $g \in \mathcal{F}_{\text{prox}}(\mathbb{R}^p)$.

*Proximal-Newton update

Similar to classical newton, proximal-newton suggests the following update scheme using second order Taylor series expansion near $x_k$.

$$x^{k+1} := \arg \min_{x \in \mathbb{R}^p} \left\{ \frac{1}{2} (x - x^k)^T \nabla^2 f(x^k) (x - x^k) + \nabla f(x^k)^T (x - x^k) + g(x) \right\}.$$  

(17)
The proximal-Newton-type algorithm

**Proximal-Newton algorithm (PNA)**

1. Given $x^0 \in \mathbb{R}^p$ as a starting point.
2. For $k = 0, 1, \cdots$, perform the following steps:
   2.1. Evaluate an SDP matrix $H_k \approx \nabla^2 f(x^k)$ and $\nabla f(x^k)$.
   2.2. Compute $d^k := \text{prox}_{H_k^{-1} g} \left( x^k - H_k^{-1} \nabla f(x^k) \right) - x^k$.
   2.3. Update $x^{k+1} := x^k + \alpha_k d^k$. 

**Remark**
- $H_k \equiv \nabla^2 f(x)$ ⇒ proximal-Newton algorithm.
- $H_k \approx \nabla^2 f(x)$ ⇒ proximal-quasi-Newton algorithm.
- A generalized prox-operator: $\text{prox}_{H_k^{-1} g} \left( x^k - H_k^{-1} \nabla f(x^k) \right)$.
The proximal-Newton-type algorithm

Proximal-Newton algorithm (PNA)

1. Given $x^0 \in \mathbb{R}^p$ as a starting point.
2. For $k = 0, 1, \ldots$, perform the following steps:
   2.1. Evaluate an SDP matrix $H_k \approx \nabla^2 f(x^k)$ and $\nabla f(x^k)$.
   2.2. Compute $d^k := \text{prox}_{H_k^{-1}} g \left( x^k - H_k^{-1} \nabla f(x^k) \right) - x^k$.
   2.3. Update $x^{k+1} := x^k + \alpha_k d^k$.

Remark

- $H_k \equiv \nabla^2 f(x^k) \implies$ proximal-Newton algorithm.
- $H_k \approx \nabla^2 f(x^k) \implies$ proximal-quasi-Newton algorithm.
- A generalized prox-operator: $\text{prox}_{H_k^{-1} g} \left( x^k + H_k^{-1} \nabla f(x^k) \right)$. 
*Convergence analysis

Theorem (Global convergence [11])

Assume generalized-prox subproblem is solved exactly for the algorithm and there exists $\mu > 0$ such that $H_k \succeq \mu I$ for all $k \geq 0$. Then:

\[
\{x^k\}_{k \geq 0} \text{ globally converges to a solution } x^* \text{ of (16)}.
\]
**Convergence analysis**

**Theorem (Global convergence [11])**

Assume generalized-prox subproblem is solved exactly for the algorithm and there exists \( \mu > 0 \) such that \( H_k \succeq \mu I \) for all \( k \geq 0 \). Then;

\[
\{x^k\}_{k \geq 0} \text{ globally converges to a solution } x^* \text{ of (16)}.
\]

**Theorem (Local convergence [11])**

Assume generalized-prox subproblem is solved exactly for the algorithm there exists \( 0 < \mu \leq L_2 < +\infty \) such that \( \mu I \preceq H_k \preceq L_2 I \) for all sufficiently large \( k \). Then;

- If \( H_k \equiv \nabla^2 f(x^k) \), then \( \alpha_k = 1 \) for \( k \) sufficiently large (full-step).
- If \( H_k \equiv \nabla^2 f(x^k) \), then \( \{x^k\} \) locally converges to \( x^* \) at a quadratic rate.
- If \( H_k \) satisfies the Dennis-Moré condition:

\[
\lim_{k \to +\infty} \frac{\| (H_k - \nabla^2 f(x^*)) (x^{k+1} - x^k) \|}{\| x^{k+1} - x^k \|} = 0,
\]

then \( \{x^k\} \) locally converges to \( x^* \) at a super linear rate.
How to compute the approximation $H_k$?

How to update $H_k$?

Matrix $H_k$ can be updated by using low-rank updates.

- **BFGS update**: maintain the Dennis-Moré condition and $H_k \succ 0$.

$$H_{k+1} := H_k + \frac{y_k y_k^T}{s_k^T y_k} - \frac{H_k s_k s_k^T H_k}{s_k^T H_k s_k}, \quad H_0 := \gamma I, \quad (\gamma > 0).$$

where $y_k := \nabla f(x_{k+1}) - \nabla f(x_k)$ and $s_k := x_{k+1} - x_k$.

- **Diagonal+Rank-1 [3]**: computing PN direction $d^k$ is in polynomial time, but it does not maintain the Dennis-Moré condition:

$$H_k := D_k + u_k u_k^T, \quad u_k := (s_k - H_0 y_k) / \sqrt{(s_k - H_0 y_k)^T y_k},$$

where $D_k$ is a positive diagonal matrix.
Pros and cons

Pros
- Fast local convergence rate (super-linear or quadratic)
- Numerical robustness under the inexactness/noise ([11]).
- Well-suited for problems with many data points but few parameters. For example,

\[ F^* := \min_{x \in \mathbb{R}^p} \left\{ \sum_{j=1}^{n} \ell_j (a_j^T x + b_j) + g(x) \right\}, \]

where \( \ell_j \) is twice continuously differentiable and convex, \( g \in \mathcal{F}_{\text{prox}}, p \ll n \).
Pros and cons

Pros

▶ Fast local convergence rate (super-linear or quadratic)
▶ Numerical robustness under the inexactness/noise ([11]).
▶ Well-suited for problems with many data points but few parameters. For example,

\[
F^* := \min_{x \in \mathbb{R}^p} \left\{ \sum_{j=1}^{n} \ell_j(a_j^T x + b_j) + g(x) \right\},
\]

where \( \ell_j \) is twice continuously differentiable and convex, \( g \in \mathcal{F}_{\text{prox}} \), \( p \ll n \).

Cons

▶ Expensive iteration compared to proximal-gradient methods.
▶ Global convergence rate may be worse than accelerated proximal-gradient methods.
▶ Requires a good initial point to get fast local convergence.
▶ Requires strict conditions for global/local convergence analysis.
**Example 1: Sparse logistic regression**

**Problem (Sparse logistic regression)**

Given a sample vector $\mathbf{a} \in \mathbb{R}^p$ and a binary class label vector $\mathbf{b} \in \{-1, +1\}^n$. The conditional probability of a label $b$ given $\mathbf{a}$ is defined as:

$$
\mathbb{P}(b|\mathbf{a}, \mathbf{x}, \mu) = \frac{1}{1 + e^{-b(\mathbf{x}^T \mathbf{a} + \mu)}},
$$

where $\mathbf{x} \in \mathbb{R}^p$ is a weight vector, $\mu$ is called the intercept.

**Goal:** Find a sparse-weight vector $\mathbf{x}$ via the maximum likelihood principle.

**Optimization formulation**

$$
\min_{\mathbf{x} \in \mathbb{R}^p} \left\{ \frac{1}{n} \sum_{i=1}^{n} L(b_i(\mathbf{a}_i^T \mathbf{x} + \mu)) + \rho \|\mathbf{x}\|_1 \right\},
$$

where $\mathbf{a}_i$ is the $i$-th row of data matrix $\mathbf{A}$ in $\mathbb{R}^{n \times p}$, $\rho > 0$ is a regularization parameter, and $\ell$ is the logistic loss function $\ell(\tau) := \log(1 + e^{-\tau})$. 
*Example: Sparse logistic regression

Real data
- Real data: w2a with $n = 3470$ data points, $p = 300$ features

Parameters
- Tolerance $10^{-6}$.
- L-BFGS memory $m = 50$.
- Ground truth: Get a high accuracy approximation of $x^*$ and $f^*$ by TFOCS with tolerance $10^{-12}$. 
*Example: Sparse logistic regression-Numerical results

![Graph showing numerical results for different methods: Pure Newton, Quasi-Newton with BFGS, Quasi-Newton with L-BFGS, Accelerated gradient method, Line Search AGD with adaptive restart.](image)

- **Number of iterations**: $(F(x^k) - F^*)/F^*$ in log scale
- **Time (s)**: $(F(x^k) - F^*)/F^*$ in log scale

Methods compared:
- Pure Newton
- Quasi-Newton with BFGS
- Quasi-Newton with L-BFGS
- Accelerated gradient method
- Line Search AGD with adaptive restart
Example 2: $\ell_1$-regularized least squares

Problem ($\ell_1$-regularized least squares)

Given $A \in \mathbb{R}^{n \times p}$ and $b \in \mathbb{R}^n$, solve:

$$F^* := \min_{x \in \mathbb{R}^p} \left\{ F(x) := \frac{1}{2} \|Ax - b\|_2^2 + \rho \|x\|_1 \right\},$$

where $\rho > 0$ is a regularization parameter.

Complexity per iterations

- Evaluating $\nabla f(x^k) = A^T(Ax^k - b)$ requires one $Ax$ and one $A^Ty$.
- One soft-thresholding operator $\text{prox}_\lambda g(x) = \text{sign}(x) \otimes \max\{|x| - \rho, 0\}$.
- Optional: Evaluating $L = \|A^TA\|$ (spectral norm) - via power iterations (e.g., 20 iterations, each iteration requires one $Ax$ and one $A^Ty$).

Synthetic data generation

- $A := \text{randn}(n, p)$ - standard Gaussian $\mathcal{N}(0, I)$.
- $x^*$ is a $s$-sparse vector generated randomly.
- $b := Ax^* + \mathcal{N}(0, 10^{-3})$. 
Example 2: $\ell_1$-regularized least squares - Numerical results - Trial 1

Parameters: $n = 750, p = 2000, s = 200, \rho = 1$
*Example 2: $\ell_1$-regularized least squares - Numerical results - Trial 2

Parameters: $n = 750, p = 2000, s = 200, \rho = 1$
References

First-order methods in optimization, volume 25.
(Cited on pages 36, 37, and 38.)

A fast iterative shrinkage-thresholding algorithm for linear inverse problems.
(Cited on pages 19, 20, and 22.)

A quasi-newton proximal splitting method.
(Cited on page 78.)

Phaselift: Exact and stable signal recovery from magnitude measurements via convex programming.
(Cited on pages 54 and 55.)
References II

Springer-Velarg, 2011.
(Cited on page 17.)

An algorithm for quadratic programming.
(Cited on page 45.)

Faster rates for the frank-wolfe method over strongly-convex sets.
(Cited on pages 49 and 50.)

Monotonicity and restart in fast gradient methods.
(Cited on page 66.)
Variance-reduced and projection-free stochastic optimization.
(Cited on page 60.)

(Cited on page 45.)

Proximal newton-type methods for convex optimization.
(Cited on pages 76, 77, 79, and 80.)

[12] Ion Necoara.
(Cited on page 43.)
A unified framework for high-dimensional analysis of $M$-estimators with decomposable regularizers.
(Cited on page 4.)

*Introductory lectures on convex optimization: A basic course*, volume 87.
(Cited on page 31.)

Proximal algorithms.
(Cited on page 17.)

[16] Pradeep Ravikumar, Martin J. Wainwright, Garvesh Raskutti, and Bin Yu.
High-dimensional covariance estimation by minimizing $\ell_1$-penalized log-determinant divergence.
(Cited on page 5.)
